T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

TÜRKÇE DİLİNDE AÇIK UÇLU BİLGİ ERİŞİMİ İÇİN COLBERT MODELİ GELİŞTİRİLMESİ

Eren TAHİR

YÜKSEK LİSANS TEZİ

Veri Bilimi ve Büyük Veri Anabilim Dalı

Veri Bilimi ve Büyük Veri Programı

Danışman

Doç. Dr. Mert BAL

Aralık, 2025

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Türkçe Dilinde Açık Uçlu Bilgi Erişimi İçin Colbert Modeli Geliştirilmesi

Eren TAHİR tarafından hazırlanan tez çalışması 22.12.2025 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Veri Bilimi ve Büyük Veri Anabilim Dalı, Veri Bilimi ve Büyük Veri Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Mert BAL
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Doç. Dr. Mert BAL, Danışman
Yıldız Teknik Üniversitesi _____

Doç. Dr. Ramazan TEKERCİOĞLU, Üye
Yıldız Teknik Üniversitesi _____

Dr. Ebu Yusuf GÜVEN, Üye
İstanbul Üniversitesi _____

Danışmanım Doç. Dr. Mert BAL sorumluluğunda tarafımca hazırlanan "Türkçe Dilinde Açık Uçlu Bilgi Erişimi İçin Colbert Modeli Geliştirilmesi" başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Eren TAHİR

İmza

Aileme

ve

eşime

TEŞEKKÜR

Eğitim yolculuğumda bana ışık tutan tüm öğretmenlerime, cumhuriyetimizin kurucularına ve vatanımızın iyiliği için fedakarlıkta bulunan, emek sarf eden herkese en derin şükranlarımı sunarım.

Eren TAHİR

İÇİNDEKİLER

KISALTMA LİSTESİ viii

ŞEKİL LİSTESİ ix

TABLO LİSTESİ x

ÖZET xi

ABSTRACT xiii

1 GİRİŞ 1

1.1 Motivasyon 1

1.2 Tezin Amacı ve Özgün Katkısı 4

2 HAZIRLIK 5

2.1 Bilgi Erişimi 6

2.1.1 Tarihsel Gelişimi……...……...6

2.1.2 Önemi……...…7

2.1.3 Faydaları……...…7

2.1.4 Nasıl Çalışır ……..8

2.1.5 Bilgi Erişim Sistemi Bileşenleri……...…8

2.1.6 Bilgi Erişim Model Türleri……..9

2.1.7 Zorluklar ve Kısıtlamalar…… 10

2.2 Doğal Dil İşleme 10

2.2.1 Metin Ön İşleme Teknikleri ……..…….. 11

2.2.2 Trendler……..……..……12

2.3 İndeksleme 12

2.4 ANN Algoritmaları 13
2.4.1 Annoy 13
2.4.2 HSNW 14
2.5 Büyük Dil Modelleri ve Transformer Mimarisi 15
2.5.1 Transformer Modeli 16
2.5.2 Encoder-Only Transformer Modelleri 17
2.6 Nöral Bilgi Erişim Paradigmaları 21
2.6.1 ColBERT Paradigması 23
2.6.2 Çok Dilli ColBERT Çalışmaları 25
2.7 IR Eğitim Verisetleri Üzerine 31
2.8 IR Başarı Metrikleri 32
2.8.1 Precision (Kesinlik) 32
2.8.2 Recall (Duyarlılık) 33
2.8.3 Precision@k 33
2.8.4 Recall@k 33
2.8.5 R-Precision 34
2.8.6 Mean Average Precision (MAP) 34
2.8.7 Cumulative Gain (CG) 34
2.8.8 Discounted Cumulative Gain (DCG) 35
2.8.9 Normalized DCG (NDCG) 35
2.8.10 Mean Reciprocal Rank (MRR) 36
3 YÖNTEM **37**
3.1 Dil Modeli Mimarisi Seçimi 37
3.2 Dil Modeli Ön Eğitimi 39
3.3 IR İçin İnce Ayar Eğitimi 41
3.3.1 Eğitim Objektifinin Seçimi 42

3.3.2 Eğitim Verisetlerini Oluşturma…… 43

3.3.3 Eğitim Aşaması…… 45

4 DEĞERLENDİRME **46**

4.1 Değerlendirme Verisetlerinin Oluşturulması 46

4.2 Değerlendirme Sonuçları 48

4.2.1 Çıkarım Yapma Verimliliğinin Değerlendirilmesi …… 53

5 SONUÇ **55**

5.1 Tez Sonucu..... 55

5.2 Gelecek Çalışmalar 56

KAYNAKÇA **57**

TEZDEN ÜRETİLMİŞ YAYINLAR **62**

KISALTMA LİSTESİ

ANN	Approximate Nearest Neighbor Search
AP	Average Precision
BEIR	Benchmark For Information Retrieval
CG	Cumulative Gain
DCG	Discounted Cumulative Gain
ENN	Exact Nearest Neighbor Search
GPU	Ekran Kartı (Graphical Processing Unit)
IDF	Inverse Document Frequency
IR	Bilgi Erişim (Information Retrieval)
KD	Bilgi Damıtma (Knowledge Distillation)
LLM	Büyük Dil Modeli (Large Lanmguage Model)
LSI	Latent Semantic Indexing
MAP	Mean Average Precision
MRR	Mean Reciprocal Rank
NDCG	Normalized Discounted Cumulative Gain
NLP	Doğal Dil İşleme (Natural Language Processing)
PLM	Ön Eğitilmiş Dil Modeli (Pretrained Language Model)
RAG	Erişim Destekli Üretim (Retrieval Augmented Generation)
SVD	Singular Value Decomposition
TF	Terim Frekansı (Term Frequency)
VRAM	Video Rastgele Erişimli Bellek (Video Random Access Memory)
YZ	Yapay Zeka
QA	Soru Cevaplama (Question Answering)

ŞEKİL LİSTESİ

Şekil 2.1 Transformer model türleri **17**
Şekil 2.2 Nöral bilgi erişimi paradigmaları **23**
Şekil 2.3 ColBERT mimarisi **24**
Şekil 2.4 MaxSim mekanizması **25**
Şekil 2.5 X-MOD mimarisi **26**
Şekil 2.6 ColBERT-XM eğitim aşamaları **27**
Şekil 2.7 M3-Embedding eğitim aşamaları **28**
Şekil 3.1 Attention implementasyonlarının hız karşılaştırması **38**
Şekil 3.2 Maskelenmiş dil modelleme objektifi **40**
Şekil 3.3 Dropout tekniği **41**
Şekil 3.4 Bilgi damıtma eğitim objektifi **43**

TABLO LİSTESİ

Tablo 2.1 Nöral bilgi erişim paradigmalarının karşılaştırılması **22**
Tablo 4.1 Türkçe NanoBEIR içeriği **48**
Tablo 4.2 Türkçe NanoBEIR nDCG@10 sonuçları **49**
Tablo 4.3 Türkçe NanoBEIR detaylı nDCG sonuçları **50**
Tablo 4.4 Türkçe NanoBEIR detaylı Recall sonuçları **50**
Tablo 4.5 Türkçe NanoBEIR detaylı Precision sonuçları **51**
Tablo 4.6 Türkçe NanoBEIR detaylı MAP sonuçları **52**
Tablo 4.7 Modellerin verimliliğinin karşılaştırılması **54**

ÖZET

Türkçe Dilinde Açık Uçlu Bilgi Erişimi İçin Colbert Modeli Geliştirilmesi

Eren TAHİR

Veri Bilimi ve Büyük Veri Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Mert BAL

Gerçekleştirilen tez çalışmasında Türkçe dili için açık alan bilgi erişimi için bağlamsal derin öğrenme modeli geliştirilmesi ve değerlendirilmesi hedeflenmiştir. Bilgi erişiminde derin öğrenme modelleri etkileşim tabanlı, temsil tabanlı ve hibrit olarak üçe ayrılmaktadır. Etkileşim tabanlı modeller yüksek başarı vadetmesine rağmen eğitim için fazla veriye ihtiyaç duymaları ve tahmin yapma hesaplama maliyetinin yüksek olması dezavantajlarına sahiptir. Temsil tabanlı modeller de veri efektif modeller olmamakla birlikte düşük tahmin yapma kompleksliğiyle öne çıkmaktadır. Hem temsil tabanlı modeller hem etkileşim tabanlı modeller eğitim verisinden farklı veri dağılımlarında ciddi başarı kaybı gösterirler. Hibrit yöntemlerden biri olan ColBERT paradigması bu iki yöntemin güçlü yanlarını birleştirerek veri efektif, güçlü genelleme yeteneğine sahip ve ölçeklenebilir tahmin yapma kompleksliğine sahip modeller sunmaktadır. Türkçe dili kaliteli veri bulmanın zor olduğu düşük-kaynak dillerden biri olduğu için model geliştirilmesinde ColBERT paradigması kullanılmıştır. Model eğitimi için sadece açık veri kullanılmasına ve çok düşük ince ayar eğitim maliyeti olmasına rağmen, geliştirilen model çok dilli ColBERT varyantlarını başarı açısından geride bırakmıştır. Tez çalışmasında oluşturulan değerlendirme ve eğitim verisetlerinin

açık erişime sunulmasıyla Türkçe bilgi erişiminde ileride yapılacak çalışmalara katkıda bulunmak hedeflenmiştir.

Anahtar Kelimeler: Bilgi erişimi, makine öğrenmesi, metin kodlayıcıları

ABSTRACT

Development of A Colbert Model For Open Domain Information Retrieval In The Turkish Language

Eren TAHİR

Department of Data Science and Big Data

Master of Science Thesis

Supervisor: Assoc.Dr. Mert BAL

The thesis study aimed to develop and evaluate a contextual deep learning model for open-domain information retrieval in the Turkish language. Deep learning models in information retrieval are categorized into three types: interaction-based, representation-based, and hybrid. While interaction-based models promise high success, they require large amounts of training data and have high computational costs for inference. Representation-based models, although not data-efficient, stand out with their low inference complexity. Both representation-based and interaction-based models suffer significant performance loss when applied to data distributions different from their training data. One hybrid approach, the ColBERT paradigm, combines the strengths of these two methods to offer data-efficient models with strong generalization capabilities and scalable inference complexity. Since Turkish is a low-resource language with limited high-quality data available, the ColBERT paradigm was utilized for model development. Despite using only open data for training and requiring very low fine-tuning costs, the developed model outperformed multilingual ColBERT variants in terms of success. By making the evaluation and training datasets publicly accessible, the thesis study aims to contribute to future research in Turkish information retrieval.

Keywords: Information retrieval, machine learning, text encoders

1
GİRİŞ

1.1 Motivasyon

İnternetin ve dijital verinin hızlı bir şekilde büyümesiyle birlikte, bilgiye etkili, hızlı ve doğru biçimde erişim, hem kullanıcılar hem de kurumlar için kritik bir ihtiyaç haline gelmiştir. Bilgi erişim (Information Retrieval - IR) sistemleri, geniş veri yığınları içerisinde kullanıcıların ihtiyaç duyduğu bilgileri bulmalarını sağlayan temel teknolojilerdir. Bugün, arama motorlarından kurumsal doküman yönetim sistemlerine kadar pek çok uygulama, IR sistemlerinin gelişmiş algoritmalarına dayanmaktadır. Ancak bu sistemlerin başarısı, büyük ölçüde kullanılan dilin yapısal özelliklerine, kaynak verilerin doğasına ve sistemin tasarımına bağlıdır.

Bilgi erişim sistemleri, özellikle Türkçe gibi eklemeli morfolojiye sahip dillerde daha farklı zorluklarla karşılaşmaktadır. Türkçede sözcüklerin kökleri üzerine getirilen eklerle çok sayıda türev oluşturulabilmesi, klasik IR yöntemlerinin doğrudan uygulanmasını zorlaştırmaktadır. Bu durum, hem belge temsilinde hem de sorgu-genişletme tekniklerinde özel dil işleme adımlarını gerekli kılmaktadır. Öte yandan, bilgiye erişim alanı son yıllarda önemli bir dönüşüm sürecinden geçmektedir. Geleneksel anahtar kelime tabanlı IR modellerinden, doğal dil işleme (Natural Language Processing - NLP) ve derin öğrenme tabanlı yöntemlere doğru bir geçiş yaşanmaktadır. Bu değişim, öncelikle bilgiye erişim ile soru-cevap (Question Answering - QA) sistemleri arasındaki çizgiyi bulanıklaştırmış, daha sonra büyük dil modellerinin (Large Language Models - LLMs) sahneye çıkmasıyla birlikte yepyeni bir paradigma yaratmıştır. Artık kullanıcılar yalnızca aradıkları belgeleri değil, doğrudan doğru cevabı almak istemekte; sistemlerden doğal dilde açıklamalar, özetler ya da öneriler beklemektedir.

Bu dönüşümde LLM'lerin etkisi yadsınamazdır. OpenAI tarafından geliştirilen GPT serisi, Meta'nın LLaMA modelleri ve benzeri pek çok model, metni anlamlandırma, özetleme, soru yanıtlama gibi görevlerde etkileyici başarılar göstermiştir. Bununla birlikte, bu modellerin etkinliği büyük ölçüde İngilizce gibi yüksek kaynaklı dillere yöneliktir. Türkçe için geliştirilen LLM sayısı sınırlı olmakla birlikte, bu modellerin bilgi erişim bağlamında nasıl kullanılabileceği henüz araştırma aşamasındadır. Dolayısıyla, Türkçe için özel olarak geliştirilmiş bilgi erişim sistemlerinin hem klasik yöntemlerle hem de modern dil modelleriyle nasıl iyileştirilebileceği önemli bir araştırma sorusu olarak önümüzde durmaktadır.

Son yıllarda derin öğrenmenin doğal dil işleme alanında gösterdiği büyük ilerlemeler, bilgi erişim sistemlerini de derinden etkilemiştir. Geleneksel IR sistemleri çoğunlukla kelime düzeyinde benzerlik ölçümlerine ve istatistiksel yaklaşımlara (TF-IDF, BM25 gibi) dayanırken, nöral IR sistemleri metinleri semantik düzeyde temsil etmeyi ve benzerliği bu temsil alanında ölçmeyi amaçlamaktadır. Bu bağlamda, nöral IR modelleri genel olarak temsil tabanlı (representation-based) ve etkileşim tabanlı (interaction-based) olmak üzere iki ana yaklaşıma ayrılmaktadır.

Temsil tabanlı modeller, sorgu ve dokümanları birbirinden bağımsız olarak vektör uzayına gömmeyi ve bu vektörler arasındaki benzerliği ölçmeyi hedefler. Bu yaklaşımda tipik olarak encoder kullanılarak hem sorgu hem de belge birbirinden bağımsız olarak sabit uzunluklu vektöre dönüştürülür. Daha sonra bu vektörler arasındaki kosinüs benzerliği veya noktasal çarpım gibi bir skor fonksiyonu yardımıyla benzerlik puanı hesaplanır. Bu modellerin en büyük avantajı tahmin yapma açısından verimli olmalarıdır; çünkü tüm belgeler önceden gömülebilir ve bir vektör indeksi üzerinde hızlı arama yapılabilir. Ancak bu yöntem, doküman ve sorgu arasında kelime düzeyindeki ince etkileşimleri göz ardı ettiği için doğruluk açısından sınırlamalar barındırabilir. Türkçe gibi bağlamdan anlam kazanma eğiliminde olan dillerde bu temsilin doğruluğu özellikle önemlidir. Temsil tabanlı modeller eğitim için büyük miktarda veriye ihtiyaç duymaktadır ve eğitim verisinden farklı veri dağılımlarında düşük başarı göstermektedirler.

Etkileşim tabanlı modeller, sorgu ve dokümanı birlikte işler ve kelime/alt kelime düzeyinde karşılıklı etkileşimleri modeller. Bu yaklaşımda önce sorgu ve doküman embedding'leri çıkarılır, ardından bu embedding'ler üzerinde attention

mekanizmaları, benzerlik matrisleri veya etkileşim katmanları kullanılarak detaylı benzerlik hesaplamaları yapılır. Bu modeller genellikle daha yüksek doğruluk sağlar; çünkü bağlam ve terim-terim etkileşimleri daha iyi yakalanır. Ancak bu başarı, hesaplama maliyetinin artmasına yol açar. Her sorgu için dokümanların tekrar encode edilmesi gerektiği için, özellikle büyük veri setlerinde ölçeklenebilirlik sorunları yaşanabilir. Temsil tabanlı modeller gibi eğitim için büyük miktarda veriye ihtiyaç duymaktadırlar ve eğitim verisinden farklı veri dağılımlarında düşük başarı göstermektedir.

Bu iki yaklaşımı birleştirmeye çalışan hibrit modellerden biri ColBERT (Contextualized Late Interaction over BERT) [1] modelidir. ColBERT, temsil tabanlı bir yapı üzerinde çalışırken, aynı zamanda etkileşim tabanlı benzerlik hesaplamasını geç safhada (late interaction) yapar. Modelin temel prensibi, her token için ayrı bir bağlamsal embedding üretmek ve sorgu-doküman benzerliğini bu kelime embeddingleri üzerinden hesaplamaktır. Bu sayede, hem temsil tabanlı modellerin verimliliği korunur, hem de kelime düzeyinde daha ince detayların dikkate alındığı bir etkileşim sağlanır.

ColBERT, yoğun bilgi içeren uzun belgelerle çalışırken özellikle etkilidir. Aynı zamanda, önceden encode edilmiş belge embeddingleri ile hızlı arama yapılabildiğinden, sistemin gerçek zamanlı kullanım açısından da avantajı yadsınamaz. Bununla birlikte, token düzeyinde hesaplanan bağlamsal benzerlikler sayesinde, modelin yalnızca eğitim aldığı veri kümesine değil, farklı konu ve tarzlara sahip alan dışı (out-of-domain) içeriklere karşı da daha güçlü bir genelleme yeteneği sergilediği gözlemlenmiştir. Bu genelleme özelliği, çeşitli alanlardan gelen çok dilli veya çok tarzlı veriyle çalışan uygulamalar için büyük önem taşımaktadır.

Türkçe gibi kaynakları görece sınırlı dillerde, bilgi erişim sistemlerinin eğitildiği veri kümesinden farklı alanlarda da başarılı olabilmesi, yani alan dışı genelleme yeteneği kritik bir özelliktir. Gerçek dünyada kullanıcılar yalnızca eğitim verilerinde karşılaşılan konularda değil, çok çeşitli ve beklenmedik alanlarda bilgiye ihtiyaç duyarlar.

Türkçe özelinde alan dışı başarıyı zorlaştıran temel etkenler arasında şunlar yer almaktadır:

- Türkçe'nin eklemeli yapısı ve bağlama duyarlılığı, dil modelinin yalnızca kelime eşleşmelerine değil, bağlamsal anlamlara da hâkim olmasını gerektirir.
- Mevcut Türkçe IR veri setlerinin sınırlılığı, modelin farklı konu alanlarında eğitilmesini ve genelleme yapabilmesini güçleştirir.
- Özellikle haber, sosyal medya, resmi belgeler gibi farklı dil stillerine sahip alanlar arasında tarz ve sözdizim farkları, modelin uyum sağlamasını zorlaştırır.

Bu bağlamda geliştirilen Türkçe IR modellerinin yalnızca belirli bir veri kümesinde değil, farklı alanlardaki içeriklerde de etkili performans göstermesi gereklidir. Bu durum, hem kullanıcı memnuniyeti açısından hem de modelin yeniden eğitilmeden farklı sistemlerde kullanılabilmesi açısından önemlidir. Değinilen noktalar ele alındığında Türkçe bilgi erişim modeli geliştirilmesi için ColBERT paradigması öne çıkmaktadır.

1.2 Tezin Amacı ve Özgün Katkısı

Bu tez çalışmasının temel amacı, Türkçe dili için modern, etkili ve esnek bir bilgi erişim modeli geliştirmek üzere ColBERT mimarisinin uygulanabilirliğini araştırmaktır. Bu kapsamda, öncelikle Türkçeye özgü morfolojik yapı ve bağlam ilişkileri göz önünde bulundurularak ColBERT modelinin Türkçe metinler üzerinde eğitilmesi hedeflenmiştir. Türkçe için optimize edilmiş bir ColBERT modeli, hem temsili hem de etkileşimi bir arada kullanan yapısıyla, klasik IR yöntemlerine kıyasla daha yüksek doğruluk ve daha güçlü bağlamsal eşleşme performansı sunma potansiyeline sahiptir. Ayrıca modelin farklı konu alanlarında gösterdiği açık uçlu erişim başarısı, Türkçe'de genel amaçlı bilgi erişim sistemlerinin oluşturulmasında önemli bir adım niteliğindedir.

Tez kapsamında yalnızca modelin eğitimi değil, aynı zamanda değerlendirilmesi için uygun Türkçe veri kümelerinin oluşturulması da amaçlanmıştır. Türkçe'de bilgi erişim alanında standart, açık ve kapsamlı değerlendirme veri setlerinin eksikliği göz önünde bulundurularak, modelin performansını anlamlı bir şekilde ölçebilecek bir test altyapısı geliştirilmiştir. Geliştirilen model diğer çok dilli ColBERT varyantlarıyla efektiflik ve verimlilik kapsamında karşılaştırılmıştır. Yapılan deneylerde çok düşük maliyetlerle, ticari modeller ile rekabet edebilen Türkçe

spesifik modellerin geliştirilebileceği gösterilmiştir. Kullanılan yöntem ve yaklaşımların diğer düşük kaynak dillere uygulanabilirliği NLP alanına değerli katkılar sunmaktadır.

Bu çalışma, hem ColBERT'in açık set Türkçe IR'ye adaptasyonunun ilk adımını atmakta hem de ileride yapılacak IR araştırmaları için yeniden kullanılabilir veri ve yöntem temeli sunarak, alanın gelişimine katkı sağlamayı hedeflemektedir.

2
HAZIRLIK

2.1 Bilgi Erişimi

Bilgi erişimi, yapılandırılmamış veya yarı yapılandırılmış bilgi koleksiyonlarından ilgili verileri verimli bir şekilde çıkarma sürecidir. Bu sistemler, kullanıcıların bilgi ihtiyaçlarıyla uyumlu içerikleri bulmalarına yardımcı olur. Bilgi erişimi, arama motorlarını günlük olarak kullanan milyarlarca kişi için temel bir araçtır. Çeşitli modeller, algoritmalar ve vektör araması gibi karmaşık teknikler kullanarak IR sistemleri, belgeler, gömülü içerik, meta veriler, resimler, videolar ve ses dosyaları içeren veritabanları dahil olmak üzere sürekli genişleyen bir kaynak yelpazesine erişim sağlar.

2.1.1 Tarihsel Gelişimi

Bilgi erişimin kökenleri, kütüphanelerin ve arşivlerin bilgiyi sistematik olarak düzenlemek ve depolamak için kurulduğu, dizinleme ve alfabetik sıralamanın kullanıldığı eski zamanlara dayanır. 19. yüzyılda delgeç kartlarının veri işleme yöntemi olarak tanıtılması ve 1931'de Emanuel Goldberg”in film şeridinde depolanan bilgileri aramak için elektromekanik belge alma sistemi olan İstatistiksel Makine’nin patentini alması dikkate değer tarihi IR gelişmeleridir.

Bilgi erişimi, 20. yüzyılın ortalarında, bilgisayardaki gelişmelerin yönlendirdiği bilimsel bir disiplin haline geldi. Gerard Salton ve Hans Peter Luhn gibi ilk öncüler, otomatik belge alma için temel modeller geliştirdiler. Salton'ın Cornell Üniversitesi'ndeki ekibi, terim-belge matrisi, vektör uzay modeli, ilişki geri bildirimi ve Rocchio sınıflandırması gibi yeniliklerle büyük bir atılım olarak SMART Bilgi Erişim Sistemini 1960'larda yarattı. 1970'lere gelindiğinde, olasılıksal IR modelleri ve vektör tabanlı işleme çerçeveleri IR tekniklerini daha da

geliştirdi. 1990'ların sonlarında arama motorlarının yükselişiyle birlikte IR teknolojisi akademik ve kurumsal kullanımın ötesine geçerek küresel ölçekte bilgiye erişim için temel bir araç haline geldi.

2.1.2 Önemi

Günümüzün Bilgi Çağı'nda her saniye muazzam miktarda veri üretilir. İlgili bilgilere erişmenin etkili bir yolu olmadan, bu verilerin çoğu kullanılamaz hale gelir. Bilgi erişim sistemleri, kullanıcıların bu karmaşıklıkta gezinmesine yardımcı olur ve sürekli artan bilgi selinin ortasında en ilgili verileri filtreleyerek onlara sunar. IR, karar alma, araştırma ve bilgi keşfini geliştirerek sektörler arasında önemli bir rol oynar. Bu sistemler kişisel kullanımdan büyük ölçekli kurumsal uygulamalara kadar, yerel dosyaları aramaktan küresel araştırmalar yürütmeye kadar çeşitli süreçleri destekler. Genomik, spam filtreleme, akademi, e-ticaret, sağlık ve savunma gibi çeşitli alanlarda yoğun olarak kullanılmaktadır. Arama motorları, doğru sonuçlar sağlamak için IR modellerinden yararlanır. E-ticaret platformları, kullanıcı tercihlerine göre ürün önermek için bu sistemleri kullanır. Dijital kütüphaneler, araştırmacıların değerli kaynaklar bulmasına yardımcı olmak için IR tekniklerini uygular. Sağlık hizmetlerinde, IR sistemleri hasta kayıtları, tıbbi çalışmalar ve tedavi kılavuzları için yapılan aramaları kolaylaştırır. Benzer şekilde, hukukçular ilgili dava emsallerini verimli bir şekilde bulmak için IR teknolojisine güvenir. Veriyi erişilebilir ve anlamlı hale getireren IR, modern yaşamın vazgeçilmez bir aracıdır.

2.1.3 Faydaları

- **Verimli bilgi erişimi:** Her şeyden önce, IR sistemleri insanlara anlatılamayacak miktarda zaman kazandırır. Bilgi alma, kullanıcıların büyük miktarda belge ve veri arasında manuel olarak arama yapmadan ilgili bilgilere hızla erişmesini sağlar.
- **Bilgi keşfi:** Bilgi erişimi, verileri anlamlandırmamızı sağlayan güçlü bir araçtır. IR ile kullanıcılar, başlangıçta belirgin olmayabilecek verilerdeki eğilimleri, kalıpları ve ilişkileri belirleyebilir.
- **Kişiselleştirme:** Bazı IR sistemleri, sonuçları bireysel kullanıcıların tercihlerine ve davranışlarına göre anlamlı bir şekilde uyarlayabilir.

- **Karar desteği:** Profesyoneller, ihtiyaç duyduklarında en alakalı bilgilere erişerek bilinçli kararlar alma yetkisine sahip olur.

2.1.4 Nasıl Çalışır?

Bilgi erişim süreci, bir kullanıcı bilgi ihtiyaçlarını belirten bir sorgu gönderdiğinde başlar. Bir sorgu girildikten sonra, sistem sorguyu anlamlandırmak ve sorgudan öznitelik çıkarmak için sorguyu analiz eder. Elde edilen sorgu temsili bilgi tabanındaki doküman temsilleriyle karşılaştırılır. Tam eşleşmeler her zaman mümkün olmadığından, sistem en alakalı sonuçları sıralanmış bir listede sunar. Bu sıralama sistemi, IR arama yöntemlerini, alaka düzeyine dayalı erişim yerine katı sorgu eşleştirmesine dayanan geleneksel veritabanı aramalarından ayırır. IR sistemi ses, resim, metin gibi çeşitli modalitelerdeki dokümanları hızlı karşılaştırma yapabilmek için önceden indeksleyebilir. İndeksleme işlemi bilgi erişim başarısını feda ederek hızlı arama yapılmasını sağlar.

2.1.5 Bilgi Erişim Sistemi Bileşenleri

Bir bilgi alma sistemi birkaç temel bileşenden oluşur:

- **Belge koleksiyonu:** Sistemin bilgi alabileceği veri kümesi. Veri kümesi çeşitli yapılandırılmış, yapılandırılmamış ve yarı yapılandırılmış kaynakları barındırabilir.
- **İndeksleme bileşeni:** Belge koleksiyonunda arama yapılabilmesi için belgelerin temsilleri indekslenir. Arama indeksi bilgi çıkarmak için tasarlanmış bir yazılım bileşenidir. Büyük miktarda veriyi verimli bir şekilde işleyerek arama sürecini kolaylaştırır, hızlı ve kesin sonuçlar sağlar. Verileri dizinleme yoluyla düzenleyerek erişilebilirliği ve arama hızını artırır. İndeksleme yöntemi terim veya vektör tabanlı olabilir. Genellikle özel, optimize edilmiş veri yapıları kullanılır.
- **Sorgu işlemcisi:** Sorgu işlemcisi, kullanıcı sorgularını ve anahtar sözcükleri analiz eder ve bunları dizinlenmiş belge koleksiyonuyla eşleştirmek üzere hazırlar.
- **Sıralama algoritması:** Sıralama algoritması, belgelerin bir sorguyla alakalı olup olmadığını belirler ve bunlara puanlar atar.
- **Kullanıcı arayüzü:** Kullanıcıların sistemle etkileşime girdiği, sorgu gönderdiği ve sonuçların sunulduğu ekrandır. Burada, sonuçlar bir kullanıcı

sorgusuna ne kadar iyi hizmet ettiklerine göre ayarlanabilir. Bazı durumlarda, mekanizmalar kullanıcıların getirilen belgelerin alakalılığı hakkında geri bildirim sağlamasına izin verebilir ve bu geri bildirimler gelecekteki modelleri iyileştirmek için kullanılabilir.

2.1.6 Bilgi Erişimi Model Türleri

Bilgi erişim modelleri çeşitli zorlukları ele almak ve ilgili bilgileri bulma sürecini optimize etmek için tasarlanmıştır. Bu modeller üç ana kategoriye ayrılır: alan için temel oluşturan klasik modeller; geleneksel yaklaşımları geliştiren klasik olmayan modeller; ve makine öğrenimi ve doğal dil işleme gibi gelişmiş teknolojileri entegre eden alternatif modeller.

Yaygın olarak kullanılan bazı IR modeli türleri şunlardır:

- **Boolean Modeli:** En eski IR modelleri arasında, sorgu koşullarıyla eşleşen belgeleri filtrelemek için Boolean mantık operatörlerine (AND, OR, NOT) dayanır. Tam eşleşmeler için etkili olsa da, sıralama yeteneklerinden yoksundur ve kısmi eşleşme içeren sonuçlar sunamaz.
- **Vektör Uzay Modeli:** Belgeler ve sorgular, her boyutun benzersiz bir terime karşılık geldiği çok boyutlu bir uzayda vektörler olarak temsil edilir. İlgililik, sorgu ve belge vektörleri arasındaki kosinüs benzerliğinin hesaplanmasıyla belirlenir. Bu model, sıralanmış sonuçları mümkün kılar ve Boolean modelinin sınırlarını aşar.
- **Olasılıksal Model:** Terim sıklığı ve belge uzunluğu gibi faktörleri göz önünde bulundurarak bir belgenin belirli bir sorguyla alakalı olma olasılığını tahmin eder. Bu model, büyük veri kümelerini işlemek için oldukça uygundur ve ağırlıklı olasılıkları kullanarak sıralanmış sonuçlar üretir.
- **Latent Semantic Indexing (LSI):** Terimler ve belgeler arasındaki anlamsal ilişkileri ortaya çıkarmak için tekil değer ayrıştırmayı (Singular Value Decomposition - SVD) kullanır. LSI, tam terim eşleşmesinden ziyade niyet ve bağlamı analiz ederek kavramsal olarak ilişkili içeriğe erişimi iyileştirir.
- **Okapi BM25:** Olasılıksal modelden türetilen bir sıralama yöntemi olan BM25, sorgu terimlerinin varlığına dayanarak belgelerin bir arama sorgusuyla ilişkisini değerlendirir. Terimlerin belge koleksiyonundaki

bulunma frekansını ve belge uzunluğunu kullanarak olasılıksal modeli geliştirir. Arama motorlarında sonuç sıralamasını optimize etmek için yaygın olarak kullanılır.

2.1.7 Zorluklar ve Kısıtlamalar

Alandaki ilerlemelere rağmen, bilgi alma alanı hala birkaç zorlukla karşı karşıyadır:

- **Belirsizlik:** Doğal dil belirsiz olabilir ve bu da IR sistemlerinin kullanıcı sorgularını doğru bir şekilde yorumlamasını zorlaştırır.
- **İlgililik:** Neyin ilişkili olduğunu belirlemek, kullanıcı amacına ve bağlama göre değişir. İlişkiyi değerlendirmek için genel standartlar her zaman bir bireyin özel ihtiyaçlarıyla uyuşmayabilir.
- **Anlamsal boşluklar:** Metinsel temsil ile insan anlayışı arasındaki uçurum nedeniyle, erişim sistemleri genellikle içeriğin ardındaki daha derin anlamı tam olarak yakalamakta zorlanır. Yapay zeka destekli doğal dil işleme bu boşlukları kapatmaya yardımcı olur, ancak zorluklar tamamen çözülebilmiş değildir.
- **Ölçeklenebilirlik:** Veri hacimleri arttıkça, verimli dizinleme ve bilgi erişimi sağlamak giderek daha fazla kaynak ve bilgi işlem gücü gerektirir.

Bu sınırlamalara rağmen, yapay zeka ve makine öğrenimindeki devam eden gelişmeler, daha doğru ve anlamlı sonuçlar için bilgi erişim alanını iyileştirmeye devam etmektedir.

2.2 Doğal Dil İşleme

Doğal dil işleme, makineler tarafından insan dilinin anlaşılmasını, işlenmesini ve üretilmesini kolaylaştırmak için bilgisayar bilimi, yapay zeka ve dil bilimini birleştiren disiplinler arası bir alandır. Metin tabanlı verilerin hacmi, özellikle sosyal medya, web siteleri ve dijital platformlar gibi kaynaklarla artmaya devam ettikçe, NLP'nin içgörüler çıkarmak ve dil ile ilgili görevleri otomatikleştirmek işlevinin değeri artmıştır.

NLP, otomatik çeviri sistemleri, konuşma tanıma yazılımları, metin özetleme teknikleri ve sohbet asistanları dahil olmak üzere dil işlemeye dayanan çok çeşitli uygulamaların temelini oluşturur. Birçok kişi, sesle desteklenen navigasyon sistemleri, dijital asistanlar, konuşmadan metne dönüştürücüler ve otomatik müşteri

hizmetleri sohbet robotları gibi NLP destekli teknolojilerle günlük olarak etkileşime girmektedir. Dahası, işletmeler karmaşık dil süreçlerini otomatikleştirerek verimliliği optimize etmek, üretkenliği artırmak ve operasyonları düzene sokmak için NLP'den giderek daha fazla yararlanmaktadır.

2.2.1 Metin Ön İşleme Teknikleri

- **Tokenizasyon:** Doğal dil işlemede bir metnin token olarak bilinen ayrı birimlere bölündüğü önemli bir ön işleme adımıdır. Bu tokenler, belirli bir uygulama için gereken ayrıntı düzeyine bağlı olarak tek tek karakterlerden tam kelimelere veya çok kelimeli ifadelere kadar değişen bir ayrıntı düzeyinde olabilir. Tokenizasyon metinsel verileri bu daha küçük, yönetilebilir bileşenlere dönüştürerek, dilbilimsel analizi kolaylaştırır ve makinelerin insan dilini daha etkili bir şekilde işlemesini, yorumlamasını ve üretmesini sağlar
- **Kök bulma (Lemmatization):** Lemmatizasyon, doğal dil işlemede kelimelerin temel veya sözlük biçimlerine, yani lemma olarak bilinen biçimlerine dönüştürüldüğü önemli bir normalleştirme sürecidir. Sadece sezgisel kurallara dayalı olarak ekleri kaldıran gövdelemenin aksine, lemmatizasyon, çıkarılan temel biçimin dilin sözlüğünde geçerli bir sözcük olduğundan emin olmak için dil özgü kuralları ve bağlamsal analizi kullanır.
- **Gövdeleme (Stemming):** Önekleri ve sonekleri kaldırarak sözcükleri basitleştiren ve bunları gövde olarak bilinen ortak bir kök biçimine indirgeyen temel bir tekniktir. Bir sözcüğün dilbilimsel olarak doğru temel biçimini bulmaya çalışan lemmatizasyonun aksine, köklendirme, ekleri kesmek için sezgisel kurallar uygular ve genellikle dilde geçerli olmayan sözcüklerle sonuçlanır.
- **Stopword Removal:** Stop words, metinlerde yüksek frekansta kullanılan ancak taşıdıkları anlam bakımından sınırlı bilgi sağlayan kelimelerdir. Türkçede "ve", "veya", "de" ve "fakat" gibi sözcükler bu kategoriye örnek verilebilir. Bu kelimelerin metinden çıkarılması, doğal dil işleme süreçlerinde önemli bir rol oynayarak metin analizi ve makine öğrenimi modellerinin performansını artırmaktadır.
- **Diğer normalizasyon teknikleri:** Söz konusu doğal dil işleme problemine göre harflerin büyük veya küçük yazılmasının getirdiği bilgi ihmal edilecek

derecede az olabilir, bu durumlarda tüm harfler küçük harfe çevrilerek problem basitleştirilir. Benzer şekilde noktalama işaretleri, alakasız-istenmeyen metin parçaları veriden temizlenebilir. Yazım hatalarını gidermek de verideki gürültüyü azaltacağı için faydalı bir yöntemdir.

2.2.2 Trendler

Doğal dil işleme, son yıllarda büyük ilerlemeler kaydetti ve gelişmeye devam ediyor. NLP'deki en önemli trendlerden biri daha güçlü ve bağlamsal anlayışa sahip büyük dil modellerinin yaygınlaşmasıdır. Çok dilli modellerin ilerlemesi sayesinde NLP sistemleri, farklı dillerde daha doğru ve akıcı çeviriler yapabiliyor. Bunun yanı sıra, duygusal zeka ve bağlam farkındalığı NLP modellerine entegre ediliyor, böylece metinler yalnızca kelime bazında değil, içerik ve duygusal bağlam açısından da değerlendirilebiliyor. Özelleştirilmiş ve sektörel NLP çözümleri de yaygınlaşıyor; örneğin hukuk, tıp veya finans alanına özgü metinleri daha etkili yorumlayabilen özel modeller geliştiriliyor. Küçük ve optimize modeller, daha az veri ve hesaplama gücü ile yüksek performans sağlamak amacıyla geliştiriliyor ve bu da mobil cihazlar ve IoT sistemlerinde NLP'nin daha verimli kullanılmasına olanak tanıyor. Son olarak, etik ve adil yapay zeka konusu da NLP alanında büyük bir trend haline geldi; önyargıları azaltmak ve daha kapsayıcı modeller oluşturmak için birçok araştırma yapılıyor. NLP'nin geleceği, daha akıllı, daha duyarlı ve daha etik sistemler oluşturma yönünde hızla ilerlemektedir.

2.3 İndeksleme

Bilgi erişiminde indeksleme, büyük veri koleksiyonları içinde aranan bilginin hızlı ve verimli bir şekilde bulunabilmesini sağlamak amacıyla gerçekleştirilen temel bir işlemdir. Bu süreçte belgeler, içeriklerinde yer alan terimlere veya semantik temsillere göre yapısal biçimlere dönüştürülerek özel veri yapıları aracılığıyla organize edilir. Geleneksel bilgi erişim sistemlerinde bu temsil genellikle ters çevrilmiş dizin (inverted index) yoluyla gerçekleştirilir. Bu yapı, her bir terimin hangi belgelerde geçtiğini gösteren bir liste oluşturur; böylece kullanıcı tarafından yapılan sorgular, doğrudan ilgili belgelere erişmeyi mümkün kılar. Örneğin "yapay zeka" ifadesini içeren belgeler, bu ifadeyle ilişkili dizin girdisi sayesinde anında bulunabilir. Öte yandan, modern bilgi erişim sistemlerinde, metinler yalnızca terim temelli değil, aynı zamanda bağlamsal anlam taşıyan gömme (embedding)

vektörleri ile temsil edilir. Bu vektörler, Annoy, HNSW veya FAISS gibi yaklaşık en yakın komşu arama (Approximate Nearest Neighbor -ANN) algoritmalarını kullanan yapılarla indekslenir. Bu sayede, kullanıcının sorgusu yalnızca anahtar kelime eşleşmeleriyle değil, anlamca benzer içeriklerle de eşleştirilebilir. İndeksleme, bilgi erişim sistemlerinin temel yapı taşı olup, arama performansını ve doğruluğunu belirleyen en kritik bileşenlerden biridir.

2.4 ANN Algoritmaları

Approximate Nearest Neighbor algoritmaları, yüksek boyutlu veri uzaylarında belirli bir sorgu vektörüne en yakın veri noktalarını, tam doğruluk gerektirmeden ancak çok daha hızlı bir şekilde bulmayı amaçlayan yöntemler bütünüdür. Geleneksel en yakın komşu (exact nearest neighbor) araması, özellikle büyük ve yüksek boyutlu veri kümelerinde oldukça maliyetli ve zaman alıcı hale gelir. Bu durum, milyonlarca veya milyarlarca vektörden oluşan veri setlerinde gerçek zamanlı arama yapılmasını imkansızlaştırır. Approximate Nearest Neighbor algoritmaları, bu sorunu çözmek için doğruluk-performans dengesini gözeterek çalışır; yani arama sonuçları %100 doğru olmasa da kullanıcıya genellikle yeterince benzer sonuçlar çok daha kısa sürede sunulur. Bu nedenle ANN yöntemleri, semantik arama, görsel benzerlik tespiti, öneri sistemleri ve biyometrik tanıma gibi pek çok alanda yaygın biçimde kullanılmaktadır. HNSW, FAISS, ScaNN ve Annoy gibi modern ANN kütüphaneleri, bu algoritmaların optimize edilmiş ve ölçeklenebilir uygulamalarını sağlar. ANN arama, günümüzün büyük ölçekli ve anlam tabanlı bilgi sistemlerinin temel yapı taşlarından biri haline gelmiştir.

2.4.1 Annoy

Spotify tarafından geliştirilen Annoy (Approximate Nearest Neighbors Oh Yeah) [2], yüksek boyutlu vektör uzaylarında yaklaşık en yakın komşu (ANN) aramaları gerçekleştirmek üzere optimize edilmiş, açık kaynak kodlu bir C++ kütüphanesidir. Özellikle büyük ölçekli öneri sistemleri, bilgi erişimi, ve gömülü vektör tabanlı arama uygulamalarında kullanılmak üzere tasarlanmıştır. Python arayüzü ile de geniş kullanım alanı bulan Annoy, Spotify'ın müzik öneri sistemlerinde gerçek zamanlı kullanıcı taleplerine hızlı yanıt verebilmek amacıyla geliştirilmiştir.

Annoy, temel olarak ağaç tabanlı bir yapı kullanarak vektör uzayını parçalara ayırır. Her bir ağaç, veri kümesini ikili bölmeler (binary partitions) yoluyla bölerek bir tür K-D tree benzeri yapılar oluşturur. Ağaç sayısı artırıldıkça doğruluk artarken, daha az sayıda ağaç ile daha hızlı fakat daha yaklaşık sonuçlar elde edilir. Bu yaklaşım, sistemin hesaplama ve doğruluk arasındaki dengeyi (speed-accuracy trade-off) esnek bir şekilde yönetmesini sağlar. Annoy'un en dikkat çekici avantajlarından biri, veriyi belleğe yüklemeden disk üzerinden salt okunur (read-only) erişim imkânı sunmasıdır. Bu özellik, hafıza kısıtlı ortamlarda büyük veri kümeleriyle çalışmayı mümkün kılar.

Annoy, kosinüs benzerliği ve öklidyen mesafe gibi yaygın kullanılan mesafe metriklerini destekler. Ayrıca, her veri vektörü için sabit boyutta float32 formatında embeddingler kullanılır. Yapının yalnızca sabit boyutlu vektörlerle çalışması, kullanım esnekliğini sınırlasa da performans açısından belirli avantajlar sağlar. İndeksleme işlemi çevrimdışı olarak gerçekleştirildiğinden, sistem yalnızca sorgulama sırasında yüksek performans sunmakla kalmaz, aynı zamanda dağıtık ortamlarda önceden oluşturulmuş indekslerin paylaşılmasına da olanak tanır.

Spotify Annoy, özellikle ön belleklenebilir öneri sistemleri, benzer içerik aramaları, ve kullanıcı profili eşleştirmeleri gibi görevlerde yaygın olarak kullanılmaktadır. Alternatif olarak Facebook AI tarafından geliştirilen FAISS ve Google tarafından geliştirilen ScaNN gibi daha yeni sistemlere kıyasla bazı eksiklikler taşısa da (örneğin GPU desteği veya daha gelişmiş optimizasyon tekniklerinin eksikliği), hafif yapısı, kolay kullanımı, ve yüksek sorgu verimliliği ile hala birçok üretim sisteminde tercih edilen bir çözüm olmaya devam etmektedir.

2.4.2 HSNW

Hierarchical Navigable Small World (HNSW) [3] algoritması, yüksek boyutlu vektör uzaylarında ANN araması gerçekleştirmek üzere geliştirilmiş, günümüzdeki en etkili ve performanslı yöntemlerden biridir. Temelini, küçük dünya graf teorisi üzerine inşa eden HNSW, hem arama doğruluğu hem de sorgu süresi bakımından birçok geleneksel yapıya kıyasla üstün performans sergilemektedir. Algoritma ilk olarak Yury Malkov ve Dmitry Yashunin tarafından 2016 yılında önerilmiş olup, FAISS, NMSLIB ve Vespa gibi birçok modern ANN sisteminde yaygın olarak kullanılmaktadır.

HNSW, veri noktalarını çok seviyeli bir navigable small-world graph yapısı içerisinde organize eder. Bu yapı, her vektörü düğüm olarak ele alır ve benzer vektörler arasında bağlantılar oluşturarak bir ağ kurar. HNSW'nin temel farkı, bu grafı hiyerarşik seviyelerde oluşturarak arama sürecini hızlandırmasıdır. En üst seviyede sadece birkaç düğüm yer alırken, alt seviyelere indikçe daha fazla düğüm ve bağlantı içerir. Arama işlemi üst seviyelerden başlatılarak daha alt seviyelere doğru derinleştirilir ve bu sayede logaritmik karmaşıklığa yakın bir sorgulama performansı elde edilir.

Bu yöntem, tipik olarak önce kaba bir global arama, ardından yerel bir inceleme stratejisiyle çalışır. Her düğüm, yalnızca en yakın k komşusuna değil, aynı zamanda rastgele ve yapısal bağlantılarla seçilen diğer düğümlere de bağlanır. Bu yaklaşım, arama sırasında lokal minimumlara takılmadan daha doğru komşulara ulaşılmasını sağlar. HNSW algoritmasının arama süreci, beam search benzeri bir stratejiyle ilerler ve çok az sayıda adımda yüksek doğrulukta sonuçlara ulaşır.

HNSW, öklidyen mesafe, kosinüs benzerliği, Manhattan mesafesi gibi çeşitli uzaklık metriklerini destekleyebilir. Ayrıca, indeksleme işlemi sırasında ek bellek kullanımı ve bağlantı yoğunluğu parametreleri üzerinden özelleştirilebilir. Bu parametreler, sistemin inşa süresi, arama doğruluğu ve bellek kullanımı arasındaki dengeyi yönetmekte kritik rol oynar.

Diğer ANN yaklaşımlarına kıyasla HNSW, özellikle yüksek doğruluk ve hızlı sorgulama gerektiren senaryolarda öne çıkmaktadır. FAISS gibi kütüphanelerde GPU hızlandırmalı versiyonları da bulunmakta, bu sayede milyarlarca vektörlük veri kümelerinde dahi etkili bir şekilde çalışabilmektedir. Ancak, HNSW yapısı özellikle indeks boyutu açısından yüksek bellek kullanımı gerektirebilir, bu nedenle büyük veri kümelerinde donanım kaynakları dikkatle planlanmalıdır.

2.5 Büyük Dil Modelleri ve Transformer Mimarisi

Büyük dil modeller, doğal dil işleme alanında devrim yaratan yapay zeka sistemleridir. Milyarlarca trilyonlarca kelimelik metin verisi üzerinde eğitilen bu modeller, insan benzeri metin üretimi, özetleme, çeviri, soru-cevap ve daha birçok dil tabanlı görevde yüksek doğrulukla çalışabilmektedir. LLM'ler, makine

öğrenmesi teknikleriyle dildeki kalıpları, bağlamları ve anlam ilişkilerini kavrayarak metinleri anlamlı ve akıcı biçimde üretme yetisine sahiptir.

Bu modellerin başarısı, eğitim süreçlerindeki büyük veri setleri ve yüksek hesaplama gücüne dayanmaktadır. Ön eğitim uygulanmış LLM'ler, daha sonra özel görevler için ince ayar (fine-tuning) veya talimatla yönlendirme (instruction tuning) gibi tekniklerle özelleştirilebilir. Bu esneklik, LLM'leri farklı sektörlerde uygulanabilir hale getirir: sağlık alanında tanı destek sistemlerinden, hukukta metin inceleme araçlarına; eğitimde kişiselleştirilmiş öğrenmeden, müşteri hizmetlerinde sohbet robotlarına kadar pek çok alanda kullanılmaktadır. Ayrıca çok dilli kapasiteleri sayesinde küresel iletişimde de önemli bir rol üstlenmektedirler.

Ancak LLM'lerin sunduğu potansiyelin yanında bazı sınırlamaları ve etik sorunları da gündemdedir. Modellerin yanlış bilgi üretmesi, önyargıları yansıtması veya kişisel verileri istemeden açığa çıkarması gibi riskler bulunmaktadır. Bu nedenle, LLM'lerin geliştirilmesi ve uygulanması sürecinde şeffaflık, güvenlik ve sorumlu yapay zeka ilkeleri büyük önem taşımaktadır. Gelecekte bu modellerin daha güvenilir, verimli ve kullanıcı dostu hale gelmesi için hem teknik ilerlemeler hem de düzenleyici çerçevelerin eşgüdümlü olarak geliştirilmesi gerekmektedir.

2.5.1 Transformer Modeli

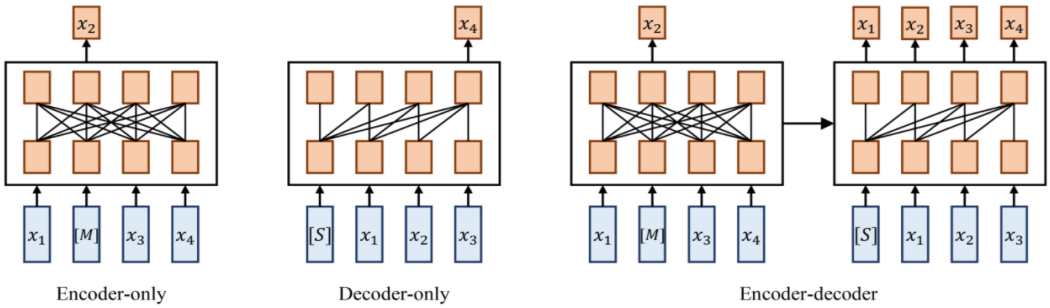
Transformer modeli, sıralı verileri verimli bir şekilde işlemek için tasarlanmış güçlü bir sinir ağı mimarisidir ve bu da onu büyük dil modellerinin temel bir bileşeni haline getirir. NLP'nin ötesinde, transformerlar bilgisayarlı görme, ses görevleri ve zaman serisi tahmini gibi yapay zeka uygulamalarında da çığır açan ilerlemeler elde etmiştir.

Başlangıçta Vaswani ve diğerleri tarafından 2017'de yayınlanan "Attention is All You Need" adlı makalede tanıtılan transformatör mimarisi [4], geleneksel yinelemeli sinir ağlarından (RNN'ler) uzaklaşarak derin öğrenmede devrim yaratmıştır. Başlangıçta makine çevirisini geliştirmek için geliştirilmiş olsa da, etkisi o zamandan beri çeşitli makine öğrenimi disiplinlerine yayılmıştır. Ancak, en dikkat çekici etkisi sohbet robotları, metin oluşturma, özetleme, soru cevaplama ve duygu analizi dahil olmak üzere NLP uygulamalarında devam etmektedir.

Transformatör tabanlı NLP modellerindeki önemli ilerlemelerden biri, Google tarafından 2019'da kelime gömme teknikleri için yeni bir standart belirleyen

BERT'in [5] tanıtılmasıydı. BERT arama motoru yeteneklerini önemli ölçüde iyileştirdi ve modern vektör üretimi için olmazsa olmaz olmaya devam ediyor. Daha sonra, OpenAI'ın ChatGPT'si gibi otoregresif decoder-only modeller, üretken yapay zekanın yükselişinde önemli bir rol oynayarak, yapay zeka tabanlı içerik oluşturmayı şekillendirdi.

Transformer modelleri, bir giriş dizisi içindeki bağımlılıkları etkili bir şekilde tanımlamalarına olanak tanıyan self-attention mekanizması ile öne çıkar. RNN'ler ve CNN'ler gibi daha önceki mimarilerin aksine, transformerlar yalnızca attention katmanlarını ve standart ileri besleme katmanlarını kullanır ve bu da ardışık veri işleme ihtiyacını ortadan kaldırır. Özellikle multi head attention yoluyla, sekanslarla ilgili görevlerde RNN'leri ve CNN'leri geride bırakır. Tekrarlayan sinir ağlarına (RNN) dayanan geleneksel NLP modelleri, girdiyi ardışık olarak işleyerek uzun menzilli bağımlılıkları iyi anlamlandıramaz. Uzun kısa süreli bellek ağları (LSTM'ler), RNN'leri daha uzun dizileri işleme konusunda geliştirse de, yine de temel sınırlamalarla karşı karşıyadırlar. Bunun aksine, transformerlardaki attention mekanizmaları tüm dizileri aynı anda analiz edebilir ve verilerin önemli bölümlerine nerede ve nasıl odaklanılacağını belirleyebilir. Bu yetenek yalnızca karmaşık bağımlılıkların anlaşılmasını geliştirmekle kalmaz, aynı zamanda paralel işlemeyi de mümkün kılar ve transformerların eğitim ve çıkarım sırasında GPU'ların tüm hesaplama gücünden yararlanmasını sağlar. Paralellik yoluyla kazanılan verimlilik, dönüştürücülerin kendi kendini denetleyen öğrenme (self supervised learning) kullanılarak büyük veri kümeleri üzerinde eğitilmesini mümkün kılarak modern yapay zeka uygulamalarındaki başarılarını artırır.



Şekil 2.1 Transformer model türleri

2.5.2 Encoder-Only Transformer Modelleri

Encoder-only modeller, doğal dil işleme görevleri için güçlü anlama ve metinsel veri temsili sağlayan özel derin öğrenme mimarileridir. Metin sınıflandırma, named

entity recognition, semantik arama, anomali tespiti ve embedding üretimi gibi uygulamalarda üstün performans gösterirler. Girdi dizilerini çok katmanlı self-attention mekanizması ile işleyerek, metindeki semantik ve sözdizimsel ilişkileri yakalayan zengin vektör temsilleri oluştururlar. Encoder-decoder modeller, çeviri ve özetleme gibi üretken görevler için kullanılırken, encoder-only modeller yalnızca bilgiyi analiz etmeye ve kodlamaya odaklanır; yeni içerik üretmezler. Bu özellikleri, metin üretiminin değil, doğru anlama ve sınıflandırmanın daha önemli olduğu görevlerde modeli son derece etkili hale getirir.

Encoder-only modellerin tarihsel gelişimi, sinir ağı mimarileri ve dil modelleme alanındaki temel ilerlemelere dayanır. Erken dönem modeller istatistiksel yöntemlere dayanıyordu, ancak derin öğrenmenin ortaya çıkışı bağlamsal anlayışı önemli ölçüde geliştirdi. Recurrent neural networks (RNNs) ve long short-term memory (LSTM) ağları yaygın olarak kullanıldı, ancak uzun vadeli bağımlılıklarla ve paralelleştirme ile ilgili zorluklar yaşadılar. 2017 yılında Vaswani ve ekibi tarafından tanıtılan transformer mimarisi, self-attention mekanizması ile daha verimli ve ölçeklenebilir modeller sağlayarak alanı kökten değiştirdi. Encoder-only modeller, özellikle BERT (Bidirectional Encoder Representations from Transformers), çift yönlü dikkat mekanizmalarını kullanarak metinsel anlamı kavramada güçlü araçlar haline geldi ve birçok NLP görevinde önceki modelleri geride bıraktı. Daha sonraki gelişmeler, RoBERTa ve DeBERTa, pretraining stratejilerini ve attention mekanizmasınıı optimize ederek yaklaşımı daha da geliştirdi.

Encoder-only modeller için üç adet ön eğitim yöntemi öne çıkmaktadır:

- **NSP:** Next Sentence Prediction (NSP) modellerin cümle ilişkilerini anlamasına yardımcı olmak için kullanılan bir ön eğitim görevidir. Modelin iki cümle almasını ve ikinci cümlenin birinciyi mantıksal olarak takip edip etmediğini belirlemesini gerektirir. Bu görev, soru cevaplama ve doğal dil çıkarımı gibi metin düzeyinde anlamlandırma gerektiren uygulamalarda performansı artırmaya yardımcı olur.
- **MLM:** Masked Language Modeling (MLM) yötneminde, bir cümledeki belirli tokenler rastgele özel bir [MASK] tokeni ile değiştirilir. Model, çevresindeki bağlama dayanarak maskelenmiş kelimeleri tahmin etmeyi öğrenir ve böylece dil yapısını derinlemesine anlamasını sağlar.

- **TLM:** Translation Language Modeling (TLM), çok dilli öğrenme için tasarlanmış Maskelenmiş Dil Modellemesi (MLM) uzantısıdır. Bu yöntem, iki farklı dildeki paralel cümleleri birleştirerek belirteçleri rastgele maskeleme işlemiyle çalışır. Model, maskelenmiş kelimeleri tahmin ederken hem bir dildeki çevresel bağlamı hem de diğer dildeki karşılık gelen çeviriyi kullanır.

RoBERTa [6], yani Robustly Optimized BERT Yaklaşımı, Facebook AI araştırmacıları tarafından geliştirilen BERT modelinin ileri bir versiyonudur. BERT'in temel mimarisini korurken, RoBERTa performansı artırmaya yönelik çeşitli iyileştirmeler sunar. En dikkat çekici değişikliklerden biri, NSP görevini kaldırması ve tamamen maskelenmiş dil modellemeye odaklanmasıdır. Ayrıca, daha büyük veri grupları, uzatılmış eğitim dizileri ve dinamik maskeleme gibi teknikler kullanarak daha zengin metin temsilleri oluşturma yeteneğini güçlendirir.

Buna ek olarak, RoBERTa 160GB'lık devasa bir metin veri kümesi üzerinde eğitilmiştir, yani BERT'in eğitim aldığı verinin on katından fazla bir veri setinden faydalanmaktadır. Bu optimizasyonlar sayesinde RoBERTa, doğal dil işleme alanındaki birçok ölçüt açısından BERT'i geride bırakmış ve metin sınıflandırma, soru yanıtlama ve duygu analizi gibi görevlerde oldukça değerli bir araç haline gelmiştir.

RoBERTa'daki dinamik maskeleme, BERT'in orijinal eğitim yöntemine kıyasla önemli bir iyileştirmedir. BERT, statik maskeleme kullanarak her eğitim turunda aynı tokenleri maskelerken, RoBERTa her eğitim turunda farklı tokenleri maskeleyerek modelin daha çeşitli bağlamsal temsiller öğrenmesini sağlar. Bu yaklaşım, modelin doğal dil işleme görevlerinde genelleme yapma yeteneğini geliştirmeye yardımcı olur. Maskelenmiş tokenleri sürekli değiştiren RoBERTa, kelime ilişkilerini ve cümle yapılarını daha iyi kavrayarak metin sınıflandırma, soru yanıtlama ve duygu analizi gibi uygulamalar için daha sağlam bir model haline gelir.

XLM-RoBERTa [7], RoBERTa'nın çok dilli bir varyantıdır ve birden fazla dilde metin işleyebilmek için tasarlanmıştır. Facebook AI tarafından geliştirilen bu model, RoBERTa'nın mimarisini temel alır ancak 100 dili kapsayan 2.5TB boyutunda filtrelenmiş CommonCrawl verisi üzerinde eğitilmiştir. Bazı çok dilli

modellerin aksine, XLM-RoBERTa dil spesifik tokenler kullanmaz, bu da farklı dillerde daha iyi genelleme yapmasını sağlar. Ana eğitim hedefi olarak maskeleme tabanlı dil modelleme yöntemini kullanarak, duygu analizi ve metin sınıflandırma gibi çok dilli görevlerde üstün performans gösterir. BERT'in çok dilli versiyonu mBERT ile kıyaslandığında, XLM-RoBERTa XNLI [8] ve MLQA [9] gibi kıyaslama testlerinde önemli doğruluk artışı sağlayarak çok dilli doğal dil işleme alanında güçlü bir araç hâline gelmiştir.

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) [10], BERT ve RoBERTa'nın gelişmiş bir evrimidir ve Microsoft tarafından geliştirilmiştir. İki önemli yenilik içerir: kelime içeriği ile konum kodlamasını ayırt eden ayrıştırılmış attention mekanizması ve maskeleme tabanlı token tahminini iyileştiren gelişmiş token sınıflandırıcısı. Bu iyileştirmeler, DeBERTa'nın RoBERTa'yı çeşitli doğal dil işleme görevlerinde geride bırakmasını sağlar ve bunu yaparken yalnızca yarı miktarda eğitim verisine ihtiyaç duyar.

RoFormer [11], self-attention mekanizmalarında konumsal kodlamayı iyileştirmek için Rotary Position Embedding (RoPE) kullanan geliştirilmiş bir transformer modelidir. Bağımlılık modellemesini optimize etmek amacıyla araştırmacılar tarafından geliştirilen RoFormer, hem mutlak hem de göreceli konum bilgilerini bir döndürme matrisi ile kodlar. Bu yaklaşım, daha esnek dizi uzunluklarına olanak tanır ve uzun metinleri daha verimli şekilde işlemesini sağlar.

Rotary Position Embedding, hem mutlak hem de göreli konum bağımlılıklarını bir döndürme matrisi aracılığıyla entegre ederek transformer modellerinde konumsal kodlamayı önemli ölçüde iyileştirir. Bu yöntem, modelin uzun dizileri verimli bir şekilde işlemesini sağlar ve derin bağlamsal anlayış gerektiren görevler için özellikle faydalıdır. RoPE'nin en büyük avantajlarından biri, eğitim sırasında karşılaşılanlardan daha uzun dizilere genelleme yapabilmesi ve genişletilmiş metin girdilerinde performansı artırmasıdır. Bununla birlikte, RoPE'nin bazı sınırlamaları vardır karmaşık matematiksel dönüşümlere olan bağımlılığı hesaplama yükünü artırabilir ve doğal dil işleme görevlerine bağlı olarak etkinliği değişebilir. Bu zorluklara rağmen, RoPE modern dil modellerinde attention mekanizmalarını iyileştirmek için güçlü bir araç olmaya devam etmektedir.

ModernBERT [12], doğal dil işleme alanında verimliliği ve performansı artırmak için tasarlanmış, son teknoloji bir encoder-only modeldir. Daha hızlı, daha az bellek kullanan ve daha uzun dizileri işleyebilen çeşitli mimari iyileştirmeler içerir. Öne çıkan özelliklerinden biri, geleneksel BERT modellerine kıyasla çok daha büyük metin girdilerini işleyebilmesini sağlayan 8192 tokenlik bağlam uzunluğudur. Ayrıca, standart MLP katmanlarını GeGLU katmanları ile değiştirerek hem hesaplama verimliliğini korur hem de performansı artırır. Uzun dizilerde bağlamı daha iyi yakalamak için RoPE kullanır.

ModernBERT'in sunduğu diğer yeniliklerden biri de Flash Attention 2 olup, modern GPU'larda daha hızlı eğitim ve çıkarım için attention mekanizmalarını optimize etmesidir. Ayrıca alternating attention mekanizması kullanarak global ve yerel attention'ı birleştirir ve verimlilik ile bağlamsal derinlik arasında denge kurar. Bu yaklaşım, bellek verimliliğini artırır ve işlem hızını yükselterek ModernBERT'i özellikle uzun metin uygulamaları için son derece etkili hâle getirir. 2 trilyon token üzerinde eğitilmiş olan ModernBERT, kod ve matematiksel metinler gibi çeşitli veri kümelerini içeren geniş kapsamlı bir eğitim süreci geçirerek birçok doğal dil işleme kıyaslama testinde en iyi sonuçları elde etmiştir.

2.6 Nöral Bilgi Erişim Paradigmaları

Bilgi erişimi alanının evrimi, derin öğrenmenin ortaya çıkışıyla birlikte büyük bir dönüşüm geçirmiştir ve bu gelişme, nöral bilgi getirme (neural information retrieval) olarak bilinen bir yöntem sınıfının doğmasına yol açmıştır. Geleneksel IR yöntemleri büyük ölçüde sembolik terim eşleştirme ve elle tasarlanmış özniteliklere dayanırken, nöral IR paradigmaları daha derin anlamsal ilişkileri yakalamak için vektör tabanlı temsilleri ve sinir ağı mimarilerini kullanır.

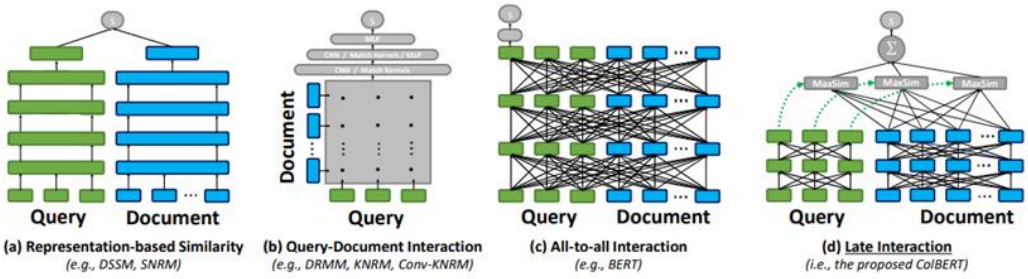
Nöral IR'nin temelinde, sorguların ve belgelerin nasıl modellendiği ve eşleştirildiği konusunda farklı bakış açıları sunan üç temel paradigma yer alır:

- **Temsil Tabanlı Modeller (Representation-Based Models):** Bu modeller, sorgular ve belgeler için bağımsız olarak yoğun vektör temsilleri öğrenmeyi hedefler. Daha sonra bu vektörler, kosinüs benzerliği veya iç çarpım gibi benzerlik ölçütleri kullanılarak karşılaştırılır. Bu paradigma, erişim verimliliği ve ölçeklenebilirlik üzerinde durur. Örnekler arasında DSSM [13], siyam sinir ağları tabanlı erişim yöntemleri yer alır.

- **Etkileşim Tabanlı Modeller (Interaction-Based Models):** Bu modeller, sorgular ve belgeler arasındaki doğrudan etkileşimleri (genellikle benzerlik matrisleri aracılığıyla) modelleyerek sözcük düzeyindeki ayrıntıyı ele alır. CNN'ler veya transformer'lar gibi sinirsel katmanlar kullanılarak ince ayrıntılı benzerlik sinyalleri yakalanır. Bu yaklaşım, birçok sıralama görevinde üstün performans sağlamasına rağmen yüksek tahmin zaman karmaşılığından dolayı ölçeklenebilir değildir. Her sorgu-pasaj çifti için uçtan uca tahmin yapılması gerektiği için indeksleme yöntemiyle birleştirilemez. Öne çıkan örnekler arasında DRMM [14] ve BERT tabanlı çapraz kodlayıcılar (cross-encoder) bulunur.
- **Hibrit Modeller (Hybrid Models):** Bu modeller, temsil tabanlı kodlayıcıların hızını, etkileşim tabanlı mekanizmaların hassas eşleştirme yetenekleriyle birleştirerek her iki yöntemin en iyi yönlerini entegre eder. Genellikle çok aşamalı eğitim stratejileri veya bilgi damıtma (knowledge distillation) stratejileri kullanarak eğitilirler. Tahmin hızı ile erişim başarısını dengelerler. Dikkate değer uygulamalar arasında ColBERT ve son zamanlardaki iki aşamalı getirici-sıralayıcı (retriever-reranker) mimariler yer alır.

Tablo 2.1 Nöral bilgi erişim paradigmalarının karşılaştırılması

Paradigma Adı	**Eğitim Verisi İhtiyacı**	**Erişim Başarısı**	**Genelleme Yeteneği**	**Erişim Hızı**
Etkileşim Tabanlı	Yüksek	Yüksek	Düşük	Düşük
Temsil Tabanlı	Yüksek	Düşük	Düşük	Yüksek
Hibrit	Düşük	Orta	Yüksek	Orta

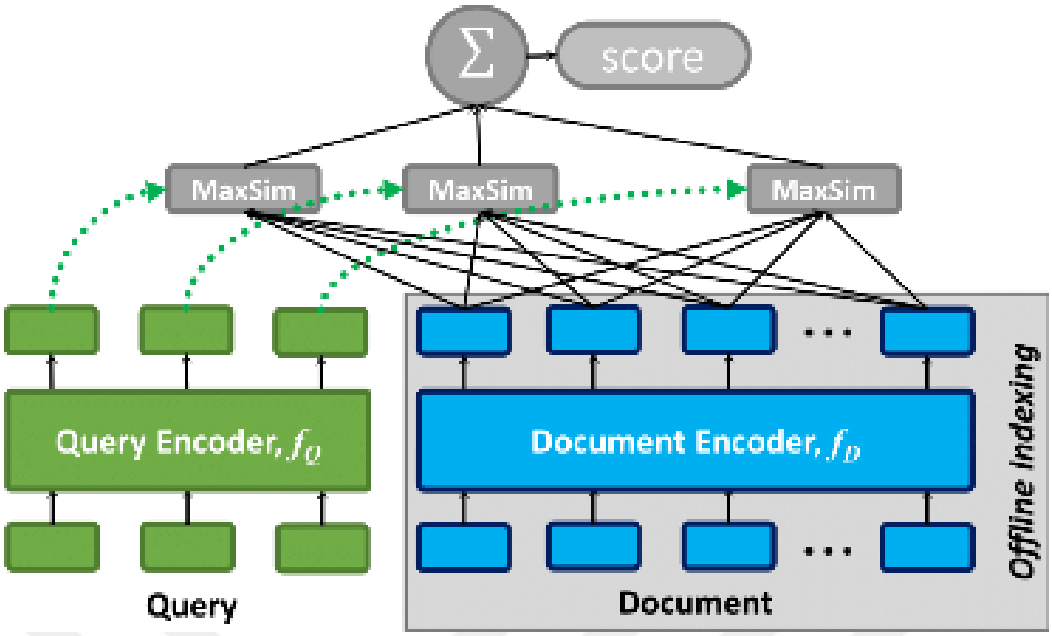


Şekil 2.2 Nöral bilgi erişimi paradigmaları [1]

2.6.1 ColBERT Paradigması

ColBERT (Contextualized Late Interaction over BERT), geleneksel embedding modellerinden köklü bir paradigma değişimini temsil etmektedir. BERT, her bir tokenin vektörel temsillerini tek bir vektöre indirgerken; ColBERT, her token için ayrı ayrı temsilleri koruyarak benzerlik hesaplamalarında daha ince taneli (granüler) bir yaklaşım sunar. ColBERT'i özgün kılan temel unsur ise, sorgu-belge benzerliklerinin hesaplanmasında kullanılan ve ilerleyen bölümlerde ayrıntılandırılacak olan özgün bir geç etkileşim (late interaction) mekanizmasını devreye sokmasıdır. Bu mekanizma, sorgu ve belgelerin birbirlerinden bağımsız biçimde işlenmesini olanaklı kılarak, erişim sürecinin yalnızca son aşamasında etkileşime girmelerini sağlar; böylece hem verimli hem de yüksek doğruluklu sıralama ve erişim gerçekleştirilir.

Özünde, BERT ve benzeri geleneksel embedding modelleri, her belge için tekil bir vektör üretir ve sorguya olan ilgisini tek bir sayısal skorla ifade eder. Buna karşılık ColBERT, sorgu ile belge arasındaki anlamsal ilişkiyi çok daha ayrıntılı biçimde ele alır. Her sorgu ve her belge için birer vektörler dizisi sunar. Üretilen vektör dizileri birbirleriyle karşılaştırılarak sorgu ve belgeler arasındaki benzerlik hesaplanır. Bu yaklaşım, sorgular ile belgeler arasındaki anlamsal ilişkinin daha hassas ve katmanlı bir şekilde çözümlenmesini mümkün kılar.



Şekil 2.3 ColBERT mimarisi [1]

Şekil 2.3'te görüldüğü üzere ColBERT mimarisi şu bileşenlerden oluşur:

- Sorgu encoderi
- Doküman (pasaj) encoderi
- Late interaction mekanizması

Bir sorgu **Q** ve bir belge **D** işlendiğinde, ColBERT öncelikle bir sorgu kodlayıcı (query encoder) kullanarak **Q**'yu sabit boyutlu vektörler kümesine dönüştürür; bu küme **Eq** ile gösterilir. Aynı şekilde, bir belge kodlayıcı (document encoder) da **D**'yi başka bir sabit boyutlu vektörler kümesine, yani **Ed**'ye dönüştürür. **Eq** ve **Ed**'deki her bir vektör, kendi bağlamındaki diğer terimlerle etkileşim hâlinde öğrenildiğinden, bağlamsal bilgiyle zenginleştirilmiştir.

Bu temsiller elde edildikten sonra, ColBERT geç etkileşim (late interaction) olarak tanımlanan bir yöntemle **Q** ve **D** arasındaki alaka düzeyini (relevance score) hesaplar. Bu yöntem, maksimum benzerlik (MaxSim) işlemlerinin bir tür toplamsal birleştirilmesiyle tanımlanır. Bu yaklaşım, **Eq**'daki her bir vektör **v**'nin **Ed**'deki vektörlerle olan kosinüs benzerliklerini hesaplar ve her biri için en yüksek benzerlik değerleri seçilir. Bu maksimum benzerlik skorları daha sonra toplanarak nihai ilişki skoru oluşturulur. Kosinüs benzerliğine ek olarak, ColBERT, vektörler arası benzerliği ölçmede karesel L2 uzaklığı metriğini de kullanabilmektedir. Sonuçta

ColBERT, her sorgu tokeni için elde edilen bu eşleşme skorlarını birleştirerek, belgenin sorguyla ilgili olup olmadığını tahmin eder.

$$S_{q,d} := \sum_{i \in [|E_q|]} \max_{j \in [|E_d|]} E_{q_i} \cdot E_{d_j}^T$$

Şekil 2.4 MaxSim mekanizması

2.6.2 Çok Dilli ColBERT Çalışmaları

Günümüzde çok dilli bilgi erişim sistemleri, özellikle düşük kaynaklı dillerde ciddi performans sorunlarıyla karşı karşıyadır; çünkü mevcut yöntemler genellikle yalnızca İngilizce gibi yüksek kaynaklı dillerde etkili sonuçlar verirken, diğer diller için yeterli etiketli veri bulunmaması nedeniyle başarısız kalmaktadır. Bu sistemler genellikle çok dilli büyük dil modellerine dayanır ve her dil için ayrı ayrı ince ayar (fine-tuning) yapılmasını gerektirir; bu da yüksek hesaplama maliyetlerine, veri bağımlılığına ve düşük ölçeklenebilirliğe yol açar. Ayrıca, bu yaklaşımlarda yeni bir dili desteklemek için modelin tamamının yeniden eğitilmesi veya karmaşık yöntemlerle güncellenmesi gerekebilir, bu da hem teknik zorluklar yaratır hem de sistemin esnekliğini kısıtlar. Sonuç olarak, düşük kaynaklı diller için etkili, ekonomik ve genişletilebilir bilgi erişim sistemleri geliştirmek büyük bir açık olarak karşımızda durmaktadır.

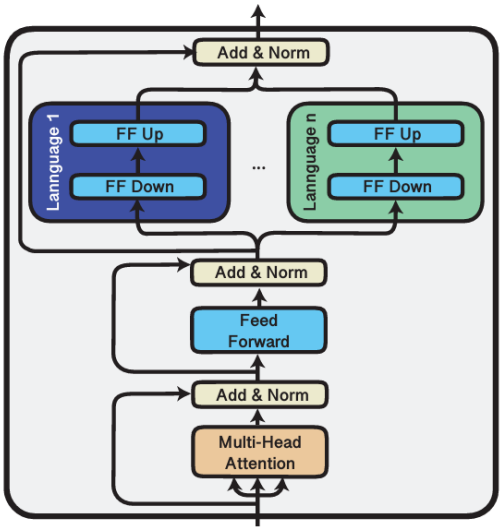
2.6.2.1 ColBERT-XM

ColBERT-XM [15], çok dilli sıfır-atışlı (zero-shot) bilgi getirme problemini hedefleyen, esnek ve modüler bir mimariye sahip bir IR modelidir. Bu model, tek bir dilden eğitilmiş bir bilgi getirme sistemini, ek eğitim verisi olmaksızın diğer dillere genelleştirebilmek için geliştirilmiştir. ColBERT-XM, önceki çok dilli IR çalışmalarında görülen başarı-kapsam ikilemini çözmek üzere tasarlanmıştır: ya yüksek doğruluk sağlanır ama çok dilli kapsayıcılık düşer, ya da birçok dili desteklemek mümkün olur ancak performanstan taviz verilir. ColBERT-XM bu dengeyi aşmak için hem etkili hem de ölçeklenebilir bir çözüm sunar.

ColBERT-XM, ColBERT yapısını çok dilliliğe genişletmek için encoder bileşenlerini modüler hâle getirir. Encoder bileşenlerini modüler hale getirmek için

X-MOD [16] yöntemini kullanır. X-MOD (Cross-lingual Modular Encoder), çok dilli dil temsillerini modüler biçimde öğrenen bir encoder mimarisidir. Her dil için ayrı bir dil modülü içerir, ancak bu modüller aynı backbone ile entegre çalışır. Bu sayede:

- Diller arası bilgi paylaşımı sağlanır (özellikle düşük kaynaklı diller fayda görür). Ancak her dilin kendine has özellikleri (morfoloji, söz dizimi) korunur.
- Yeni bir dil ekleneceği zaman tüm model yeniden eğitilmek zorunda kalmaz, yalnızca o dilin modülü eğitilir.



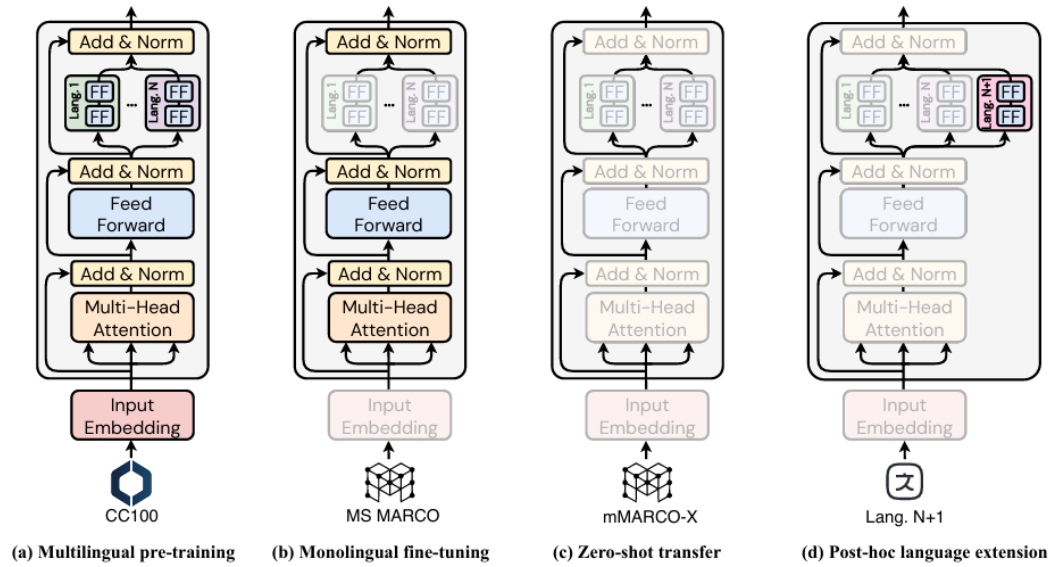
Şekil 2.5 X-MOD mimarisi [16]

ColBERT-XM modelinin eğitiminin ilk aşamasında modelin X-MOD mimarisiyle çok dilli olarak modüler eğitimi gerçekleşir. Bu ön eğitim ile düşük ve yüksek kaynaklı diller aynı öznitelik uzayında temsil edilir. İnce ayar aşamasında sadece İngilizce IR verisi kullanılarak dil spesifik olmayan modüllerin eğitimi gerçekleşir.

Bu mimarideki önemli avantajlardan biri, tüm diller için ayrı ayrı büyük miktarda eğitilmiş IR verisi gerektirmeden, sıfır-atışlı arama yapabilmesidir. Yani, örneğin sadece İngilizce üzerinde eğitilmiş model, Almanca veya Japonca gibi bir dilden gelen sorgularla da etkili arama yapabilir. Bu, hem kaynak bakımından sınırlı diller için IR sistemleri geliştirmeyi kolaylaştırır hem de sistemin geniş bir dil yelpazesinde çalışmasını mümkün kılar. Ayrıca, modüler yapı sayesinde yeni diller

sisteme eklenebilirken mevcut encoderler bozulmadan kalabilir, bu da bakım ve genişletme maliyetlerini düşürür.

Deneysel sonuçlar, ColBERT-XM'in özellikle düşük kaynaklı dillerde bile güçlü sıfır-atışlı performans sunduğunu ve hem doğruluk hem de çok dillilik açısından mevcut yöntemleri geride bıraktığını göstermektedir. Özetle ColBERT-XM, çok dilli bilgi getirme sistemlerinde pratik, etkili ve ölçeklenebilir bir çözüm sunan ileri seviye bir yaklaşımdır.



Şekil 2.6 ColBERT-XM eğitim aşamaları [15]

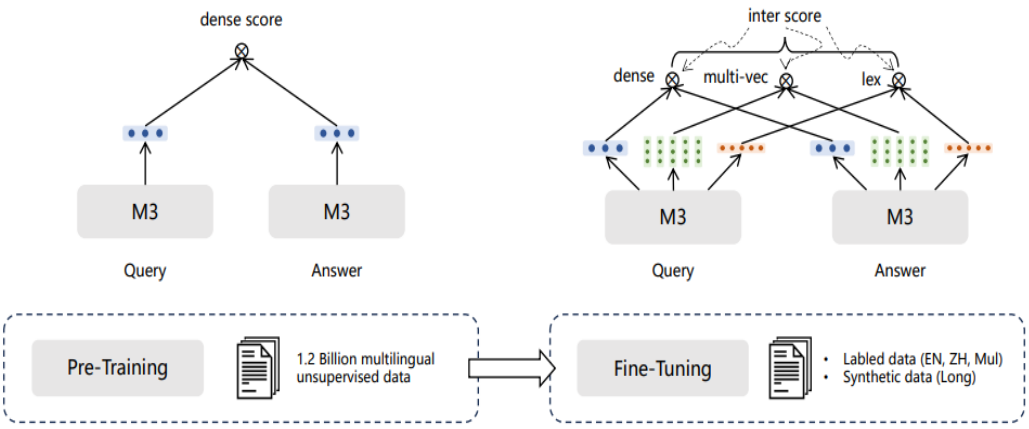
2.6.2.2 BGE-M3

Yapay zeka alandaki gelişmeler, kurumların farklı görevler, diller veya giriş uzunlukları için çok sayıda uzmanlaşmış yapay zeka modelini yönetme, entegre etme ve güncelleme karmaşıklığıyla giderek daha fazla karşı karşıya kalması nedeniyle, "genelci" yapay zeka modellerine olan talebin artmasına sebep olmaktadır. Bu trende ayak uyduran M3-Emnbedding (Multi Linguality, Multi Functionality, Multi Granulity) [17] çalışması sparse, dense ve çok vektörlü (ColBERT) nöral IR paradigmalarını tek modelde birleştirmiştir. Geliştirilen model paradigmadan ayrı olarak doküman uzunluğu ve dil desteği açısından da genelci bir IR modeldir. M3-Embedding gibi tek, son derece çok yönlü bir model, operasyonel yükü önemli ölçüde azaltarak, sistem tasarımını basitleştirerek ve ölçeklenebilirliği artırarak cazip bir çözüm sunmaktadır. Bu durum, yapay zeka

geliştirmede, daha geniş bir kullanım senaryosu yelpazesine uyum sağlayabilen temel modellere doğru genel bir eğilimi işaret etmekte, böylece pratik ayarlarda dağıtımı ve yeniliği hızlandırmaktadır.

Yapay zeka uygulamaları yaygınlaştıkça, ağırlıklı olarak İngilizce merkezli modeller doğal olarak pazar erişimlerini sınırlar ve dijital eşitsizlikleri sürdürür. Çok çeşitli dillerde en son teknoloji performans sunarak, M3-Embedding dünya çapında daha eşit ve erişilebilir yapay zeka sistemlerinin geliştirilmesini doğrudan kolaylaştırmaktadır. Bu, İngilizce konuşmayan pazarlar için önemli bir giriş engelini ortadan kaldırmakta ve yapay zekada dilsel çeşitliliği teşvik ederek gelişmiş IR'ın faydalarının birkaç baskın dille veya kültürle sınırlı kalmamasını sağlamaktadır.

M3-Embedding'in çok işlevliliğinin merkezinde, ortaya konulan kendi kendine bilgi damıtma (self knowledge distillation) çerçevesi yer almaktadır. Bu çerçeve, birden fazla erişim işlevinin (dense, sparse, çok vektörlü) birlikte öğrenilmesini ve karşılıklı olarak güçlendirilmesini sağlar. Bu yaklaşım, farklı erişim işlevlerinden türetilen ilişki puanlarının, bilgi damıtma yoluyla genel öğrenme sürecini geliştirmek ve eğitim kalitesini artırmak için bir "öğretmen sinyali" olarak entegre edilmesini içerir. Bu yenilikçi yaklaşım, modelin çeşitli erişim ihtiyaçlarına hizmet eden birleşik bir temsil öğrenmesini sağlar.



Şekil 2.7 M3-Embedding eğitim aşamaları [17]

M3-Embedding'in başarısı için model mimarisi ve eğitim algoritmaları kadar, kapsamlı ve yüksek kaliteli bir veri kümesinin titizlikle kürasyonu da kritik öneme sahiptir. Bu, modelin çok dilli ve çok taneli yeteneklerinin temelini oluşturmaktadır. Araştırmacılar, üç ana kaynaktan veri derlemişlerdir:

- **Denetimsiz Öğrenme Ön Eğitim Verisi:** Bu ön eğitim verisi, büyük çok dilli veri kümelerinden (örneğin, Wikipedia, S2ORC [18], xP3 [19], mC4 [20], CC-News, MTP [21]) zengin anlamsal yapılar (örneğin, başlık-gövde çiftleri) şeklinde elde edilmiştir. Diller arası anlamsal eşleştirmeyi kolaylaştırmak için NLLB[22] ve CCMatrix'ten [23] paralel cümleler eklenmiştir. Bu çaba, 194 dilde 1,2 milyar metin çifti ile sonuçlanmıştır. Bu muazzam hacim ve dilsel genişlik, modelin çok dillilik ve çok tanelilik yeteneğinin kaynağıdır.
- **Denetimli Öğrenme İnce Ayar Verisi**: Daha küçük boyutlu ancak çeşitli ve yüksek kaliteli ince ayar verileri (İngilizce için HotpotQA [24], TriviaQA [25], MS MARCO [26]; Çince için DuReader [27], T2-Ranking [28]; diğer diller için Mr. TyDi [29] ve MIRACL [30]) dahil edilmiştir. Uzun belge erişim görevlerindeki eksikliği gidermek ve farklı dillerde ince ayar verileri sağlamak için stratejik olarak sentetik veri (MultiLongDoc) oluşturulmuştur. Bu, GPT-3.5 kullanılarak uzun makalelerden rastgele örneklenmiş paragraflara dayalı sorular oluşturularak yeni metin çiftleri oluşturulmuştur.

Yapılan değerlendirmelerde kendi kendine bilgi damıtmanın, sentetik veri üretmenin ve veri kürasyonunun katkısı gösterilmiştir. Model birçok dilde, yayınlandığı tarihte mevcut olan modelleri IR başarısı açısından geride bırakmıştır.

2.6.3 Jina ColBERT v2

BGE-M3 gibi Jina-ColBERT-v2 [31] çalışması da benzer problemleri ele almaktadır, ek olarak tahmin yapma kompleksliğini düşürme üzerinde de durulmuştur. Bahsedilen problemleri çözmek, istenen hedeflere ulaşmak için Jina-ColBERT-v2 çalışmasında eğitim stratejisi, model mimarisi ve eğitim verisi açısından değişiklikler yapılmıştır. Temsil tabanlı nöral bilgi erişiminde kullanılan eğitim yöntemleri kullanılmıştır.

Backbone modeli için XLM-RoBERTa mimarisi kullanılmıştır, değişiklik olarak mutlak pozisyon embeddingleri yerine rotary pozisyon embeddingleri kullanılmıştır. Ön eğitimi yapılmış XLM-RoBERTa, pozisyon embeddingleri değiştirildikten sonra Refined Web korpusunda [32] 160 bin adım boyunca MLM objektifiyle eğitilmiştir. ColBERT paradigması token seviyesindeki

embeddinglerin boyutunu tahmin yapmayı ve semantik benzerlik aramayı hızlandırmak için 768'den 128'e lineer bir katmanla düşürür. Bu çalışmada hızlandırmayı arttırmak için embedding boyutunu 512,256,128, 96 ve 64'e düşüren birden fazla ayrı katman kullanılmıştır. Tüm bu katmanlar eş zamanlı olarak eğitilmiştir ve boyutu 128'den daha az boyuta indirgemenin getirdiği başarı kaybının minimal düzeyde olduğu gösterilmiştir.

Klasik ColBERT yaklaşımından farklı olarak, IR ince ayarı için ilk aşamada temsil tabanlı nöral IR modellerinde kullanılan tek vektörlü kontrastif öğrenme objektifi kullanılmıştır. %50'si İngilizce olmak üzere 450 milyon birbiriyle ilişkili soru-doküman, soru-cevap ve cümle-cümle çiftiyle bu eğitim gerçekleştirilmiştir.

IR ince ayarı için ikinci aşamada bilgi damıtma kullanılmıştır ve öğretmen model olarak jina-reranker-v2-base-multilingual [33] etkileşim tabanlı modelin rehberliğinde 100 bin adım boyunca eğitim gerçekleşmiştir. Eğitim verisi kompozisyonu belirsizlikler barındırmakla beraber şu şekildedir:

- MSMARCO, DuReader ve MIRACL **gibi** çeşitli biçimlerde seçilmiş olumsuz örnekler içeren, yüksek kaliteli ve insanlar tarafından etiketlenmiş araştırma veri kümeleri.
- İngilizceden Çince, Fransızca, Almanca, Japonca, Rusça ve İspanyolca'ya çevrilmiş MSMARCO ve NQ [34] **gibi** yüksek kaliteli veri kümeleri.
- Hukuk **gibi** niş alanlarda yaygın olarak gözlemlenen başarı düşüşlerinin önüne geçmek için niş alanlarda sentetik olarak oluşturulmuş veri kümeleri.

Bu üçlü veri kümesi, Arapça, Çince, İngilizce, Fransızca, Almanca, Japonca, Rusça ve İspanyolca dillerine özellikle ağırlık verilerek, yaygın biçimde kullanılan toplam 14 dili kapsamaktadır. Değerlendirme sonuçlarında mMARCO [35] ve MIRACL çok dilli karşılaştırma verisetlerinde üstün başarı elde edilmiştir. Dikkate değer bir yorum olarak yazarlar IR verisetlerinin milyonlarca örnek içermesinin değerlendirme açısından pratik olmadığını belirtmişlerdir. Büyük verisetlerinin küçük örneklemlerinde elde edilen başarı metriklerinin ana verisetiyle elde edilen metriklerle paralellik gösterdiği belirtilmiştir. Her ne kadar yapılan çalışma tekrar edilebilir olmasa da embedding boyutunu düşürme ve farklı eğitim stratejilerinin karşılaştırılması açısından değerlidir.

2.7 IR Eğitim Verisetleri Üzerine

İyi bir pasaj erişim veri seti, bilgiye dayalı sorgulara verilen doğru cevapların güvenilir bir şekilde öğrenilebilmesi ve değerlendirilmesi için belirli niteliklere sahip olmalıdır. Öncelikle, sorgu ile gerçekten ilişkili olan pasajların yüksek doğrulukla eşleştirildiği, kaliteli bir sorgu-pasaj yapısı gereklidir. Bu eşleşmeler, kullanıcıların gerçek bilgi ihtiyacını yansıtan sorgularla yapılmalı ve cevap olarak verilen pasajlar, soruyu doğrudan ve açık biçimde yanıtlamalıdır. Bunun yanında, modeli yanıltmamak ve genelleme becerisini artırmak için sadece doğru değil, aynı zamanda konuyla alakalı ama cevabı içermeyen (hard negative) ya da tamamen alakasız (random negative) örneklere de yer verilmelidir.

Veri seti, farklı türdeki bilgi ihtiyaçlarını kapsayacak şekilde çeşitli sorgular içermelidir. Bunlar yalnızca doğrudan bilgi isteyen yüzeysel sorular değil; aynı zamanda açıklayıcı, karşılaştırmalı veya daha derinlemesine anlamaya yönelik sorgular da olmalıdır. Bu çeşitlilik, modellerin çok yönlü sorgulara karşı daha doğru sonuçlar üretmesini sağlar. Ayrıca veri setinin kapsadığı konular da geniş bir yelpazeye yayılmalı; örneğin sağlık, teknoloji, tarih, bilim ve kültür gibi farklı alanlardan içerikler içermelidir. Bu temsili çeşitlilik, modelin yalnızca belirli alanlarda değil, genel bilgi dağılımı içinde de iyi performans göstermesine katkı sağlar.

Pasajların uzunluk ve yapı bakımından da çeşitli olması önemlidir. Bazı cevaplar kısa bir tanımken, bazıları detaylı açıklamalar içerebilir. Bu da modelin kısa ve öz cevaplarla, daha kapsamlı metinler arasında ayrım yapabilme yetisini geliştirir. Ayrıca, değerlendirme yapılacaksa her sorguya karşılık gelen birden fazla doğru cevap bulunmalı ve bu cevaplar uzmanlar veya kullanıcılar tarafından dikkatlice etiketlenmiş olmalıdır. Bu sayede, modelin performansı çoklu doğru üzerinden daha adil ve gerçekçi şekilde ölçülebilir.

Son olarak, veri seti yapay veya sentetik verilerle destekleniyorsa, bu içeriklerin doğal dil yapısına uygun ve gerçek kullanıcı ifadelerine benzer biçimde oluşturulmuş olması gerekir. Aksi halde, modelin öğrenme süreci yapay örüntülere odaklanabilir ve gerçek dünyadaki sorgulara karşı zayıf performans gösterebilir. Tüm bu özellikler, güçlü ve güvenilir bir pasaj erişim verisetinin temelini oluşturur.

Question answering veri setleri, doğrudan bilgiye dayalı soru-cevap eşleştirmeleri sunduklarından, bilgi erişimi ve sıralama problemleri açısından zengin bir kaynak teşkil etmektedir. Özellikle açık domain QA veri setlerinde her soru, bir veya birden fazla ilgili paragrafla (context) eşleştirildiğinden, bu yapıların pasaj erişim görevine dönüştürülmesi mümkündür. Bu dönüşümde, her bir soru-pasaj çifti bir aday olarak değerlendirilmekte; pozitif örnekler, cevabı içeren pasajlar olarak, negatif örnekler ise cevapsız veya konuyla ilişkisi zayıf pasajlar olarak tanımlanmaktadır. Bu bağlamda, QA veri setleri, sıralama modellerinin denetimli olarak eğitilmesinde kullanılabilmektedir. Ancak bu tür veri setlerinin doğrudan pasaj erişim amaçlı üretilmemiş olması, bazı metodolojik sınırlamaları da beraberinde getirmektedir. Örneğin, pozitif pasajların sıklıkla yalnızca bir taneyle sınırlı olması veya negatif örneklerin gerçekçi dağılımlardan gelmemesi, sıralama performansını olumsuz etkileyebilmektedir. Dolayısıyla, QA veri setlerinin pasaj sıralama bağlamında kullanımı, dikkatli bir negatif örnekleme stratejisi ve uygun bir yeniden yapılandırma süreci ile desteklenmelidir. Bu yaklaşımla, QA kaynakları sıralama görevlerinde güçlü bir başlangıç noktası olarak değerlendirilebilir.

2.8 IR Başarı Metrikleri

Bilgi erişimi başarının ölçümünde her metrik her sorgu-pasaj ilişkisinde kullanılamayacağı için öncelikle sorgu-pasaj ilişkisinin tanımlanması gerekir. Sorgu-pasaj ilişkisi farklı şekillerde tanımlanabilir:

- İlişkili ve ilişkisiz olarak ikili sınıflandırılabilir.
- Çok ilişkili, az ilişkili, ilişkisiz gibi ordinal bir değişkenle tanımlanabilir.
- İlişki oranı mutlak sıfıra sahip sürekli bir değişkenle temsil edilebilir.

2.8.1 Precision (Kesinlik):

Kesinlik metriği ikili sınıflandırma problemlerinde hedef sınıfla sınıflandırılan örneklerden yüzde kaçının hedef sınıfa ait olduğunu gösterir. Denklem 2.1'de #TP doğru tahmin edilen pozitif sınıfa ait örnek sayısını, #FP yanlış tahmin edilen negatif sınıfa ait örnek sayısını belirtmektedir.

$$\text{Precision} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FP}} \tag{2.1}$$

Bilgi erişimindeki karşılığı ise getirilen pasajlardan yüzde kaçının alakalı pasaj olduğudur (bakınız Denklem 2.2) .

$$\text{PrecisionIR} = \frac{|\{\text{alakalı pasajlar}\} \cap \{\text{erişilen pasajlar}\}|}{|\{\text{erişilen pasajlar}\}|} \tag{2.2}$$

2.8.2 Recall (Duyarlılık):

Duyarlılık metriği ikili sınıflandırma problemlerinde hedef sınıf örneklerinin yüzde kaçının doğru tahmin edildiğini gösterir. Denklem 2.3'de #TP doğru tahmin edilen pozitif sınıfa ait örnek sayısını, #FN yanlış tahmin edilen pozitif sınıfa ait örnek sayısını belirtmektedir.

$$\text{Recall} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FN}} \tag{2.3}$$

Bilgi erişimindeki karşılığı ise alakalı pasajlardan yüzde kaçının getirildiğidir (bakınız Denklem 2.4).

$$\text{RecallIR} = \frac{|\{\text{alakalı pasajlar}\} \cap \{\text{erişilen pasajlar}\}|}{|\{\text{alakalı pasajlar}\}|} \tag{2.4}$$

Kesinlik ve duyarlılık metriğinin bilgi erişimi açısından anlamlı olabilmesi için getirilen pasajların sıralamalarının dikkate alınması gerekir. Alakalı pasajların getirilen pasajların ilk sıralarında olması bilgi erişim başarısının yüksek olduğunu gösterir. Bu yüzden kesinlik ve duyarlılık hesaplanırken getirilen ilk k adet pasajı incelemek sıralama başarısını daha iyi yansıtır.

2.8.3 Precision@k

$$\text{Precision@k} = \frac{|\{\text{getirilen ilk k pasaj}\} \cap \{\text{alakalı pasajlar}\}|}{\text{k}} \tag{2.5}$$

2.8.4 Recall@k

$$\text{Recall@k} = \frac{|\{\text{getirilen ilk k pasaj}\} \cap \{\text{alakalı pasajlar}\}|}{|\{\text{alakalı pasajlar}\}|} \tag{2.6}$$

2.8.5 R-Precision

Precision metriğinin IR alanı için uyarlanmış halidir. Koleksiyondaki alakalı pasaj sayısına bağımlı bir metriktir. Precision@K formülünde K yerine alakalı pasaj sayısı konarak hesaplanır.

2.8.6 Mean Average Precision (MAP)

Recall@K ve Precision@K getirilen ilk k dökümanın nasıl sıralandığını dikkate almaz. Örneğin bir bilgi erişim modeli alakalı pasajı ilk sırada getirebilir, bir diğer model ise alakalı pasajı üçüncü sırada getirebilir. Precision@k ve Recall@k metrikleri her iki model için aynı başarı değerini verecektir. İlk k pasajın sıralamasının başarıya etki etmesi için Precision ve Recall metriklerinin tek bir k değeriyle hesaplanması yerine 1'den k'ya her rakam için hesaplanabilir ve ortalaması alınabilir. Mean average precision metriği bahsedilen yaklaşımı kullanır.

MAP@K hesaplanması için öncelikle her sorgu için average precision (AP) değeri hesaplanır. AP'yi hesaplamak için getirilen ilk K pasajdaki alakalı her pasajın bulunduğu konum için Precision@k değeri hesaplanır ve hesaplanan değerlerin ortalaması alınır (bakınız Denklem 2.7) . Denklem 2.7'de gösterilen "rel(k)" fonksiyonu alakalı pasajlar için 1, alakasız pasajlar için 0 değerini döndürür. Sonrasında her sorgu için hesaplanan AP değerlerinin ortalaması alınarak MAP değeri bulunur (bakınız Denklem 2.7).

$$\text{AP@K} = (\tfrac{1}{\text{N}})\text{x}\sum_{\text{k=1}}^{\text{K}}(\text{Precision@k x rel(k)}) \tag{2.7}$$

$$\text{MAP@K} = (\frac{1}{\#\text{sorgu}})\text{x}\sum_{\text{u=1}}^{\#\text{sorgu}}\text{AP@K}_{\text{u}} \tag{2.8}$$

2.8.7 Cumulative Gain (CG)

Sorgu-pasaj ilişkileri her zaman ikili değerlerle temsil edilmeyebilir. İlişki sürekli değerler ile temsil edildiğinde Precision ve Recall tabanlı metrikler kullanılamaz. Cumulative Gain (CG) sürekli ilişki değerleriyle kullanılabilen bir metriktir. Basitçe getirilen pasajların sorguyla olan ilişki değerlerinin toplamıdır.

$$\mathrm{CG@K} = \sum_{k=1}^{K} \mathrm{İlişki}_i \tag{2.9}$$

2.8.8 Discounted Cumulative Gain (DCG)

MAP metriğinin zayıf yanı 1'den K'ya kadar ağırlıklandırmaksızın her konumun Precision değerlerinin ortalamasını almasıdır. Örneğin getirilen ilk 3 pasajdaki Precision değeri ilk 10 pasajdaki Precision değeriyle aynı oranda AP@10 değerini etkileyecektir. Benzer şekilde CG metriği de sıralamayı skor hesaplamasına dahil etmez. İlk sıralamaların başarı skorunu daha çok etkilemesi için ilk sıralamaların ödüllendirilmesi gerekmetedir veya başka bir deyişle son sıralamaların cezalandırılması gerekmektedir. DCG bu yaklaşımı kullanır. İlişki değerlerini toplamadan önce her değeri konumuna göre logaritma fonksiyonu ile cezalandırır. Sonuç olarak yüksek ilişki değerlerine sahip pasajların ilk sırada getirilmesi skoru daha çok yükseltir.

$$\mathrm{DCG@K} = \sum_{i=1}^{K} \mathrm{İlişki}_i \,/\log(i+1) \tag{2.10}$$

2.8.9 Normalized DCG (NDCG)

DCG metriği skalasız bir başarı metriğidir. Örneğin bir bilgi erişim modelinin bir verisetinde ürettiği DCG metriği tek başına yorumlanamaz ayrıyetten referans bir değere ihtiyaç vardır. Değerin düşük veya yüksek olması tek başına anlam ifade etmez sadece başka bir modelin aynı verisetinde ürettiği DCG değerine bakılarak iki model karşılaştırılabilir. NDCG metriği bu problemi çözmek için yapılabilecek en iyi-ideal sıralamanın üreteceği DCG değerini referans değer olarak kullanır. Bu sıralamanın ürettiği değer ideal-DCG olarak adlandırılır. Modelin ürettiği DCG değeri ideal-DCG'ye bölünür ve sonuçları yorumlamak için başka bir modele ihtiyaç duyulmaz. NDCG değeri sıfır ve bir arasında değer alabilir. Bir değeri mükemmel sıralamayı gösterir.

$$\mathrm{NDCG@K} = \mathrm{DCG@K}/\mathrm{IdealDCG@K} \tag{2.11}$$

2.8.10 Mean Reciprocal Rank (MRR):

MRR metriği ikili sorgu-pasaj ilişkileri için kullanılan bir metriktir. Getirilen pasajlardan alakalı olanlarından en iyi sıralamaya sahip olan pasajın konumunu kullanır. Alakalı pasaja ne kadar çabuk erişildiğini gösteren bir metriktir. Hızlı erişimin önemli olduğu problemlerde diğer metriklerden daha önemli olabilir. Sadece getirilen ilk ilişkili pasajı dikkate aldığı için birden fazla ilişkili pasajın olduğu problemlerde başarı temsil gücü düşüktür.

$$\text{MRR} = \left(\frac{1}{\#\text{sorgu}}\right) \sum_{u=1}^{\#\text{sorgu}} 1/\text{En İyi Konum}_u \tag{2.12}$$

3
YÖNTEM

Bu bölümde Türkçe bilgi erişim modelinin geliştirilme aşamaları ve yapılan tercihler anlatılacaktır.

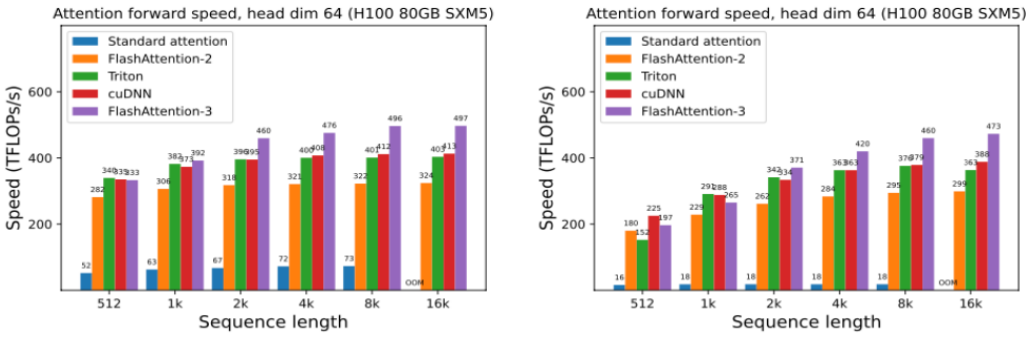
3.1 Dil Modeli Mimarisi Seçimi

Transformer mimarileri, hazırlık bölümünde de açıklandığı üzere üç ana kategoriye ayrılmaktadır: encoder, decoder ve encoder-decoder modelleri. Decoder ve encoder-decoder yapıları, tasarımları gereği dil üretiminin gerekli olduğu doğal dil işleme görevlerinde tercih edilmektedir. Her ne kadar decoder tabanlı modeller, bilgi erişimi bağlamında encoder modellere kıyasla bazı tasarımsal dezavantajlara sahip olsa da, bu modellerin pasaj erişiminde kullanıldığı çalışmalar da mevcuttur [36]. Bu kullanım, decoder modellerin terabaytlarca metin verisiyle önceden eğitilmiş olmalarının, söz konusu yapısal dezavantajları telafi etmesiyle mümkün hale gelmiştir. Ancak bu modellerin eğitiminin yüksek maliyetli olması ve çıkarım sürecinde ölçeklenebilirlik açısından sınırlamalar taşıması nedeniyle, bu tez çalışmasında decoder modeller değerlendirme dışı bırakılmıştır.

ColBERT'in ilk versiyonunda da isminden anlaşılacağı üzere BERT mimarisi temel alınmıştır. Bir encoder modeli olan BERT, çift yönlü bağlam etkileşimi ile öznitelik çıkarımında güçlüdür ve yüksek ölçeklenebilirlik özellikleri sayesinde pasaj erişimi için uygun bir yapı sunar. Genel olarak encoder tabanlı modellerin kullanım alanları, sekans seviyesinde veya token seviyesinde olarak ikiye ayrılabilir. Pasaj erişimi her ne kadar sorgu ve pasaj arasındaki ilişkiyi sekans düzeyinde skorlayan bir görev gibi görünse de, ColBERT'in benimsediği geç etkileşim paradigması, bu ilişkiyi token seviyesinde elde edilen temsiller üzerinden değerlendirmektedir. Dolayısıyla ColBERT ile uyumlu bir encoder tasarımının

token düzeyindeki performansı doğrudan önem kazanmaktadır. Hazırlık bölümünde ele alınan RoFormer mimarisi, kullandığı rotary pozisyonel embedding mekanizması sayesinde, MLM görevinde daha düşük kayıp değerleri üretmektedir. Maskelenmiş tokenlerin daha doğru şekilde tahmin edilebilmesi, modelin token düzeyindeki başarısının daha iyi olduğunu göstermektedir. Bu nedenle, ColBERT tabanlı bir modelin eğitilmesinde rotary pozisyonel embedding kullanan bir mimarinin tercih edilmesi, görev performansı açısından avantaj sağlayacaktır.

Derin öğrenme özelinde son yıllarda sadece donanımsal gelişmeler olmamış aynı zamanda yazılımsal gelişmeler de yaşanmıştır. Geliştirilen optimize kütüphaneler aynı bilgisayar donanımıyla daha az enerji tüketerek daha hızlı eğitim ve tahmin yapılabilmesini sağlamaktadır. Transformer modellerinde yapılan işlemlerde zaman karmaşıklığı olarak attention işlemi öne çıkmaktadır. Attention işlemi için yazılım seviyesinde geliştirilen farklı implementasyonlar bulunmaktadır. Seçilecek encoder model mimarisinin söz konusu gelişmeleri desteklemesi eğitim maliyetlerinin azaltılması açısından önemlidir. ModernBERT mimarisi eski BERT varyantlarının aksine Flash Attention implementasyonunu [37] desteklemektedir. Geliştirilecek bilgi erişim modeli için, hem rotary pozisyon embeddinglerini kullanması hem optimize attention implementasyonlarını desteklemesinden dolayı, ModernBERT-base mimarisi seçilmiştir. ModernBERT'in makalesinde alternating attention yönteminin 100 milyar eğitim tokeninden az eğitim verisinde başarı kaybına sebep olduğuna değinilmiştir. Alternating attentionın veri ihtiyacı ve öznitelik çıkarımını kısıtlamasından dolayı yerine global attention kullanılmıştır.



Şekil 3.1 Attention implementasyonlarının hız karşılaştırması [37]

3.2 Dil Modeli Ön Eğitimi

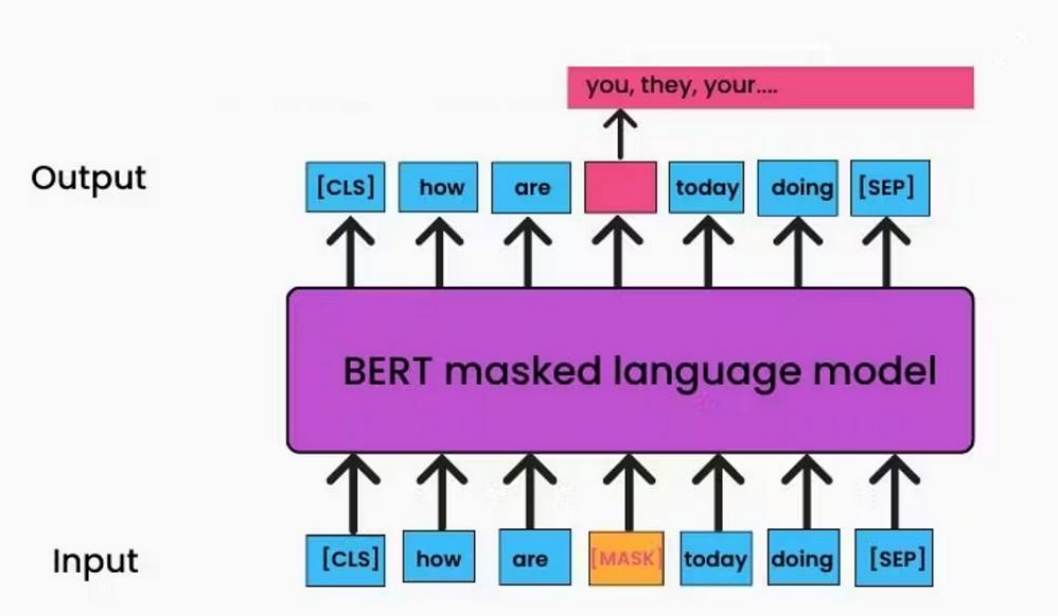
İngilizce dili için encoder only modeller özelinde sistematik onlarca ön eğitim çalışması bulunsa da Türkçe dilinde hiperparametre, veri kompozisyonu, eğitim objektifi, model mimarisi gibi faktörlerin incelendiği hiçbir ablasyon çalışması bulunmamaktadır. Türkçe dili için yapılan çalışmalarda diğer dillerde yapılan çalışmalarda bulunan en iyi yöntemler transfer edilmiştir. Makine öğrenmesinde deney aşaması sonucunda bulunan en iyi eğitim seçenekleri eğitim verisi ve probleme doğrudan bağlıdır. Bu kapsamda birbirinden çok farklı dillerde bulunan stratejilerin transferi sağlıklı olmayabilir. Türkçe encoder only model ön eğitimi özelinde araştırmaya değer noktalar şunlardır:

- **NSP görevinin terk edilmesi:** BERT çalışmasında 35GB veri kullanılmıştır ve NSP görevinin başarıyı arttırdığı gözlemlenmiştir. RoBERTa çalışmasında ise veriseti büyütüldüğünde NSP eğitim objektifinin katkısının olmadığı gösterilmiştir. NSP görevi tamamen kaldırıldığında sadece MLM kullanılması modele sadece token seviyesinde geri bildirim verileceği anlamına gelir. Türkçe dili İngilizce'ye göre bağlama daha duyarlı bir dil olduğu için NSP görevinin terk edilerek bağlam düzeyinde modele geri bildirim verilmemesi performansı nasıl etkileyecektir? Bir diğer nokta ise İngilizce dilinde çok yüksek kaliteli sekans sınıflandırma verisetleri bulunmaktadır ve modellere ön eğitim sonrasında bu verisetler kullanılarak transfer öğrenme uygulanmaktadır. Türkçe'de ise sekans sınıflandırma verisetleri ölçek ve kalite açısından çok daha geridedir.
- **Pozisyon embedding yöntemi seçimi:** İngilizce dilinde yapılan çalışmalarda farklı pozisyon embedding yöntemleri karşılaştırılmıştır. Fakat İngilizce'den çok farklı olan Türkçe dilinde paralel sonuçlar elde edilebilecek midir?
- **MLM görevinde maskeleme oranı seçimi:** İngilizce'de eğitilen modellerde bile en iyi maskeleme oranı konusunda farklı sonuçlara ulaşılmıştır. Özgün bir çalışmada ise sabit maskeleme oranı yerine maskeleme oranının dinamik olarak değiştirilmesinin model başarısını olumlu etkilediği gösterilmiştir [38]. MLM maskeleme oranı Türkçe spesifik modeller için araştırmaya değer bir konudur.

Bahsedilen noktaların araştırılması için sistematik şekilde çok sayıda deney yapılması gerekmektedir. Ortaya çıkacak maliyet sebebiyle bu araştırma konusu sonraki çalışmalara bırakılmıştır. Her ne kadar maliyet yüksek olsa da daha iyi bir encoder only model çok sayıda NLP görevini iyileştireceği için böyle bir çalışma Türkçe NLP alanını ileriye götürmek için gereklidir ve iyi fırsatlar sunmaktadır.

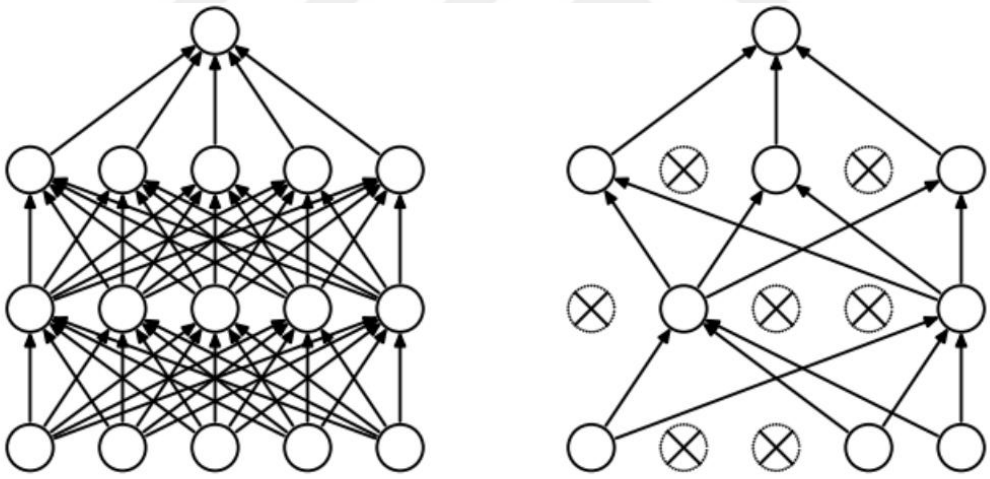
Gerçekleştirilen tez çalışmasında en düşük maliyetle en iyi Türkçe ColBERT modelini geliştirmek hedeflenmiştir. Ön eğitim aşamasında yapılacak olan yatırımın çıktısı ince ayar kısmında yapılacak olan yatırımın çıktısından daha az olacağı için bu tez çalışmasında kapsamlı bir ön eğitim araştırması yapmak mantıklı olmayacaktır.

ColBERT modeli için seçilen mimari ön eğitim aşamasında 35 GB boyutundaki Oscar korpusunda [39] RoBERTa çalışmasında kullanılan dinamik maskelemeyle MLM görevi üzerinde 8 epok boyunca eğitilmiştir. %20 maskeleme oranı, 2048 batch boyutu, 7e-4 öğrenme oranı ve lineer öğrenme oranı zamanlayıcısı kullanılmıştır. Model tokenizeri için BERTurk [40] çalışmasında eğitilen 32 bin tokenlik sözlüğe sahip uncased tokenizer kullanılmıştır. Büyük küçük harf duyarı olmayan tokenizer seçiminin arkasındaki neden aynı metnin daha az tokenle temsil edilebilmesinin amaçlanmasıdır. NER ve POS gibi görevlerde büyük-küçük harf ayrımı önemli katkı sağlasa da IR görevi için getirdiği katkı ihmal edilebilir.



Şekil 3.2 Maskelenmiş dil modelleme objektifi

Yapılan diğer encoder only modeli ön eğitimlerinden farklı olarak dropout kullanılmamıştır. Maskelenmiş dil modelleme görevi modele beslenen tokenlerin rastgele [MASK] tokeni ile değiştirilip tahmin edilmesine dayanır. Dinamik maskeleme ise her epokta her input için farklı tokenleri maskeler. Dropout tekniği [41] nöral sinir ağı modelinin aşırı öğrenmesini engellemesi için kullanılan bir regülarizasyon tekniğidir. Dropout sinir ağı katmanlarındaki aktivasyonları belirlenen olasılıkla düşürür, başka bir değişle sıfırla değiştirir. Dropout ve dinamik maskeleme karşılaştırıldığında birbirlerinden çok farklı olmadığı görülecektir. Model kapasitesi, eğitim verisi boyutu ve dinamik maskelemeden gelen regülarizasyon etkisi beraber değerlendirildiğinde ayrıca dropout uygulamanın aşırı regülarizasyona sebep olabileceği hipotezlenmiştir. Model ön eğitiminde dropout kullanılmamıştır ve model 8 epok boyunca validasyon setindeki kayıp fonksyionu değerini iyileştirmiştir. Bu bulgu aşırı öğrenme olmadığını gösterir ve kurulan hipotezi desteklemektedir. Aşırı regülarizasyon eğitim süresini uzatacağı için ön eğitim gibi maliyetli bir aşamada bunun önüne geçmek çok önemlidir.



Şekil 3.3 Dropout tekniği

3.3 IR İçin İnce Ayar Eğitimi

ColBERT paradigması potansiyeline rağmen yapılan çalışma sayısı açısından temsil tabanlı modellerin gerisinde kalmıştır. Kısıtlı sayıdaki çalışmaya rağmen geliştirilen ilk ColBERT modelinden bu yana farklı eğitim stratejileri denenmiş ve ilk versiyona göre ciddi ilerlemeler kaydedilmiştir. Hazırlık kısmında bahsedilen çok dilli varyantlara ek olarak Japonca dili için geliştirilen JaColBERT [42 ,43]

çalışmaları ColBERT'in farklı dillere uyarlanması açısından iyi bir referans noktasıdır. Bu bölümde farklı çalışmalarda elde edilen bulgular ile modeli geliştirken alınan kararlar açıklanacaktır.

3.3.1 Eğitim Objektifinin Seçimi

ColBERT eğitim objektifi olarak kullanılan iki ana yöntem vardır: contrastive öğrenme ve bilgi damıtma. Contrastive öğrenmede triplet olarak da adlandırılan sorgu-pozitif pasaj-negatif pasaj üçlüleri kullanılır. Model sorgu ve pozitif pasaj arasındaki ilişki skorunu arttırmaya çalışırken diğer yanda sorgu ve negatif pasajlar arasındaki ilişki skorunu azaltmaya çalışır. Tripletlerde kullanılan negatif örneklerin yanıltıcı olması yani sorguya benzer olması daha öğretici sinyaller sağlamaktadır. Bu tür zor örnekler zor negatif (hard negative) olarak adlandırılır. Zor negatif örnekler önceden eğitilmiş veya eğitim tabanlı olmayan modeller aracılığıyla bulunur. Çok büyük verisetlerinde benzerlik araması fizibıl olmaktan çıkacağı için tipik olarak indeksleme tekniğiyle arama işlemi gerçekleştirilir. Basit şekilde contrastive öğrenme objektifini incelediğimizde modele sağlıklı geri bildirim yapılmadığını görebiliriz. Örnek ile açıklarsak; araba ile kamyon, motor ve bisikletin arasındaki benzerliği öğrenmeye çalıştığımızı varsayalım. Araba ve kamyonun pozitif çift olduğunu düşünürsek araba-motor, araba-bisiklet çiftleri ise negatif çiftler olur. Contrastive öğrenme perspektifinde araba-motor arasındaki benzerlikle araba-bisiklet arasındaki benzerliğin bir farkı yoktur. Pozitif-negatif ilişki aslında ikili ilişki türüne girmektedir ve ikili ilişki sürekli ilişkiye göre daha az bilgilendiricidir. Aynı örneği sürekli ilişkiyle yorumlarsak; araba-kamyon için 0.80, araba-motor için 0.40, araba-bisiklet için 0.20 benzerlik değerleri olduğunu söyleyebilirdik. Sürekli ilişki yaklaşımı bize arabanın motora bisikletten daha çok benzediğini ifade edebilmektedir ki bu ayrım ikili ilişki temsiliyle yapılamaz. Contrastive öğrenmede kullanılamayan bu sürekli ilişki temsili bilgi damıtmada kullanılabilmektedir. Bilgi damıtmada contrastive öğrenmedeki tripletlerdeki pozitif-negatif ayrımı kaldırılarak yerine sorgu-pasaj arasındaki sürekli ilişkiyi temsil eden skor değerleri getirilir. Öğretmen modeller aracılığıyla üretilen sürekli ilişki değerleri öğrenci modele öğretilebilmektedir. İlk ColBERT çalışmasında constrastive öğrenme kullanılmıştır ve ColBERTv2 [44] adlı çalışmada bilgi damıtma kullanılmıştır. Açıklanan örnekten sonra tahmin edilebileceği üzere bilgi damıtma eğitim objektifi üstün başarı göstermiştir. Gösterilen üstün performans ve

zengin ilişki temsili geliştirilecek model için bilgi damıtma eğitim objektifinin seçilmesinin nedenleridir.

$$\text{KLDivLoss}(y_{\text{pred}}, y_{\text{true}}) = y_{\text{true}} \cdot \log \frac{y_{\text{pred}}}{y_{\text{true}}} \quad (1)$$

$$\text{LogSoftmax}(x_i) = \log\left(\frac{\exp(x_i)}{\sum_j \exp(x_j)}\right) \quad (2)$$

$$\text{Training Loss} = \text{KLDivLoss}\left(\text{LogSoftmax}(\text{studentScores}), \text{LogSoftmax}(\text{teacherScores})\right) \quad (3)$$

Şekil 3.4 Bilgi damıtma eğitim objektifi

3.3.2 Eğitim Verisetlerini Oluşturma

İnce ayar aşaması için üç farklı verisetinden bilgi damıtma koleksiyonu [45] oluşturulmuştur. Bilgi damıtmada öğretmen modelin yanlılıklarından kurtulmak için aynı örneğin birden fazla öğretmenle skorlandırılması kullanılan bir yöntemdir. Birden fazla öğretmenle elde edilen skorların ortalaması alınarak daha yansız bir skor elde edilebilir. Bahsedilen yaklaşım veriseti oluşturma maliyetini kullanılan öğretmen sayısı kadar arttıracaktır. Maliyeti az tutup model yanlılıklarından kurtulmak için tez kapsamında oluşturulan üç verisiti için iki öğretmen model kullanılmıştır. Böylelikle veriseti oluşturma maliyeti sabit tutularak model yanlılıklarının önüne geçilmesi amaçlanmıştır.

3.3.2.1 Türkçe MSMarco

Bu veriseti [46] MS Marco verisetinin makine tercümesiyle Türkçe'ye uyarlanmış halidir. MS Marco verisetinin tamamı kullanılmamış olup 718 bin pasaj ve 603 bin sorgu kullanılmıştır. Verisetinin bilgi damıtmaya uyarlanması için ilk aşamada zor negatif örneklerin bulunması gerekmektedir. Zor negatif örnekleri bulmak için öncelikle temsil tabanlı ayarda bge-m3 modeliyle sorgu ve pasajların vektör temsilleri elde edilmiştir. 603 bin sorgu vektörünün 718 bin pasaj vektörüyle karşılaştırılması hesaplama maliyeti açısından mümkün olmadığı için ANN yöntemine başvurulmuştur. Spotify Annoy python kütüphanesi ile vektörler indekslenmiş ve her sorgu için en benzer 9 pasaj benzerlik araması ile bulunmuştur. Exact nearest neighbor search ile 400 saat sürecek işlem ANN ile 1 saatte

tamamlanmıştır. Bulunan 9 pasaj ve verisetinde önceden etiketlenmiş 1 adet doğru pasaj olarak toplamda onar adet sorgu-pasaj benzerliği bge-m3-reranker-v2 [47] etkileşim tabanlı modeliyle elde edilmiştir.

3.3.2.2 WebFAQ Retrieval Türkçe Kümesi

Günümüz QA sistemleri genellikle IR tekniklerini içerdiğinden, QA veri setleri IR alanı için de faydalı olabilir ve bu alandaki eğitimlerde yaygın şekilde kullanılmaktadır. WebFAQ Soru-Cevap Veri Seti [48], internet üzerindeki FAQ (Frequently Asked Questions - Sıkça Sorulan Sorular) sayfalarından toplanmış, 75 dili kapsayan 96 milyon doğal soru-cevap çiftinden oluşan geniş çaplı bir koleksiyondur. Bu veri seti, schema.org'un FAQPage yapılandırılmış açıklamalarını kullanarak oluşturulmuş ve bu sayede büyük ölçekli QA araştırmaları için değerli bir kaynak haline gelmiştir. Çok dilli kapsamı ve düzenli yapısıyla dikkat çeker.

WebFAQ Retrieval Veri Seti [49] ise bu büyük veri setinin özenle seçilmiş, filtrelenmiş bir alt kümesidir ve özellikle IR görevlerini desteklemek amacıyla tasarlanmıştır. Bu alt küme, IR tabanlı araştırmalar ve uygulamalar için yüksek kaliteli ve hedefe yönelik veriler sunar. Verisetinde 145 bin Türkçe sorgu-cevap çifti bulunmaktadır.

WebFAQ Retrieval verisetinin Türkçe alt kümesi için zor negatif örnek bulma işlemi aşağıdaki şekilde yapılmıştır:

- Ön eğitimini yaptığımız ModernBERT modeli Türkçe MS Marco ile oluşturulan bilgi damıtma veri setinde eğitilmiştir.
- Eğitilen model ve HSNW ANN yöntemi kullanılarak her sorgu için 9 adet zor negatif elde örnek edilmiştir.

İlişki skorlandırılmasında jina-multilingual-reranker-v2 [50] etkileşim tabanlı modeli kullanılmıştır.

3.3.2.3 Türkçe Karma Soru-Cevap Veriseti

Bu koleksiyon [51] farklı açık kaynak Türkçe Soru-Cevap verisetilerinin birleştirilmesi ve filtrelenmesi ile oluşturulmuştur. 82 bin adet soru-cevap-bağlam üçlüsü içermektedir. Verisetindeki sorularun cevapları pasaj sıralama ile bulunamayacağı için soru-bağlam çiftleri sorgu-pasaj olarak ele alınmıştır. Zor

negatif bulma işleminde jina-colbert-v2 modeli ve HSNW ANN yöntemi kullanılmıştır. Diğer verisetilerinde uygulandığı gibi 9 adet zor örnek bulunmuştur. Bulunan zor örneklerle sorgular arasındaki ilişki skoru yine jina-multilingual-reranker-v2 modeli ile hesaplanmıştır.

3.3.3 Eğitim Aşaması

Oluşturulan bilgi damıtma veri koleksiyonuyla toplamda 730 bine yakın sorgu, her sorgu için 10 adet pasaj ve ilişki skoru elde edilmiştir. Ön eğitim aşamasında geliştirilen model, PyLate [51] çerçevesi kullanılarak bilgi damıta koleksiyonu üzerinde 4 epok boyunca eğitilmiştir. 128 batch boyutu, 0.10 dropout oranı, 8e-5 öğrenme oranı, lineer öğrenme oranı zamanlayıcısı kullanılmıştır. Her epok sonunda model ağırlıkları kaydedilmiştir.

Deney aşamasında diğer çalışmalarda uygulanıp iyi sonuçlara yol açmış teknikler denenmiştir. Aşağıda listelenen yöntemlerden hiçbiri değerlendirme verisetindeki başarıyı kayda değer şekilde arttırmamıştır:

- İlk ColBERT çalışmasında ortaya konan query expansion tokenlerinin kullanılması.
- Jina-colbert-v2 çalışmasında uygulanan query expansion tokenlerinin tüm tokenler için attention hesaplamasına dahil edilmesi (attend to query expansion tokens).
- Öğretmen skor normalizasyonu [52].
- Ön eğitim aşamasında öğrenilen temsilin korunması için modelin ilk katmanlarının dondurulması.

4
DEĞERLENDİRME

Bu bölümde eğitimi tamamlanmış ColBERT modelinin diğer çok dilli ColBERT varyantlarıyla karşılaştırılması yapılmıştır. Etkileşim ve temsil tabanlı modeller kullanılabilirlik açısından farklı potansiyellere sahip olduğu için adil bir değerlendirme olması açısından ele alınmamıştır.

4.1 Değerlendirme Verisetlerinin Oluşturulması

Tez kapsamında Türkçe ColBERT modeli geliştirildiği için oluşturulacak değerlendirme verisetlerinin ve kullanılacak değerlendirme yöntemlerinin düşük kaynaklı dillerde karşılaşılan engelleri ele alması gerekmektedir. Düşük kaynak dillerdeki en önemli sorun eğitim verisi bulmanın veya üretmenin zorluğudur. Bu yüzden yapılacak değerlendirmenin IR modellerinin alan dışı yeteneklerini ölçmesi gerekmektedir. Eğer yapılacak değerlendirme alan içi değerlendirme olsaydı geliştirilen Türkçe ColBERT modeli ile diğer çok dilli varyantlar arasındaki "domain gap" (modellerin değerlendirme verisine benzer veriyle eğitilmiş olmaması) değerlendirmeyi adil olmaktan çıkarırdı. Alan içi değerlendirmenin adil olabilmesi için çok dilli varyantların kullandığımız eğitim verisiyle eğitilmesi gerekirdi. Model eğitiminde denenecek çok fazla yöntemin olması ve eğitimin model sahipleri tarafından yapılmaması sürecin sağlıklı yürütülmesi önündeki önemli engellerdir.

Literatürdeki çalışmalarda büyük ölçekli değerlendirme verisetleriyle model geliştirilmesinin pratik ve mümkün olmadığını ele alan örnekler vardır. Örneğin BEIR değerlendirmesi günlerce sürebilmektedir. Bu kısıtlamadan kurtulmak için BEIR'in küçük örneklemleriyle validasyon yapılması son çalışmalarda yaygın

olarak görülmektedir. Jina-colbert-v2 çalışmasında bu tarz küçük örneklemlerle elde edilen değerlendirme sonuçlarının ana verisetiyle elde edilen sonuçlarla paralellik gösterdiği, tutarlı olduğu belirtilmiştir. Bu yöntemin sektördeki kabulune yönelik verilebilecek bir örnek: en popüler embedding modeli geliştirme çerçevelerinden biri olan Sentence Transformers [53] çerçevesinde söz konusu değerlendirme yönteminin uygulanmaş olmasıdır.

Pasaj sıralama için kullanılacak veriler çeşitli yöntemlerle oluşturulabilir. En yaygın yöntemlerden biri insan anotasyonudur. Bu yöntemde, uzmanlar veya crowdsourcing platformlarındaki (örneğin Amazon Mechanical Turk) anotatörler, belirli bir sorguya karşılık gelen pasajları değerlendirerek uygunluklarına göre puanlar veya sıralar. İnsan anotasyonu yüksek doğruluk sağlar ancak maliyetli ve zaman alıcıdır; ayrıca anotatörler arası tutarsızlık gibi problemler de olabilir. Bir diğer yöntem makine çevirisidir. Mevcut İngilizce veri kümeleri, otomatik çeviri sistemleriyle diğer dillere çevrilerek pasaj sıralama verisi oluşturulabilir. Bu yöntem hızlı ve ucuzdur ancak çeviri hataları, anlam değişmesi gibi sorunlara neden olabilir. Makine çevirisinde oluşan tercüme hataları, pasaj sıralama görevinde model performansının düşük çıkmasına neden olsa da model seçimini ciddi şekilde etkilemeyebilir. Bunun temel nedeni, her bir sorgunun binlerce aday pasajla karşılaştırılmasıdır. Bu kadar büyük bir aday havuzunda, çeviri hatasına rağmen doğru cevabın rastlantısal olarak üst sıralarda yer alması oldukça zordur. Son dönemde yaygınlaşan bir diğer yöntem ise üretken büyük dil modelleri destekli veri oluşturmadır. Bu yaklaşımda, bir LLM belirli bir pasaj için uygun soruları üretir ya da verilen sorgu-pasaj çiftlerini derecelendirir. LLM'ler ayrıca zayıf etiketli (weakly-supervised) veriyi filtreleyerek veya veri zenginleştirme (data augmentation) için kullanılarak veri kalitesini artırabilir. Bu yöntem anotasyon maliyetini düşürürken, LLM'in eğitildiği dağılımın dışına çıktığında tutarsız sonuçlar üretebilmesi gibi riskleri de barındırır. LLM destekli yöntemler makine çevirisine göre daha başarılı olsa da daha maliyetlidirler. Özetle, insan anotasyonu yüksek doğruluk ve kalite sunarken pahalıdır; makine çevirisi daha ucuz ancak çeviri hatalarına açıktır; LLM destekli yöntemler ise hızlı ve esnektir fakat dikkatli doğrulama gerektirir. Bu yöntemler bazen bir arada da kullanılabilir; örneğin, LLM ile oluşturulan veriler sonradan insanlar tarafından gözden geçirilerek hibrit bir yaklaşım izlenebilir.

Gerçekleştirilen tez çalışması bireysel kaynaklar ile yapıldığı için en ucuz ve hızlı yöntem olan yabancı verisetlerinin Türkçe'ye tercüme edilmesi tercih edilmiştir. Milyonlarca örnek içeren karşılaştırmaların pratik ve mantıklı olmamasından dolayı uygulanabilir büyüklükte bir karşılaştırma koleksiyonu olan NanoBEIR [54] Türkçe'ye çevrilmiştir. NanoBEIR, BEIR koleksiyonundan 13 verisetinin küçük örneklemlerinden oluşmuştur. NanoBEIR'de her veriseti için ortalama 50 sorgu ve 5 bin pasaj bulunmaktadır. Alan dışı değerlendirmenin ölçülmesi amaçlandığı için eğitim verisinde bulunan MS Marco veriseti Türkçe NanoBEIR'e dahil edilmemiştir ve kalan 12 verisetiyle değerlendirme yapılmıştır. Örnekler Türkçe'ye tercüme edilirken açık kaynaklı OpusMT [55] modeli kullanılmıştır. Veri koleksiyonundaki bazı örnekler makine tercümesi modelinin bağlam uzunluğunu aştığı için örnekler NLTK [56] kütüphanesi aracılığıyla cümlelere ayrılıp tercüme edilmiştir.

Tablo 4.1 Türkçe NanoBEIR içeriği

Veriseti Adı	*#Pasaj*	*#Alakalı Çift*	*#Sorgu*	*Konu*
FiQA2018	4598	123	50	Finans
ClimateFEVER	3408	148	50	İklim değişikliği
DBPedia	6045	1158	50	Varlık erişimi
FEVER	4996	57	50	Wikipedia QA
HotpotQA	5090	100	50	Wikipedia QA
NFCorpus	2953	2518	50	Medikal
NQ	5035	57	50	Açık alan QA
Quora	5046	70	50	Quora soruları
Scidocs	2210	244	50	Bilim
ArguAna	3635	50	50	Tartışma
SciFact	2919	56	50	Bilim
Touche2020	5745	932	49	Argüman erişimi
Toplam	**51680**	**5513**	**599**	**-**

4.2 Değerlendirme Sonuçları

Geliştirilen modelin ince ayar aşamasında her epok sonu kaydedilen 4 model ağırlığı ve hazırlık kısmında değinilen çok dilli ColBERT varyantları Türkçe NanoBEIR koleksiyonunda test edilmiştir. Başarı metriklerinin hesaplanmasın PyTrec [57] kütüphanesi kullanılmıştır. Gerçekleştirilen değerlendirme sonucunda ince ayar aşamasında kaydedilen 4 epoktan 2.si en iyi performansı göstermiştir. Modelin 2 epok boyunca tek bir Nvidia 6000 Ada Generation ekran kartıyla eğitimi yaklaşık olarak 6 saat sürmüştür.

Tablo 4.2 Türkçe NanoBEIR nDCG@10 sonuçları

Veriseti Adı	*TrColBERT*	*Jina ColBERTV2*	*ColBERT-XM*	*BGE-M3 (ColBERT)*
FiQA2018	**0,492216**	0,412139	0,398984	0,460835
ClimateFEVER	**0,292758**	0,253207	0,253512	0,288291
DBPedia	0,635947	**0,681396**	0,623168	0,559665
FEVER	**0,919845**	0,903199	0,857440	0,847881
HotpotQA	0,850588	**0,886873**	0,724111	0,809577
NFCorpus	**0,336974**	0,310284	0,326435	0,334395
NQ	0,555577	**0,650597**	0,528781	0,632394
Quora	**0,914262**	0,902941	0,862832	0,899617
Scidocs	0,299677	**0,340469**	0,284012	0,314170
ArguAna	0,449078	0,460118	0,357842	**0,540127**
SciFact	**0,785042**	0,725386	0,664517	0,695833
Touche2020	0,560338	0,538785	**0,562243**	0,512609
Ortalama nDCG@10	**0,590275**	0,588783	0,536990	0,574616

Tablo 4.3 Türkçe NanoBEIR detaylı nDCG sonuçları

Metrik Adı	*TrColBERT*	*Jina ColBERTV2*	*ColBERT-XM*	*BGE-M3 (ColBERT)*
NDCG@5	0,583429	**0,588929**	0,520791	0,567915
NDCG@10	**0,590275**	0,588783	0,536990	0,574616
NDCG@20	**0,597864**	0,593854	0,543184	0,583148
NDCG@30	**0,603381**	0,596876	0,547657	0,589781
NDCG@100	**0,627595**	0,625387	0,575031	0,617902
NDCG	**0,676968**	0,674777	0,628144	0,666384
R-NDCG	**0,582446**	0,574616	0,522139	0,565785

Tablo 4.4 Türkçe NanoBEIR detaylı Recall sonuçları

Metrik Adı	*TrColBERT*	*Jina ColBERTV2*	*ColBERT-XM*	*BGE-M3 (ColBERT)*
Recall@5	0,532639	**0,541200**	0,475013	0,531456
Recall@10	**0,599148**	0,594395	0,563935	0,594291
Recall@15	0,639484	0,631323	0,604221	**0,641566**
Recall@20	**0,668367**	0,661739	0,634631	0,663574
Recall@30	**0,703873**	0,690362	0,665878	0,699041
Recall@100	0,792802	0,794043	0,767508	**0,800131**
Recall@200	0,834245	0,843299	0,823716	**0,849130**
Recall@500	0,890959	0,894451	0,873777	**0,899619**
Recall@1000	0,933093	0,931562	0,916275	**0,936651**

Tablo 4.5 Türkçe NanoBEIR detaylı Precision sonuçları

Metrik Adı	*TrColBERT*	*Jina ColBERTV2*	*ColBERT-XM*	*BGE-M3 (ColBERT)*
Precision@5	0,275667	**0,281605**	0,257333	0,268571
Precision@10	**0,192323**	0,191939	0,186673	0,184272
Precision@15	**0,154912**	0,153998	0,149705	0,148102
Precision@20	**0,133281**	0,131248	0,127197	0,127158
Precision@30	**0,103324**	0,100798	0,098998	0,100255
Precision@100	0,044005	**0,044185**	0,042927	0,044156
Precision@200	0,025838	0,026196	0,025095	**0,026228**
Precision@500	0,012269	0,012272	0,011811	**0,012374**
Precision@1000	0,006982	0,006966	0,006766	**0,007021**
R-Precision	**0,464098**	0,448508	0,393998	0,446627

Tablo 4.6 Türkçe NanoBEIR detaylı MAP sonuçları

Metrik Adı	*TrColBERT*	*Jina ColBERTV2*	*ColBERT-XM*	*BGE-M3 (ColBERT)*
MAP@5	**0,446353**	0,444746	0,383537	0,434125
MAP@10	**0,469814**	0,466482	0,410466	0,455423
MAP@15	**0,481379**	0,477790	0,422366	0,466485
MAP@20	**0,488910**	0,484984	0,429238	0,473149
MAP@30	**0,496953**	0,492175	0,436709	0,481389
MAP@100	**0,509805**	0,505889	0,449985	0,495681
MAP@200	**0,512928**	0,509351	0,435087	0,499310
MAP@500	**0,514672**	0,510980	0,454533	0,500836
MAP@1000	**0,515306**	0,511580	0,455132	0,501460
MAP	**0,516209**	0,512473	0,456107	0,502371

- Tablo 4.2'de BEIR için literatürde standart olarak kullanılan değerlendirme metriği nDCG@10 değerleri görülebilir. Geliştirilen model 12 verisetinden 7'sinde ve genel ortalamada diğer çok dilli varyantları geride bırakmıştır.
- Tablo 4.3 ve 4.6'da, sıralamaya bağlı olan nDCG ve MAP metriklerinin biri hariç hepsinde geliştirilen model diğer modelleri geride bırakmıştır.
- Tablo 4.4 ve 4.5'te sıralamadan bağımsız Precision ve Recall değerleri incelendiğinde herhangi bir modelin diğerlerine kesin üstünlüğü görülememektedir.
- İlişkili sorgu-pasaj sayısını dikkate alan MAP, R-NDCG ve R-Precision metriklerinde geliştirilen model diğer modellerden anlamlı şekilde üstündür.
- Genel bir değerlendirme olarak ColBERT-XM hariç diğer modeller birbirine yakın performans göstermiştir.

4.2.1 Çıkarım Yapma Verimliliğinin Değerlendirilmesi

Pasaj sıralama görevinde erişim hızı, indeksin kapladığı alan ve "kelime başına token oranı" gibi teknik detaylar, sistemin hem performansı hem de ölçeklenebilirliği açısından son derece kritiktir.

Öncelikle erişim hızı, bir sorguya karşılık gelen en iyi pasajların ne kadar hızlı kullanıcıya sunulduğunu gösterir. Özellikle kullanıcıya gerçek zamanlı yanıt verilmesi gereken arama sistemlerinde bu süre milisaniyelerle ölçülür. Yavaş erişim, kullanıcı deneyimini olumsuz etkiler ve sistemin kullanılabilirliğini sınırlar.

İndeksin kapladığı alan, özellikle büyük doküman koleksiyonlarında önemli hale gelir. Modern pasaj erişim sistemleri genellikle milyonlarca pasajı temsil eden embeddingleri saklar. Bu vektörlerin bellekte veya diskte tuttuğu yer ne kadar fazlaysa, sistemin çalışması o kadar maliyetli ve yavaş olur. Küçük ve verimli indeksler hem daha hızlı yüklenir hem de daha az donanım kaynağı gerektirir.

Multi-vector erişim sistemlerinde bir metin ne kadar az tokenle temsil edilebilirse oluşturulacak indeks o kadar küçük olur ve hızlı çalışır. Bu yüzden "kelime başına düşen token sayısını gösteren" kelime başına token oranı çok önemlidir. ColBERT paradigmasında n sorgu vektörü m pasaj vektörüyle karşılaştırılır, bu işlemde n çarpı m adet vektör benzerliği hesaplanması gerekmektedir. Dolayısıyla, sorgu ve pasajlardaki token sayısının X kadar azaldığı durumda yapılacak vektör benzerliği hesaplama sayısı X'in karesi kadar azalmış olur.

Değerlendirilen modellerin sadece erişim başarı açısından değerlendirilmesi gerçek hayat koşullarını görmezden gelmek demek olur. Erişim başarısı, hızı ve maliyet dengesizliklerini yorumlaması için modellerin ortalama "kelime başına token" oranı 45MB boyutundaki Türkçe Wikipedia paragraflarında hesaplanmıştır. Bu oranı hesaplamak için öncelikle söz konusu paragraf verisetindeki kelime sayısı NLTK kütüphanesinin kelime ayrıştırıcısıyla hesaplanmıştır. Sonrasında aynı paragraf verisi, modellerin tokenizerlarıyla tokenlere çevrilmiştir ve token sayıları hesaplanmıştır. Son olarak bulunan token sayısıları kelime sayısına bölünerek "kelime başına token" oranı hesaplanmıştır. Jina Colbert-v2, ColBERT-XM ve BGE-M3 aynı XLM-RoBERTa tokenizerına sahip olduğu için oranları eşit çıkmıştır. Tablo 4.7'de görülebileceği üzere dil spesifik tokenizera sahip modelimiz

tokenizasyon konusunda daha verimlidir. Ek olarak tabloda çok dilli modellerin parametre sayısı açısından dezavantaja sahip olduğu görülebilir.

Tablo 4.7 Modellerin verimliliğinin karşılaştırılması

	TrColBERT	*Jina ColBERTV2*	*ColBERT-XM*	*BGE-M3*
Model Parametre Sayısı	**135 milyon**	559 milyon	853 milyon (270 milyon aktif)	560 milyon
Kelime Başına Token Oranı	**1,423**	1,592	1,592	1,592

5
SONUÇ

5.1 Tez Sonucu

Türkçe için yalnızca açık kaynak verilerle ve 6 saatlik sınırlı ince ayar süresiyle çok dilli ColBERT modelleriyle rekabet edebilen, model boyutu küçültülmüş ve tokenizasyon açısından verimli bir ColBERT modeli geliştirmek, hem teknik hem stratejik açıdan önemlidir. Türkçe, İngilizce kadar büyük ve yüksek kaliteli veri kümesine sahip olmayan bir dil olduğundan, bu tür verimli modeller, düşük kaynaklı dillerin bilgi erişim kalitesini artırma konusunda büyük katkı sağlar. Açık veriyle eğitilen hafif ve optimize modeller sayesinde, Türkçe içeriklerin semantik arama ve bilgi çıkarımındaki etkinliği önemli ölçüde yükseltilebilir. Buna ek olarak, böyle bir modelin açık kaynak olarak sunulması, Türkçe IR araştırmalarına ciddi ivme kazandırabilir. Yerli, hafif ve verimli ColBERT türevleri; üniversitelerdeki araştırmacılar, şirketler ve geliştiriciler için erişilebilir bir altyapı sunarak, Türkçe dijital ekosistemde daha güçlü ve bağımsız arama çözümlerinin önünü açar.

Eğer kaynaklar artırılırsa, yani daha fazla yüksek kaliteli Türkçe veri, daha uzun fine-tuning süresi, daha yüksek model kapasiteleri ve daha güçlü donanım kullanılırsa, Türkçe ColBERT modellerinin performansı ciddi ölçüde artabilir. Bu, özellikle sıralama doğruluğu, uzun sorgularla başa çıkma kabiliyeti, domain adaptasyonu (örneğin sağlık, hukuk gibi alanlar) ve kullanıcı taleplerine göre özelleştirme gibi açılardan büyük avantaj sağlar. Model, çok dilli muadillerine kıyasla daha derin dil anlayışı kazanarak, daha isabetli ve bağlama duyarlı sonuçlar üretebilir. Dahası, güçlü bir yerel IR modeli; Türkiye'deki kurumlar için veri egemenliğini koruyarak, dışa bağımlılığı azaltır ve yerli yapay zeka altyapılarının gelişimine doğrudan katkı sağlar. Bu da Türkiye’nin dijital bağımsızlık ve yapay zekada rekabet gücü açısından stratejik bir avantaj elde etmesine yardımcı olur.

5.2 Gelecek Çalışmalar

Yüksek lisans seviyesinde bireysel kaynaklarla gerçekleştirilen bir tez çalışmasında malesef alandaki tüm boşluklar doldurulamamıştır. Dolayısıyla gerçekleştirilen tez çalışmasında en az maliyetle en iyi sonuçlar elde edilmeye çalışılmıştır. Türkçe spesifik ColBERT geliştirme, literatürde ve sektörde yeterince çalışılmamış bir konu olduğu için yüksek potansiyelli çok sayıda araştırma konusu bulunmaktadır. Gelecekteki çalışmaların ele alabileceği konular:

- **Türkçe spesifik encoder modelleri için ön eğitim stratejilerinin karşılaştırılması:** ColBERT geliştirmenin ilk adımı bir dil modelinin ön eğitimidir. Dil modeli ön eğitimi için yabancı dillerde bulunan en iyi yöntemlerin Türkçe'ye transferi başarı garantisi vermemektedir. Yöntem kısmında detaylıca açıklandığı üzere ön eğitim objektifleri, hiperparametreler ve model dizaynında Türkçe'de en iyi sonucu verecek kararların alınması sadece Türkçe IR başarısını arttırmakla kalmayıp birçok NLP görevinde Türkçe çalışmaları ileriye götürecektir.
- **Büyük tokenizer kullanmak:** Türkçede eklemeli yapı nedeniyle aynı kökten birçok varyasyon oluşur, bu da küçük tokenizerlarda kelimelerin çok sayıda tokene bölünmesine yol açar. Gerçekleştirdiğimiz çalışmada kullanılan 32 bin tokene sahip tokenizerın aksine daha çok tokene sahip büyük bir tokenizer, kelime başına token sayısını azaltarak modelin daha kısa dizilerle daha fazla bilgi işlemesini sağlar. Bu, hem hız kazancı hem de bağlamı daha iyi kavrama açısından avantaj sağlayabilir.
- **Eğitim ve değerlendirme verisetlerini büyütmek ve iyileştirmek:** Yapılan çalışmada kullanılan ön eğitim ve ince ayar verisetleri web kazıma ve makine çevirisi karışımından oluşmaktadır. Veri kalitesini arttırmak için öncelikle otomatik veri temizleme konusunda çalışmalar yapılabilir. Daha kapsamlı çalışmalar olarak insan anotasyonu ve LLM desteğiyle daha büyük, daha kaliteli eğitim ve değerlendirme verisetleri oluşturulabilir. Geliştirilecek verisetlerinin Türk kültürünü daha iyi yansıtması gerçek hayat koşullarının daha iyi temsil edilmesini sağlayacaktır.

KAYNAKÇA

[1] O. Khattab and M. Zaharia, “ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT,” arXiv:2004.12832 [cs], Jun. 2020, Available: https://arxiv.org/abs/2004.12832

[2] “spotify/annoy,” GitHub, Oct. 30, 2020. https://github.com/spotify/annoy

[3] Y. A. Malkov and D. A. Yashunin, “Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs,” arXiv:1603.09320 [cs], Aug. 2018, Accessed: Feb. 26, 2022. [Online]. Available: https://arxiv.org/abs/1603.09320

[4] A. Vaswani et al., “Attention is All you Need,” Neural Information Processing Systems, 2017. https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proceedings of the 2019 Conference of the North, vol. 1, pp. 4171–4186, 2019, doi: 10.18653/v1/n19-1423.

[6] Y. Liu et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” arXiv.org, Jul. 26, 2019. https://arxiv.org/abs/1907.11692

[7] A. Conneau et al., “Unsupervised Cross-lingual Representation Learning at Scale,” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, doi: 10.18653/v1/2020.acl-main.747.

[8] A. Conneau et al., “XNLI: Evaluating Cross-lingual Sentence Representations,” Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, doi: 10.18653/v1/d18-1269.

[9] P. Lewis, B. Oguz, R. Rinott, S. Riedel, and H. Schwenk, “MLQA: Evaluating Cross-lingual Extractive Question Answering,” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, doi: 10.18653/v1/2020.acl-main.653.

[10] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with Disentangled Attention,” arXiv:2006.03654 [cs], Mar. 2021, Available: https://arxiv.org/abs/2006.03654

[11] J. Su, Y. Lu, S.-F. Pan, B. Wen, and Y. Liu, “RoFormer: Enhanced Transformer with Rotary Position Embedding,” Apr. 2021, doi: 10.48550/arxiv.2104.09864.

[12] B. Warner et al., “Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference,” arXiv.org, 2024. https://arxiv.org/abs/2412.13663

[13] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, “Learning deep structured semantic models for web search using clickthrough data,” Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM ’13, 2013, doi: 10.1145/2505515.2505665.

[14] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, “A Deep Relevance Matching Model for Ad-hoc Retrieval,” Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM ’16, 2016, doi: 10.1145/2983323.2983769.

[15] A. Louis, V. Saxena, G. Van Dijck, and G. Spanakis, “ColBERT-XM: A Modular Multi-Vector Representation Model for Zero-Shot Multilingual Information Retrieval,” 2025. Accessed: Jul. 16, 2025. [Online]. Available: https://aclanthology.org/2025.coling-main.295.pdf

[16] J. Pfeiffer et al., “Lifting the Curse of Multilinguality by Pre-training Modular Transformers,” Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Jan. 2022, doi: 10.18653/v1/2022.naacl-main.255.

[17] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, “BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation,” arXiv.org, 2024. https://arxiv.org/abs/2402.03216

[18] K. Lo, L. L. Wang, M. Neumann, R. Kinney, and D. Weld, “S2ORC: The Semantic Scholar Open Research Corpus,” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, doi: 10.18653/v1/2020.acl-main.447.

[19] Niklas Muennighoff et al., “Crosslingual Generalization through Multitask Finetuning,” arXiv (Cornell University), Jan. 2023, doi: 10.18653/v1/2023.acl-long.891.

[20] L. Xue et al., “mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer,” ACLWeb, Jun. 01, 2021. https://aclanthology.org/2021.naacl-main.41/

[21] G.-B. Ho, C. Tan, Z. Darban, M. Salehi, R. Haf, and W. Buntine, “MTP: A Dataset for Multi-Modal Turning Points in Casual Conversations,” Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 314–326, 2024, doi: 10.18653/v1/2024.acl-short.30.

[22] NLLB Team et al., “No Language Left Behind: Scaling Human-Centered Machine Translation,” arXiv.org, Aug. 25, 2022. https://arxiv.org/abs/2207.04672

[23] H. Schwenk, Guillaume Wenzek, Sergey Edunov, E. Grave, A. Joulin, and A. Fan, “CCMatrix: Mining Billions of High-Quality Parallel Sentences on the

Web," arXiv (Cornell University), Jan. 2021, doi: 10.18653/v1/2021.acl-long.507.

[24] Z. Yang et al., "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," arXiv:1809.09600 [cs], Sep. 2018, Accessed: Jul. 19, 2022. [Online]. Available: https://arxiv.org/abs/1809.09600

[25] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer, "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension," Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, doi: 10.18653/v1/p17-1147.

[26] P. Bajaj et al., "MS MARCO: A Human Generated MAchine Reading COmprehension Dataset," arXiv:1611.09268 [cs], Oct. 2018, Available: https://arxiv.org/abs/1611.09268

[27] W. He et al., "DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications," Jan. 2018, doi: 10.18653/v1/w18-2605.

[28] X. Xie et al., "T2Ranking: A large-scale Chinese Benchmark for Passage Ranking," arXiv.org, 2023. https://arxiv.org/abs/2304.03679# (accessed Jul. 16, 2025).

[29] X. Zhang, X. Ma, P. Shi, and J. Lin, "Mr. TyDi: A Multi-lingual Benchmark for Dense Retrieval," arXiv (Cornell University), Jan. 2021, doi: 10.18653/v1/2021.mrl-1.12.

[30] X. Zhang et al., "MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages," Transactions of the Association for Computational Linguistics, vol. 11, pp. 1114–1131, Jan. 2023, doi: 10.1162/tacl_a_00595.

[31] H. Xiao, B. Wang, and R. Jha, "Jina-ColBERT-v2: A General-Purpose Multilingual Late Interaction Retriever," pp. 159–166, Jan. 2024, doi: 10.18653/v1/2024.mrl-1.11.

[32] G. Penedo et al., "The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only," arXiv.org, 2023. https://arxiv.org/abs/2306.01116 (accessed Jul. 27, 2023).

[33] "jinaai/jina-reranker-v2-base-multilingual · Hugging Face," Huggingface.co, Sep. 17, 2024. https://huggingface.co/jinaai/jina-reranker-v2-base-multilingual

[34] T. Kwiatkowski et al., "Natural Questions: A Benchmark for Question Answering Research," Transactions of the Association for Computational Linguistics, vol. 7, pp. 453–466, Mar. 2019, doi: 10.1162/tacl_a_00276.

[35] L. Bonifacio et al., "mMARCO: A Multilingual Version of the MS MARCO Passage Ranking Dataset," arXiv.org, 2021. https://arxiv.org/abs/2108.13897

[36] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Improving Text Embeddings with Large Language Models," arXiv.org, 2023. https://arxiv.org/abs/2401.00368 (accessed Sep. 04, 2024).

[37] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness," arXiv.org, Jun. 23, 2022. https://arxiv.org/abs/2205.14135

[38] Z. Ankner, N. Saphra, D. Blalock, J. Frankle, and M. L. Leavitt, “Dynamic Masking Rate Schedules for MLM Pretraining,” arXiv.org, 2023. https://arxiv.org/abs/2305.15096 (accessed Jul. 17, 2025).

[39] O. Team, “OSCAR 21.09- OSCAR Documentation,” Github.io, 2019. https://oscar-project.github.io/documentation/versions/oscar2109/ (accessed May 30, 2025).

[40] S. Schweter, (2020). BERTurk-BERT models for Turkish. Zenodo, 2020, 3770924.

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” Journal of Machine Learning Research, vol. 15, no. 56, pp. 1929–1958, 2014, Available: https://jmlr.org/papers/v15/srivastava14a.html

[42] B. Clavié, “JaColBERT and Hard Negatives, Towards Better Japanese-First Embeddings for Retrieval: Early Technical Report,” arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2312.16144.

[43] B. Clavié, “JaColBERTv2.5: Optimising Multi-Vector Retrievers to Create State-of-the-Art Japanese Retrievers with Constrained Resources,” arXiv (Cornell University), Jul. 2024, doi: 10.48550/arxiv.2407.20750.

[44] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia, “ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction,” Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Jan. 2022, doi: 10.18653/v1/2022.naacl-main.272.

[45] Tr-IR-KD-Data: https://huggingface.co/datasets/99eren99/Turkish-IR-KD-Data (accessed May 30, 2025).

[46] Parsa Kazerooni, “msmarco-tr,” Huggingface.co, 2016. https://huggingface.co/datasets/parsak/msmarco-tr (accessed May 30, 2025).

[47] “jinaai/jina-reranker-v2-base-multilingual · HuggingFace, ”Huggingface.co, Sep. 17, 2024. https://huggingface.co/jinaai/jina-reranker-v2-base-multilingual

[48] M. Dinzinger, L. Caspari, D. K. Ghosh, J. Mitrović, and M. Granitzer, “WebFAQ: A Multilingual Collection of Natural Q&A Datasets for Dense Retrieval,” arXiv.org, 2025. https://arxiv.org/abs/2502.20936 (accessed May 30, 2025).

[49] Chair, “webfaq-retrieval,” Huggingface.co, 2025. https://huggingface.co/datasets/PaDaS-Lab/webfaq-retrieval (accessed Jul. 20, 2025).

[50] U. Sahin, “TR-Extractive-QA-82K,” Huggingface.co, Jul. 06, 2024. https://huggingface.co/datasets/ucsahin/TR-Extractive-QA-82K (accessed May 30, 2025).

[51] A. Chaffin and R. Sourty, ‘PyLate: Flexible Training and Retrieval for Late Interaction Models’. 2024.

[52] S. Sun, W. Ren, J. Li, R. Wang, and X. Cao, “Logit Standardization in Knowledge Distillation,” 2022 IEEE/CVF Conference on Computer Vision

and Pattern Recognition (CVPR), pp. 15731–15740, Jun. 2024, doi: 10.1109/cvpr52733.2024.01489.

[53] “UKPLab/sentence-transformers,” GitHub, Sep. 04, 2020. https://github.com/UKPLab/sentence-transformers

[54] Tr-NanoBEIR: https://huggingface.co/datasets/99eren99/Tr NanoBEIR (accessed May 30, 2025).

[55] “Helsinki-NLP/opus-mt-tc-big-en-tr · Hugging Face,” Huggingface.co, 2020. https://huggingface.co/Helsinki-NLP/opus-mt tc-big-en-tr (accessed May 30, 2025)

[56] Bird, Steven, Edward Loper and Ewan Klein (2009). Natural Language Processing with Python. O'Reilly Media Inc.

[57] cvangysel, “GitHub - cvangysel/pytrec_eval: pytrec_eval is an Information Retrieval evaluation tool for Python, based on the popular trec_eval.,” GitHub, 2017. https://github.com/cvangysel/pytrec_eval (accessed Jan. 20, 2025).

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirileri

1. E.Tahir, M.Bal, "Cost Effective and Generalized Turkish Passage Retrieval with ColBERT", International Conference of Innovative Computer Engineering (ICE'25), 2025.