
GAZİ ÜNİVERSİTESİ
Gazi
1926

ÖNERİ SİSTEMLERİNDE DERİN ÖĞRENME TABANLI OTOMATİK SORGU TAMAMLAMA

Abdur Rehman Anwar QURESHI

YÜKSEK LİSANS TEZİ
BİLGİSAYAR BİLİMLERİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

HAZİRAN 2021

Abdur Rehman Anwar QURESHI tarafından hazırlanan "Öneri Sistemlerinde Derin Öğrenme Tabanlı Otomatik Sorgu Tamamlama" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Bilimleri Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. M. Ali AKCAYOL
Bilgisayar Bilimleri, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum…….

Başkan: Prof. Dr. O. Ayhan ERDEM
Bilgisayar Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum…….

Üye: Doç. Dr. Abdullah Talha KABAKUŞ
Bilgisayar Mühendisliği, Düzce Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum…….

Tez Savunma Tarihi: 22/06/2021

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

...…….
Prof. Dr. Aslıhan TÜFEKCİ
Bilişim Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

İmza

Abdur Rehman Anwar QURESHI

22/06/2021

ÖNERİ SİSTEMLERİNDE DERİN ÖĞRENME TABANLI OTOMATİK SORGU TAMAMLAMA

(Yüksek Lisans Tezi)

Abdur Rehman Anwar QURESHI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

Haziran 2021

ÖZET

Bu çalışmada, girdi önekini kullanarak bir sorgu tamamlama listesi oluşturmak için Uzun Kısa Süreli Bellek (Long Short-Term Memory-LSTM) tabanlı Sorgu Otomatik Tamamlama (Query Auto Completion - QAC) önerilmiştir. QAC sisteminin performansı, ilgililik skoru kullanılarak değerlendirilmiş ve geliştirilen QAC modelinin kalitesi, kısmi ve tam eşleştirme stratejileri, başarı oranı, ortalamaların ortalaması hassasiyet (mean average precision), ve normalleştirilmiş azaltılmış toplam kazanç (normalized discounted cumulative gain) kullanılarak değerlendirilmiştir. Geliştirilen LSTM tabanlı QAC sistemi, American Online (AOL) ve Open Resource for Click Analysis in Search (ORCAS) veri kümeleri kullanılarak kapsamlı bir şekilde test edilmiştir. Deneysel sonuçlara göre, önerilen QAC sisteminin performansı kısmi eşleştirme stratejisi kullanıldığında daha başarılı olmuştur. Ayrıca QAC sistemi tarafından önerilen listenin kalitesi tam eşleştirme stratejisinde daha başarılı olmuştur.

Bilim Kodu	:	92432
Anahtar Kelimeler	:	Otomatik sorgu tamamlama, uzun kısa süreli bellek, tek parçalı kodlama, metinsel bilgi erişimi
Sayfa Adedi	:	49
Danışman	:	Prof. Dr. M. Ali AKCAYOL

DEEP LEARNING BASED QUERY AUTO-COMPLETION FOR RECOMMENDATION SYSTEM

(M.Sc. Thesis)

Abdur Rehman Anwar QURESHI

GAZİ UNIVERSITY

INFORMATICS INSTITUTE

June 2021

ABSTRACT

In this study, Long Short-Term Memory based Query Auto-Completion (QAC) has been proposed to generate a query completion list using input prefix. The proposed LSTM based QAC system has been extensively tested using AOL and ORCAS datasets. The performance of the QAC system has been evaluated by using the relevancy score, and the quality of the QAC generation system has been evaluated by using partial and complete matching strategies, success rate, mean average precision and normalized discounted cumulative gain. According to the experimental results, the performance of the proposed QAC system more successful with the partial matching strategy. Also, the quality of the QAC generation list by the proposed QAC system is better on the complete matching strategy.

Science Code	:	92432
Key Words	:	Query auto-completion, long short-term memory, one-hot encoding, textual information retrieval
Page Number	:	49
Supervisor	:	Prof. Dr. M. Ali AKCAYOL

TEŞEKKÜR

Her şeyden önce tez raporumu zamanında tamamlama yeteneği ve cesaretini bana veren Yüce Allah'a minnettarım. Mutluluk ve sıkıntı durumunda her zaman Allah'tan yardım istedim ve Allah ihtiyaç anında bana her zaman yardım etti. Onun üzerimdeki lütfu ile yüksek lisans tez çalışmamı tamamladım. Tüm bu süreçte bana rehber olmaya devam eden danışmanım sayın Prof. Dr. M. Ali AKCAYOL'a özellikle teşekkür ederim. Sayın AKCAYOL bana bu çalışmamın içeriği ve tez hedeflerim konusunda sürekli tavsiyelerde bulundu. Bu tez incelemesinin her adımında başarımda kendisinin oldukça fazla emeği vardır. Tüm çabalarımda bana destek olan ve bende başarılı bir insan potansiyeli gören sevgili aileme teşekkür ediyorum. Son olarak, sessiz desteği beni bu görevi tamamlamama ve üniversitedeki yüksek lisans eğitiminin tadını çıkarmamı sağlayan tüm arkadaşlarıma teşekkür etmek istiyorum.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGE LİSTESİ ix

ŞEKİL LİSTESİ x

KISALTMALAR xi

1.GİRİŞ 1

2. LİTERATÜRE GENEL BAKIŞ 5

3. OTOMATİK SORGU TAMAMLAMA 13

3.1. Problem Tanımı 13

3.2. Sorgu ve Otomatik Tamamlama Kelimeleri Arasındaki İlişki 14

3.3. Etiketleme Stratejisi veya Öneklerin Çıkarılması 15

3.4. QAC Listesinin Oluşturulması 16

3.5. Sıralama Algoritması 17

3.6. Sorgu Önerileri 18

4. GELİŞTİRİLEN OTOMATİK SORGU TAMAMLAMA YÖNTEMİ 21

4.1. Sinirsel QAC 21

4.2. Uzun Kısa Süreli Bellek Sinir Ağları 21

4.3. Tek Parçalı Kodlama (One Hot Encoding) 22

4.4. Genel Mimari 22

5. DENEYSEL SONUÇLAR 25

5.1. Veri Kümeleri 25

Sayfa

5.2. Değerlendirme Ölçütleri 27

5.3. Geliştirilen Uygulama 30

6. DENEYSEL SONUÇLAR VE TARTIŞMA 31

7. SONUÇ 41

KAYNAKLAR 43

ÖZGEÇMİŞ 49

ÇİZELGE LİSTESİ

Çizelge **Sayfa**

Çizelge 1.1. İlk 5 sorgu tamamlama listesi……...…….....…… 4
Çizelge 3.1. QAC problem tanımının en basit örnekleri……...……. 15
Çizelge 3.2. Etiketleme stratejisi…….…….. 16
Çizelge 5.1. AOL veri kümesinin parçalanması….……... 26
Çizelge 5.2. ORCAS veri kümesinin parçalanması.……. 27
Çizelge 5.3. LSTM hiper parametreleri…….……... 30
Çizelge 6.1. QAC modellerinin genel performansı…….……. 32
Çizelge 6.2. Önerilen modelin PMRR metriği kullanılarak değerlendirilmesi.…...…. 35
Çizelge 6.3. Önerilen modelin MRR metriği kullanılarak değerlendirilmesi..…….. 37
Çizelge 6.4. Popüler kelime öneklerinde en iyi 5 aday listesini oluşturduk.……..…… 40

ŞEKİL LİSTESİ

Şekil **Sayfa**

Şekil 1.1. Google trends'e göre Pakistan'da son beş yılda football ve flight için yapılan web araması……..... 2
Şekil 3.1. QAC akış şeması……..... 14
Şekil 3.2. QAC sıralama akış şeması……..... 17
Şekil 3.3. Sorgu önerisi ile QAC arasındaki fark..... 19
Şekil 4.1. Önerilen QAC modeli……..... 23
Şekil 5.1. Veri kümesinin kullanımı……..,……..... 25
Şekil 6.1. Kısmi eşleştirme stratejisi PMRR kullanılarak test sorgu kümesinin beş farklı örneği üzerinde her iki modelin performans analizi…..……..... 33
Şekil 6.2. Tam eşleştirme stratejisi MRR kullanılarak test sorgu kümesinin beş farklı örneği üzerinde her iki modelin performans analiz…..... 34
Şekil 6.3. Bu çalışmada kısmi eşleştirme stratejisi PMRR kullanılmıştır..... 36
Şekil 6.4. Bu çalışmada tam eşleştirme stratejisi MRR kullanılmıştır……..... 38

KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar açıklamalarıyla birlikte aşağıdaki gibi sıralanmıştır:

Kısaltmalar	Açıklamalar
AOL	Amerikalı çevrimiçiler (American online)
CNN	Evrişimli sinir ağları (Convolutional neural network)
CLSM	Evrişimli gizli anlamsal modeli (Convolutional latent semantic model)
GRU	Geçitli tekrarlayan ünite (Gated recurrent unit)
LSTM	Uzun kısa süreli bellek sinir ağları (Long short-term memory)
MAP	Ortalamaların ortalaması hassasiyet (Mean average precision)
MRR	Ortalama karşılıklı sıra (Mean reciprocal rank)
NDCG	Normalleştirilmiş azaltılmış toplam kazanç (Normalized discounted cumulative gain)
NLP	Doğal dil işleme (Natural language processing)
ORCAS	Aramada tıklama analizi için açık kaynak (Open resource for click analysis in search)
PMRR	Kısmi ortalama karşılıklı sıra (Partial-matching mean reciprocal rank)
QAC	Sorgu otomatik tamamlama (Query auto-completion)
RNN	Tekrarlayan sinir ağları (Recurrent neural network)
SR@K	İlk k elemanda başarı oranı (Success rate at top k candidates list)

1. GİRİŞ

Bu bölümde, Sorgu Otomatik Tamamlama (Query Auto-Completion - QAC) konusunda tez yazma amaçlarımızdan bahsettik. Ardından çalışmamızda keşfettiğimiz araştırma sorularını ele aldık. Daha sonra QAC ile ilgili literatür çalışması yaptık. Son olarak QAC sisteminin ana hatlarını gerçek hayattan aldığımız örneklerle örneklendirdik. Burada her alt bölümü de detaylandırdık.

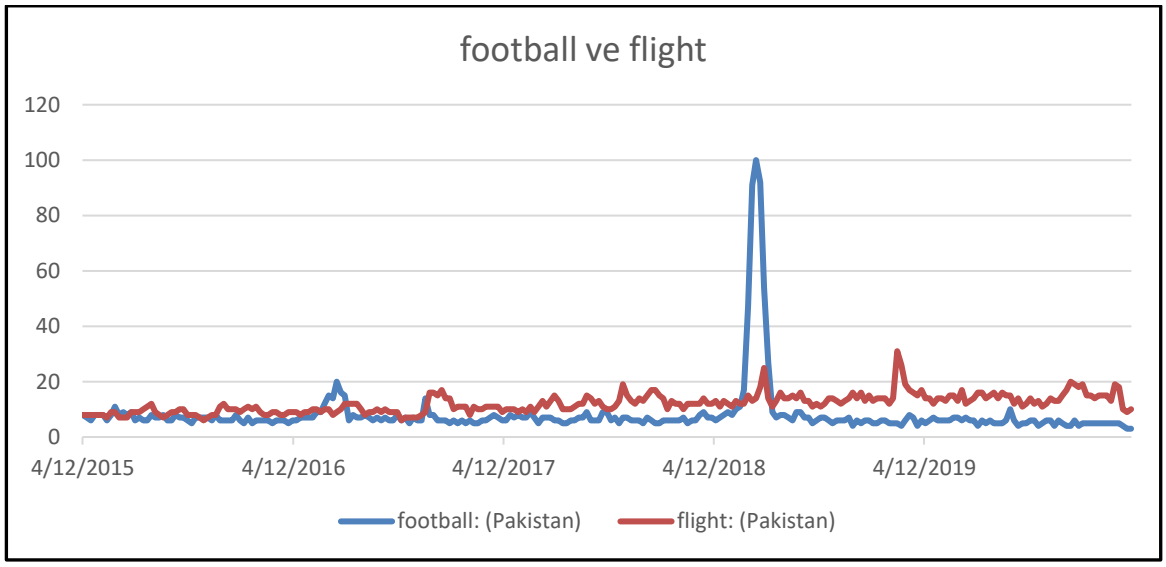
Modern arama motorları tecrübesiz kullanıcılar için daha duyarlı ve etkileşimlidir. Kullanıcının ilk 20 saniye içinde istediğini alamaması durumunda kullanıcının o uygulamadan çıkıp bir başkasına geçme olasılığının daha yüksek olduğu incelenmiştir [1]. Bu bir tür insan-bilgisayar etkileşim problemidir (kullanıcı kullanılabilirliği sorunu olarak da biliniyor) ve bu problem arama motorlarına QAC özelliği eklenerek çözülmüştür. Yazım hatası, harcanan zaman, küçük tuş takımı gibi başka çeşitli sorunlar da gerçek dünya uygulamalarında QAC özelliğinin eklenmesiyle çözülmüştür. QAC ayrıca bağlama, eğilime, dönemlere, zamana, kişiselleştirme ve diğer kısıtlamalara göre bir QAC listesi oluşturmaya yardımcı olmaktadır.

Kelime tamamlama bir veya birden fazla harf girildiğinde kullanıcılara bir kelime listesi sunar. Kelime tamamlama bilgi alma alanında QAC olarak bilinir [2]. QAC, kullanıcılara yazarken amaçlanan sorgularının ortaya konulmasında yardımcı olur. QAC, bilgi işlem cihazlarında yazım hatalarını önlemede önemli bir rol oynar. İngilizce sorguları gönderirken önerilen sorguları seçmenin 2014'te Yahoo arama motorunda %50 daha az tuş vuruşuna neden olduğu görülmüştür [3]. QAC, arama süresini azaltır ve arama motoruna nominal bir yük verir ve bu da makine kaynakları üzerindeki yükü azaltır. Bir bakıma önleyici bakım sağlar.

QAC, bir kullanıcı sorgusunu arama alanına girmeye başladığında bir aday tamamlama listesi oluşturur. Kullanıcılardan gelen ilk giriş sorgu öneki olarak bilinir, aday tamamlama listesi ise QAC olarak bilinir. QAC sistemleri çeşitli kaynaklardan bilgi alır: demografik, sosyal, coğrafi vb. Bu tür bilgiler bir aday tamamlama listesi oluşturmak için kullanılır. QAC, modern arama motorlarının, e-ticaret web sitelerinin, diğer mobil ve masaüstü uygulamalarının her yerde bulunan bir özelliğidir. QAC özelliğinin amacı, kullanıcıya birkaç

giriş karakterine dayalı olarak amaçlanan arama sorgusunun formülasyonunda yardımcı olmaktır.

Ticari arama motorları için QAC özelliğini derinlemesine incelememiz konusunda bizi motive eden örneği düşünün. Yalnızca *f* giriş önekine göre f*ootball* ve *flight* (futbol, uçuş) aramasını örneklersek flight kelimesinin football kelimesinden daha çok tekrarlanan bir sorgu olduğu fark edilecektir. Ancak Dünya Kupası futbol günlerinde kullanıcılar flight kelimesinden çok football kelimesini arayabilir. Sıralamanın nasıl yükseltilebileceği zor görünüyor. Zamansal sorgulara başka bir örnek de kullanıcıların sabahları haberler, güncel olaylar ve finansla ilgili daha fazla ararken gece saatlerinde film, dizi ve diğer eğlenceli aktivitelerle ilgili daha fazla arayabilir. Sonuç olarak, kullanıcı sorgularının zaman içinde oldukça farklı olduğu belirtilebilir. Google Trendler'den[1] football ve flight arama sorgularının karşılaştırmasının bir örneği Şekil 1'de gösterilmiştir:



Şekil 1.1. Google trends'e göre Pakistan'da son beş yılda football ve flight için yapılan web araması.

Bu tez çalışmasında önerdiğimiz model, Bölüm 3.3'te verilen etiketleme stratejisinden çıkardığımız değişken uzunluklu önek için bir tamamlama listesi oluşturmaktadır. Önerilen

[1] Google trendleri web sitesi, google.com'daki en iyi arama sorgularının popülerliğini analiz eder. https://trends.google.com/trends/?geo=PK

yaklaşımın ayrıntıları Bölüm 3.4'te verilmiştir. Bu çalışmanın temel araştırma soruları aşağıdaki gibidir:

1. Önerilen model, değişken uzunluklu önekler için listenin adaylarını oluşturabilir mi?
2. Önerilen modelin performansı ve kalitesi, sırasıyla 10K, 20K, 40K, 50K ve 100K olmak üzere beş farklı test sorgu kümesi örneğinde yeterli düzeyde başarılı mıdır?

QAC güncel bir araştırma alanı olduğundan yapılan çalışmalarda çok sayıda farklı teknikler yöntemler önerilmiştir. Bu çalışmalarda bir listenin adaylarını girdi sorgu öneki kullanarak tahmin eden QAC modelinin eğitimi için kural tabanlı, istatistiksel ve öğrenme teknikleri kullanılmıştır. Son yıllarda derin öğrenme algoritmalarının ve doğal dil işleme tekniklerinin kullanımı daha fazla görülmeye başlanmıştır. Bu çalışmada, açık iki veri kümesini kullanarak kullanıcının amaçladığı sorgunun oluşturulması için derin öğrenmeye dayalı bir yaklaşım önerilmiş ve uygulama geliştirilmiştir. Her iki veri kümesinin ayrıntıları sırasıyla Bölüm 5.1.1 ve Bölüm 5.1.2'de sunulmuştur. Önerilen yaklaşım için ayrıntılı örnekler Çizelge 1.1'de gösterilmektedir.

Çizelge 1.1'de ilk 5 tamamlama listesi önerilen yaklaşım kullanılarak elde edilmiştir. Bölüm 5.1.1'deki veri kümesi, QAC listesinin bu versiyonunda kullanılmıştır. Eşik değerinden memnun olmayan adaylar tamamlama listesinden çıkarılmıştır. Tamamlama adayları "*computer science for sale in the continental airline*" ve "*computer science classified*" anlamsal (semantically) olarak doğru değildir. Bu, önerilen çalışmamızın sınırlamasıdır.

Çizelge 1.1. İlk 5 sorgu tamamlama listesi

Önek	**Aday Listesi**
computer science	*computer science projects* *computer science career* *computer science classified* *computer science for sale in the continental airline* *computer science project*
programming	*programming club.org* *programming community college australia* *programming and southern law* *programming.com* *programming charlotte*
america	*america consulate* *america.com* *americal school* *american* *american map*

2. LİTERATÜRE GENEL BAKIŞ

QAC günümüzde oldukça popüler bir araştırma konusudur ve bilimsel çalışmalarda çok çeşitli teknikler ve yöntemler önerilmiştir. Bu çalışmada QAC sistemleriyle ilgili kapsamlı literatür araştırması yapılmış ve geliştirilen yöntemler karşılaştırmalı olarak analiz edilmiştir.

Bar-Yossef ve Kraus [4], kullanıcı günlüklerinin bir QAC sisteminde aday listesinin oluşturulması için etkili olabileceğini göstermişlerdir. Kullanıcının son günlüklerini kullanarak amaçladığı sorgunun tahmin edilmesi için En Popüler Tamamlama (Most Popular Completion - MPC) QAC yaklaşımı olarak bilinen yeni bir yaklaşım önermişlerdir. QAC çalışmalarında yaygın kullanılan bir yaklaşım haline gelen MPC, [5,6,7,8,9,10] gibi diğer önemli QAC çalışmalarında yaygın olarak kullanılmaktadır. Kosinüs benzerlik ölçütü, giriş önekiyle kullanıcı sorgu günlükleri arasındaki benzerliği belirlemek için kullanılmaktadır. İlgili çalışmada AOL veri kümesi [11,12] kullanılmıştır.

Shokouhi ve Radinsky [7], zaman duyarlı bir yöntem geliştirerek tamamlama adayları listesinin daha iyi sıralanması için yeni bir QAC yaklaşımı önermişlerdir. Önerilen yaklaşım daha kısa ve güncel sorguları kullanmanın yanı sıra zaman niteliğine de sahiptir. Yazarlar, MPC'yi [4] temel olarak alıp sorguların zaman niteliklerini kullanarak QAC performansını artırdılar.

Shokouhi [8], kişiselleştirilmiş QAC alanında çalışmıştır ve çevrimdışı etiketler oluşturmak için yeni etiketleme stratejisini geliştirmiştir. Bu etiketleme stratejisi çok popüler hale gelmiştir ve QAC ile ilgili çalışmalarda yaygın olarak kullanılmıştır [6,9,10]. Yazarlar, çeşitli kullanıcı odaklı ve demografik özellikleri kullanmışlardır. Bunun üzerine kullanıcı arama günlüklerinin ve konumunun kişiselleştirilmiş QAC sistemleri için önemli özellikler olduğu ortaya çıkmıştır. Bu çalışmada AOL veri kümesi [11,12] ve Microsoft Bing veri kümesi kullanılmıştır.

Di Santo vd. [13] tek veri kümesinde QAC sistemleri için mevcut sıralama yöntemlerini karşılaştırmışlardır. Bunun için 11 farklı sıralama yöntemi uygulanmıştır ve ardından her bir derecelendirme yönteminin aynı veri kümesi üzerindeki etkinliğini belirlemek için analiz yapılmıştır. QAC sisteminin performansının kullanıcı tarafından yazılan giriş karakterlerinin sayısına bağlı olduğu tespit edilmiştir. Ancak kullanıcı kişisel bilgilerinin, sorgu

günlüklerinde mevcutsa QAC sisteminin performansının kullanıcı giriş öneki dikkate alınmadan önemli ölçüde artmakta olduğu görülmüştür.

Jiang vd. [14], periyodik ve ani artma eğilimine dayanan zaman duyarlı hibrit QAC sıralama modelini önermişlerdir. Periyodiklik, yenilik sorgularının popülerliğini tanımlamakta ve ani değişimler geçmişte hiç görülmemiş son dakika haberleri veya görünmeyen sorgular gibi yeni olayları göstermektedir. Önce sorgunun dönemselliği tespit edilmiş ve ardından Ayrık Fourier Dönüşümü (Discrete Fourier Transform - DFT) kullanılarak sonuçlar elde edilmiştir. Bu çalışma, aday listesinin sıralama açısından kalitesini önemli ölçüde artırmıştır.

Kannadasan ve Aslanyan [9], QAC'nin kişiselleştirme özelliği üzerinde çalışarak kullanıcı bağlamını kullanarak kişiselleştirilmiş QAC modelini önermişlerdir. Önerilen model iki amaçla kullanılmıştır: ilk nesil aday listesi ve ardından kişiselleştirmeye göre listedeki her adayın sıralaması. Önerilen yaklaşım ticari arama motoru eBay için kullanılmıştır ve uygulamada eBay a ait bir veri seti kullanılmıştır. Kullanıcı bağlamı, kişiselleştirilmiş QAC sistemlerinin uygulanmasında hayati bir rol oynamaktadır. Kişiselleştirilmiş QAC sisteminde, kırsal bölgeden gelen bir kullanıcı, "*apple*" sorgusunu aradığında arama motoru çeşitli "*apple*" meyvelerini ve tarım yöntemlerini gösterir, İstanbul gibi gelişmiş büyükşehirden bir kullanıcı "*apple*" sorgusunu aradığında arama motoru "*iPhone*", "*apple watch*", "*Apple TV*", "*iPad*", "*Macintosh*" gibi "*Apple*" teknoloji şirketinin ürünlerini gösterir.

Addepalli vd. [15] harita tabanlı QAC sıralama problemi üzerinde çalışmıştır. Bu çalışmada ikili (pairwise) ve liste tabanlı (listwise) sıralama yaklaşımlarını değerlendirmişlerdir. QAC tabanlı haritalar için uygun aday listesi sorgu önerileri, iş adresleri ve belirli konumu içermektedir.

Park ve Chiba [6], QAC doğrultusunda sorgu tamamlama zorluğunu analiz etmek için çalışmıştır. Kullanıcı giriş önekinin tamamlanması için sinirsel dil modelini (neural language model) önermişlerdir. Önerilen bu model, sorgu günlüklerini dikkate almadan kullanıcının amaçladığı sorguyu üretmekteydi. Önerilen algoritma, geçmişte hiç görülmemiş sorgular için yaygın kullanılan bir yaklaşım haline gelmiştir. Önerilen yaklaşım QAC sistemi için geleneksel temel yaklaşımla [4] karşılaştırıldığında daha önce görülen sorgular için benzer sonuçlar vermektedir.

Fiorini ve Lu [10], Park ve Chiba [6] tarafından yapılan çalışmayı temel alarak QAC sisteminin kişiselleştirme özelliği üzerinde çalışmalar yapmışlardır. QAC için sinirsel dil modeline kullanıcının kişisel bilgilerini ve zaman özelliğini eklemişlerdir.

Kim [5], sinirsel dil modelinin karakter bazında sorgu tamamlamaya dayandığını ve bu yaklaşımın QAC sisteminin genel performansını yavaşlattığını ifade etmiştir. Bu sınırlamanın üstesinden gelmek için QAC için alt kelime dil modeli getirilmiştir. Önerilen yaklaşım görülen sorgular [4] ve görünmeyen sorgular [6] için karşılaştırılan yaklaşımlardan 2,5 kat daha hızlı sonuç üretmiştir.

Rossiello, G. ve diğerleri [16] sorgu günlüklerinin yokluğunda bir aday tamamlama listesi oluşturmak için topluluk tabanlı yeni QAC yaklaşımını önermişlerdir. Bu yaklaşım özellikle küçük ölçekli arama motorlarına yöneliktir. Önerilen teknik, sorgu tamamlama kümesini tahmin etmek için sistemin içerik bilgisinden isim öbeklerini (noun phrases) çıkarmaktadır. Deneysel sonuçlar, temel yaklaşımlara kıyasla soğuk başlangıç (cold-start) kullanıcısının performansında ve kalitesinde önemli bir gelişme olduğunu göstermektedir.

Chen, W. ve diğerleri [17], sorgu günlüğü ve kullanıcı arasındaki ilişkinin önemini göstermişlerdir. Dikkat mekanizması (attention mechanism) ile sorgu önerisi için sinirsel dil modelini önermişlerdir. Bir sorgu öneri listesi oluşturmak için kullanıcı geçmişini kullanmışlardır. Bu makale sorgu önerisiyle ilgili olsa da sorgular ve kullanıcı arasındaki güçlü ilişkiyi anlamak için çok önemli bir çalışmadır. Sorgu önerisi ve QAC arasındaki farklılıklar Bölüm 2.6'da ayrıntılı olarak anlatılmıştır.

Jaech, A. ve Ostendorf, M. [18], Park ve Chiba [6] tarafından görünmeyen sorgular için önerilen sinirsel dil modeline kullanıcının kişiselleştirilmiş özellikleri eklenmiştir. Jaech, A. ve Ostendorf, M. [18] önerilen kişiselleştirilmiş sorgunun ilk çalıştırma problemi durumunda bile kullanıcının çevrim içi faaliyetlerinden yararlanarak bir aday tamamlama listesi oluşturduğunu belirtmiştir. Bu kişiselleştirilmiş sinirsel dil modeli, hedeflenen kullanıcının bağlamsal bilgilerini kullanmaktadır.

Whiting, S. ve Jose, J. M. [19] QAC doğrultusunda son sorgular özelliğini araştırmıştır. Binlerce müşterinin potansiyel ihtiyaçlarını ve gereksinimlerini karşılamak için İnternette oluşturdukları sorgularla arama yaptığını ve bu sorguların %10'undan fazlasının daha önce görülmediğini ifade etmişlerdir. Görünmeyen sorgular, haber uyarısı, son dakika haberi, güncel olay gibi kendine özgü nitelikleri nedeniyle sorgu tamamlama listesi oluşturmak için

tahmin edilememektedirler. Whiting, S. ve Jose, J. M. [19] QAC sisteminin bu sorununun üstesinden gelmek ve son sorgular için aday tamamlama listesi oluşturmak için sezgisel bir yaklaşım önermişlerdir.

Mitra, B. ve Craswell, N. [20] QAC için sık rastlanmayan önekler problemi üzerinde çalışmışlardır. En popüler tamamlama yaklaşımında bile [4] sorgu öneki sorgu geçmişinde yoksa yeni bir sorgu tamamlama aday listesi oluşturulamadığını belirtmişlerdir. Sorgu geçmişinde sunulan son eklerin çoğunu kullanarak az rastlanan önek sorununun üstesinden gelen QAC modelini önermişlerdir. Daha sonra aday liste sıralama problemi için web aramasında [21] bilgi erişim görevleri tarafından orijinal olarak sunulan Evrişimli Gizli Anlamsal Modeli (Convolutional Latent Semantic Model - CLSM) kullanmışlardır. CLSM modeli aday tamamlama listesini yeniden sıralamak için n-gram, önek ve sonek uzunluğu, kosinüs benzerliği gibi çeşitli özellikleri kullanmaktadır.

Wang, S. ve diğerleri [22] tamamlama listesinin anlamsal içeriğini elde etmek için bağlam modelleme tekniğini ekleyerek sinirsel dil modeli tabanlı QAC sistemini önermiştir. Önerilen yaklaşım aday listesinin oluşturulması ve sıralanmasını içermektedir. Aday listesinin oluşturulması için bağlamsal bilgiden yararlanılmış ve aday listesinin sıralanması için normalleştirilmemiş sinirsel dil modeli kullanılmıştır. Önerilen derin öğrenmeye dayalı yaklaşım, QAC için son teknoloji yaklaşımlardan görülen ve görünmeyen her iki sorgu türü için de işe yaramaktadır.

Gog, S. ve diğerleri [23] QAC'nin son teknoloji yaklaşımlarının verimliliğini ve etkinliğini test etmek amacıyla eBay için uyguladı. QAC sistemi için önek arama (prefix-search) ve birleşik arama (conjunctive-search) arasındaki farkı açıklamışlardır. Önek arama, sorguyu yalnızca ileri yönde tamamlarken, bağlantılı arama sorguyu ileri, geri ve orta yönlerde tamamlamaktadır. Önek araması hızlıdır, ancak önek sorgu geçmişinde eşleşemezse tamamlama listesini oluşturamaz. Birleşik arama daha çeşitlidir ve sorgunun öneki sorgu günlüklerinde bulunmasa bile bir ilk tamamlama listesi oluşturur. Birleşik arama sonuçları, önek aramaya göre QAC sisteminin etkinliğini güçlendirir.

Li, Y. ve diğerleri [24] aday tamamlama listesi oluşturulmasına kullanıcı etkileşimini ekleyerek QAC sisteminin performansını araştırmışlardır. Kullanıcı davranış özelliklerini oturum tabanlı sorgu geçmişlerinden çıkararak sorgu aday kümesinin tahmini için kullanıcı davranış özelliğini kullanan bir QAC modeli önermişlerdir. Deneysel sonuçların, kullanıcı davranış özelliklerinin QAC sisteminin performansını güçlendirdiğini göstermişlerdir.

Cai, F. ve diğerleri [25] QAC sisteminin farklı kullanıcılar için çeşitlendirilmesi (diversification) üzerinde çalışmışlardır. Sorgu tamamlama aday listesinden tekrarlamayı en aza indirdiler ve belirli kullanıcı için ilk aşamada daha iyi bir tamamlama listesi oluşturdular. Kullanıcının son arama geçmişi, QAC sistemine çeşitlendirme özelliklerinin eklenmesinde önemli bir rol oynamıştır. Deneysel sonuçları En Popüler Tamamlama yaklaşımı [4] ile karşılaştırdılar ve önerilen yaklaşımın performans, kalite ve kullanıcı deneyimi açısından daha başarılı olduğunu göstermişlerdir.

Shao, T. ve diğerleri [26] QAC yönündeki anlamsal benzerliğin önemini vurgulamışlardır. Sorgu anlamsal benzerliğine ve tekrar tekrar sorgulamaya dayalı QAC modelini önermişlerdir. Tamamlama aday listesi ve kullanıcının son arama geçmişi arasındaki anlamsal benzerliği bulmak için kelimeden vektöre yöntemi kullanılmıştır. Deneysel sonuçlar, önerilen yaklaşımın QAC sisteminin performansını ve ardından son teknoloji En Popüler Tamamlama yaklaşımını [4] geliştirdiğini göstermiştir.

Jiang, J. Y. vd. [27] kullanıcının bağlamsal bilgilerinden çıkarılan kullanıcı davranış özelliklerini ekleyerek QAC sisteminin performansını araştırmışlardır. Önerilen denetimli öğrenme tekniği terim, oturum ve sorgu seviyesi özelliklerini kullanmaktadır. Deneysel sonuçlar, kullanıcı davranış özelliklerinin 4 farklı QAC temel yaklaşımıyla karşılaştırıldığında QAC sisteminin performansını güçlendirdiğini göstermektedir. İyileştirilmiş sonuçlar, farklı veri kümelerinde de aynı şekilde başarılı olmuştur.

Vargas, S. vd. [28] terime dayalı sorgu tamamlamanın öneminden bahsetti. Tam bir sorgu yerine tamamlama aday listesini oluşturan yeni QAC modelini önerdiler. Özellikle tuş takımının diğer masaüstü ve web tabanlı uygulamalara göre çok küçük olduğu mobil aramaya odaklandılar. Önerilen terime dayalı QAC yaklaşımı, En Popüler Tamamlama yaklaşımı [4] ile karşılaştırıldığında mobil cihazlar için daha başarılı olmuştur.

Krishnan, U. vd. [29], QAC'nin Amazon, Twitter, Google, Facebook, Wikipedia, LinkedIn vb. gibi çok sayıda arama platformunda yararlı bir özellik olarak ortaya çıktığını belirtmişlerdir. QAC, kullanıcıların arama taleplerini formüle etmelerine yardımcı olmaktadır. QAC, bir sorgu girmek için gereken tuş vuruşlarının sayısını azaltmakta ve kullanıcıları amaçlanan sorguya yönlendirmektedir. Girilen tamamlanmamış sorgu ile uygun eşleşme için büyük veri yöntemleri ile sorgu tamamlama listesi çıkarılmıştır. QAC, sorgu tamamlama süresinde ve sunulan önerilerin kalitesinde başarılı olmuştur.

Arkoudas, K. ve Yahya, M [30] büyük boyutlu Bloomberg[2] haber verisi kullandılar ve semantik tabanlı bir QAC yöntemi önermişlerdir. Tasarlanan ve uygulanan algoritmalar en popüler tamamlama, atomik tamamlama ve şablon tamamlamadır. Bu, kullanıcının işini kolaylaştırmak ve insan-bilgisayar etkileşimini geliştirmek için daha çok Bloomberg haber verisi için özel olarak gerçekleştirilen veriye özgü bir çalışma olmuştur.

Tahery, S. ve Farzi, S [31], arama motoru özellikleri ve metinsel bilgi erişim alanları yönünde QAC'ye odaklanmışlardır. Konum, zaman, demografi ve bağlam özelliklerinin kullanıcıların aldıkları bilgiyi iyileştirdiği ve süreyi kısalttığı görülmüştür. Özelleştirilmiş QAC sistemi için konum, zaman, demografik ve bağlamsal bilgiler açısından kapsamlı inceleme yapılmıştır.

Dehghani, M. vd. [32], kullanıcı sorgusunu yeniden düzenleyerek daha iyi sonuçlar elde edildiği ve arama motorlarının yanı sıra kullanıcının işini de kolaylaştırmak için kullanıcı sorgularının belirsizliğinin ortadan kaldırıldığını belirtmişlerdir. Sorguyu yeniden düzenleme yöntemi belirli ve kesin bir öneri listesi sağlamaktadır. Sorgu yeniden biçimlendirme, en son teknolojinin farkında olmayan veya amaçlanan bir sorgu oluşturmak için doğru harfleri yazamayan kullanıcılara yardımcı olmaktadır. Bununla birlikte sorgu yeniden düzenlemenin kullanıcıların arama sonuçlarından memnuniyetini artırdığı görülmüştür.

Li, L. vd. [33] Hawkes süreçlerine (Hawkes processes) dayalı QAC modelini önermişlerdir. Önerilen yaklaşım, önek girdisi üzerinden formüle edilen aday tamamlama listesi için zamansal, bağlamsal ve tıklama hassasiyetleriyle ilgilenmektedir. Hawkes süreçleri, olay dizisi verileri için sağlam bir istatistik yaklaşımıdır. Çıkarım algoritması, aday tamamlama listesinin optimum çözümünü hesaplamak için de kullanılmaktadır. Önerilen yaklaşım, iki gerçek veri kümesi üzerinde uygulanmış ve deneysel sonuçlar son teknoloji QAC yaklaşımlarına göre verimlilikte önemli bir gelişme göstermiştir.

Hu, Y. vd. [34] aynı oturumdaki son sorgu günlüklerine dayanarak QAC modelini önermişlerdir. Önerilen yaklaşım, aday tamamlama listesini oluşturmak için anlamsal benzerlik tekniğini kullanmıştır. Kullanıcının bağlamsal bilgileri, tamamlama aday listesinin kalitesini iyileştirmek için kullanılmıştır. Deneysel bulgular, anlamsal benzerlik yönteminin

[2] Bloomberg, özel bir yazılım, finans, veri ve medya şirketidir. https://www.bloomberg.com/middleeast

son teknoloji yaklaşımlarla karşılaştırıldığında QAC sisteminin performansını artırdığını göstermektedir.

Wang, Y. vd. [35], çevrim içi trendlere göre bir tamamlama aday listesi oluşturan trend duyarlı bir QAC modelini önermişlerdir. Önerilen yaklaşımda Bayesian Çıkarım (Bayesian Inference) ve Thompson Örnekleme (Thompson Sampling) tekniklerini kullanmışlardır. Bu yaklaşım, bir kullanıcı tarafından girilen girdinin öneki üzerinden tamamlama aday listesinin oluşturulması ve sıralanmasındaki zamansal eğilimlerini de ele almaktadır. Karar, en son trendlerin öğrenilmesi ve buna göre sonuçların formüle edilmesi için çevrim içi olarak alınmıştır.

Hu, S. vd. [36], konuma duyarlı QAC sorununu araştırmışlardır ve bir aday tamamlama listesi oluşturan QAC modeli önermişlerdir. Mobil cihazlarda konuma duyarlı QAC sistemlerinin kullanımının son birkaç yılda çok büyük bir artış gösterdiğini ve bu tür yaklaşımların sistemin genel kalitesini ve performansını güçlendirdiğini ifade etmişlerdir. Konuma duyarlı QAC, bir aday tamamlama listesi oluşturmak için belirlenmiş noktalardan oluşan bir konum özelliğine sahip sorgudan oluşmaktadır. Bu tür sorgularda dize öneki, uzamsal konum ve metin içeriği vardır. Konuma duyarlı QAC sistemini anlamak için bir senaryo düşünülebilir: İslamabad'daki kullanıcının amaçlanan bir sorgu oluşturmak için Bahria Üniversitesine girdiğini düşünelim. Bu durumda, beklenen sorgu "Bahria Üniversitesi, İslamabad"dır. Ancak konumun farkında olmayan QAC teknikleri ilgili kullanıcı konumundan çok uzakta bulunan QAC aday tamamlama sonuçları olarak "Bahria Üniversitesi, Lahor" veya "Bahria Üniversitesi, Karaçi" sonucunu verir. Eğer arama motoru bu durumda İslamabad olan kullanıcı konumunu biliyorsa, arama motoru konum olarak İslamabad şehri olanları listede gösterecektir.

Qin, J. ve diğerleri [37], hızlı bir şekilde aday tamamlama listesi oluşturan hata toleranslı QAC sistemini geliştirmek için düzenleme mesafesi (edit distance) yöntemini kullanmışlardır. Jiang, D. vd. [38], kullanıcı oturum geçmişini kullanarak kişiselleştirilmiş bir aday tamamlama listesi oluşturan QAC yaklaşımını önermişlerdir. Mitra, B. ve diğerleri [39] aynı kullanıcının sorgu günlüğünden çıkarılan kullanıcı etkileşim özelliklerini ekleyerek QAC sisteminin etkinliğini araştırmışlardır. Krishnan, U. ve diğerleri [40] ise QAC için sentetik sorgu günlükleri üretme üzerine çalışmışlar ve etkinliği gerçek sorgu günlüğüyle karşılaştırmışlardır.

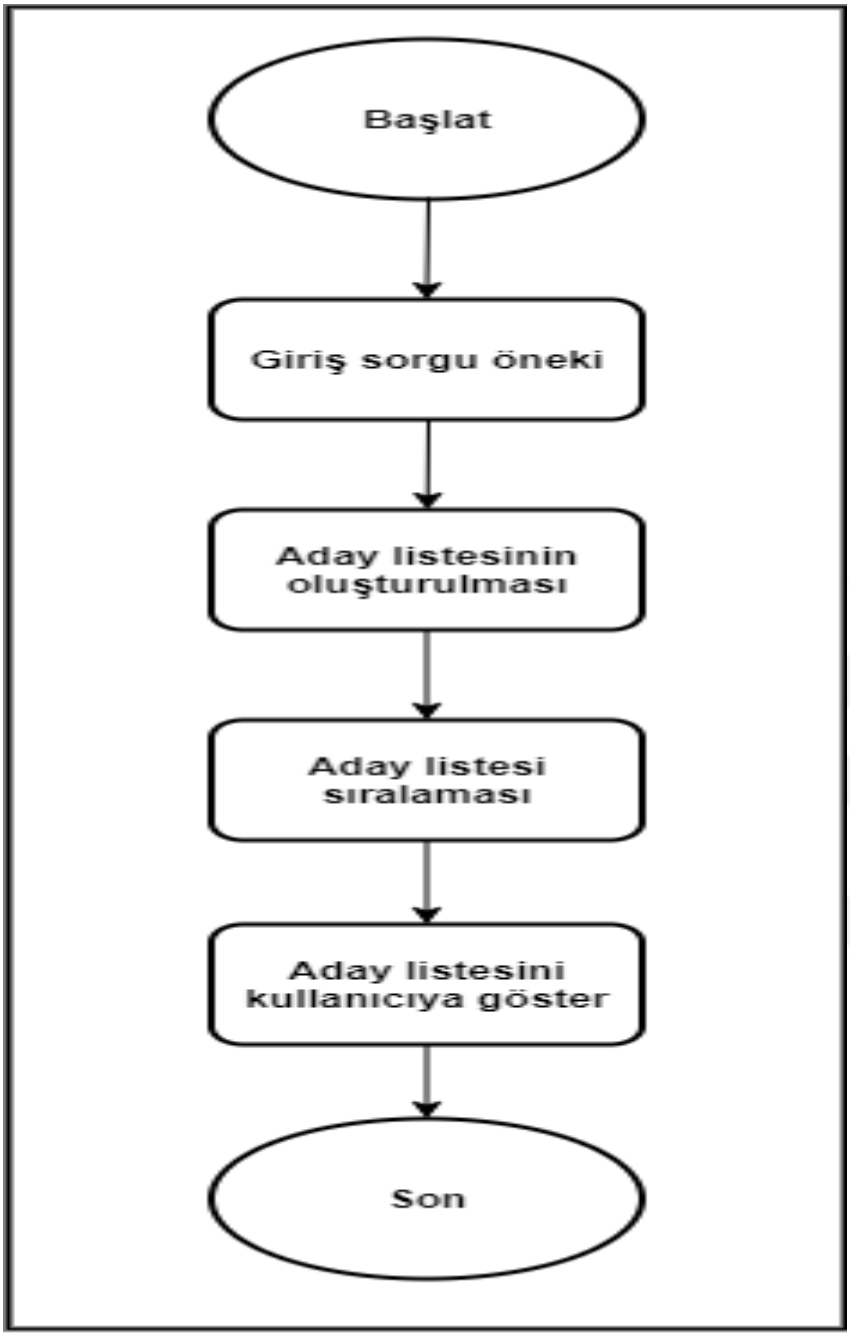
Bu çalışmada, QAC araştırma yöntemine yönelik literatür çalışmalarını detaylı bir şekilde incelenmiş ve ardından kullanıcının amaçladığı tamamlama aday listesinin oluşturulması için derin öğrenmeye dayalı bir QAC modeli geliştirilmiştir. Önerilen yaklaşımın Bölüm 4.4'te detaylı şekilde sunulmuştur.

3. OTOMATİK SORGU TAMAMLAMA

Bu bölümde QAC problemi detaylı bir şekilde tanımlanmıştır ve sorgu ile otomatik tamamlama kelimeleri arasındaki ilişki biçimsel şekilde ifade edilmiştir. Daha sonra QAC için etiketleme stratejisi, QAC listesinin oluşturulması, sıralama algoritması ve sorgu önerisi için yaklaşımlar ele alınmıştır. Önerilen yaklaşımda etiketleme stratejisi, QAC listesi oluşturulması ve sıralama algoritması geliştirilmiştir. Son olarak da sorgu önerisi ve QAC arasındaki temel farklar açıklanmıştır.

3.1. Problem Tanımı

QAC problemi üç ana bileşen ile tanımlanır; birincisi, P olarak belirtilen kullanıcı girişi sorgu önekidir. İkincisi, L = {C_1, C_2, …, C_n} olarak gösterilen aday eleman listesinin oluşturulmasıdır. Burada, C_n, liste L'nin bir adayı ve n doğal sayıdır. İlk önce n tane aday seçilmiştir ve L listesinde kaydedilmiştir. Bu aday listesi, kullanıcı girişi sorgu öneki kullanılarak bir sonuç olarak gelmekte ve her aday P önekinden başlamaktadır. Son ve üçüncü adım ise belirli bir kullanıcı için aday listesinin sıralanmasıdır. Sıkı eşleştirme politikası, kullanıcı geçmişi, oturum geçmişleri, bağlamsal faktör, zamansal faktör, kişiselleştirme gibi çeşitli özelliklere dayanan sıralama algoritmasına sahip QAC için işlem adımları Şekil 3.1'de görülmektedir.



Şekil 3.1. QAC akış şeması

3.2. Sorgu ve Otomatik Tamamlama Kelimeleri Arasındaki İlişki

Sorgu ve otomatik tamamlama kelimeleri arasındaki ilişki ticari arama motorları için çok önemlidir. Bu ilişki Çizelge 3.1'de görülmektedir.

Çizelge 3.1'de QAC problem tanımının en basit örneği şu şekilde tanımlanmaktadır, kullanıcı girişi sorgu öneki: "*P = New*". Oluşturulan aday listesi şu şekilde temsil edilir: "*L={New, Newspaper, New York, New jersey, News24, New look}*".

Çizelge 3.1. QAC problem tanımının en basit örnekleri

Önek	Aday Listesi
New	*News* *Newspaper* *New york* *New jersey* *News24* *New look*
Bahria	*Bahria College* *Bahria University* *Bahria Town* *Bahria Town Islamabad* *Bahria Town Lahore* *Bahria Town Karachi*
Bahria Uni	*Bahria University Islamabad Campus* *Bahria University Lahore Campus* *Bahria University Karachi Campus* *Bahria University Medical and Dental College* *Bahria University Institute of Professional Psychology* *Bahria University National Center for Maritime Policy Research*

3.3. Etiketleme Stratejisi veya Öneklerin Çıkarılması

QAC sistemi, giriş sorgu önekine göre kullanıcının amaçladığı sorguyu oluşturmaktadır. Bu nedenle, giriş sorgusu önekinin uzunluğunu başlangıçta varsayarak almak önemli bir adımdır. Shokouhi ve arkadaşları [8] tarafından önerilen QAC sisteminin etiketleme stratejisi analiz edilmiştir. Bu makalenin önerilen yaklaşımı, girdi sorgusu önekini veya etiketini orijinal sorguya düzenlemektedir. Fakat önerilen yaklaşım sadece orijinal sorgunun ilk terimi için bir giriş öneki sorgusu oluşturacak şekilde değiştirilmiştir. Örneğin, orijinal sorgu "*computer science*" ve değiştirilmiş etiketleme stratejimiz, "*computer science*" orijinal

sorgusuna karşı bir girdi sorgusu öneki olarak "*comp*" üretmektedir. Değiştirilmiş etiketleme stratejisinin ayrıntılı bir örneği Çizelge 3.2'de verilmektedir.

Çizelge 3.2. Etiketleme stratejisi

q_u	**c**	**co**	**com**	**comp**
q_1	casters.com	com	com	company
q_2	cholson	county	compass	company.com
q_3	cardina status	county club	communications	computer com ×
q_4	channel.com	corps	community	companies
q_5	college	copers	community college	company conference
PMRR	0,00	0,00	0,00	0,33

Çizelge 3.2'de q_u, her tuş vuruşunda kullanıcı giriş örneğini belirtmektedir ve q1 - q5 ise giriş önekine karşı beş otomatik tamamlama adayıdır. Kısmi eşlemeli ortalama karşılıklı sıra matrisi (partial-matching mean reciprocal rank - PMRR) makalede [6] tanıtılmaktadır. × sembolü, giriş önekini kullanarak üretilen, adayın orijinal sorgu ile başarılı bir şekilde kısmen eşleşmesini göstermektedir. Olası girdi sorgusu öneki, orijinal sorgunun en azından ilk terimini oluşturduğunda girdi öneki sorgusu seçilmiştir. Bu durumda, orijinal sorgu olarak "*computer science*" için giriş öneki olarak "*comp*"u seçtik. Sonuçlar, Bölüm 5.1.1'de verilen veri kümesine göre bu çalışmada geliştirilen model kullanılarak elde edilmiştir.

Her iki veri kümesi için test bölümünün öneklerini almak amacıyla bu etiketleme stratejisi uygulanmıştır ve daha sonra her iki modelin kapsamlı deneysel testlerinde çıkarılan ön ekler kullanılmıştır. Deney düzeneğinin ayrıntıları Bölüm 5 ve Bölüm 6'te deneysel sonuçlar ve tartışmalar bölümünde ayrıntılı olarak sunulmuştur.

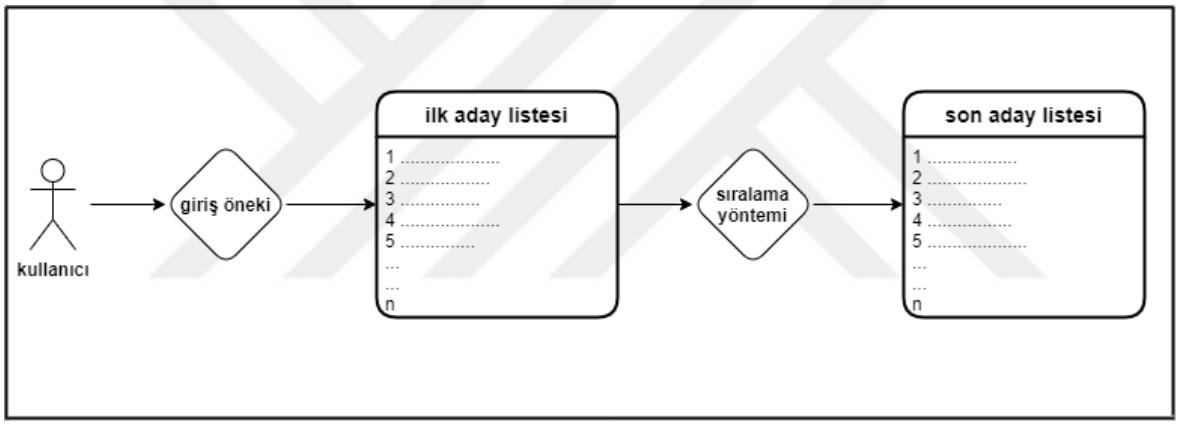
3.4. QAC Listesinin Oluşturulması

QAC sisteminin bir modeli oluşturulduğunda q giriş önekine karşı bir tamamlama aday listesi oluşturulabilir. QAC listesinin oluşturulması, tamamlamaları sağlayarak kullanıcıya yardımcı olmayı amaçlamaktadır. QAC listesi yaklaşımının oluşturulmasının amacı, sorgu tamamlama sonucunda kullanıcının önek sorgusunu aday sorguya dönüştürmek için karakter tamamlamaktır. Bu çalışmada, Bölüm 3.3'te verilen etiketleme stratejisiyle çıkarılan her bir giriş sorgusu önekine karşı 10 sorgu adayı oluşturulmuştur.

Gerçek sorgu ve girdi sorgu önekini kullanarak tamamlama sorgusu sonek uzunluğu hesaplanmıştır. Daha sonra, sonek uzunluğunun sonuna kadar 10 sorgu adayı oluşturulmuştur. Eğer giriş önekinin son karakteri yeni bir satır karakteri (“\n”) oluşturuyorsa o zaman devam edilmeden yeni satıra geçilmektedir. Bundan sonra önerilen yaklaşım satır sonu karakteri haricinde başka bir karakter üretene kadar aynı döngü tekrar edilmektedir.

3.5. Sıralama Algoritması

QAC aday listesi sıralaması QAC sistemlerindeki en önemli işlemlerden birisidir. QAC listesinin çeşitli faktörlere göre sıralaması şöyledir: kullanıcı geçmişi, oturum geçmiş bilgileri, bağlamsal faktör, zamansal faktör, kişiselleştirme vb. QAC sıralama adımının akış şeması Şekil 3.2’de gösterilmektedir.



Şekil 3.2. QAC sıralama akış şeması

Literatürde, QAC için çok sayıda sıralama yaklaşımı önerilmiştir [4,8,13,15,20]. Bu çalışmada Di Santo ve arkadaşlarının [13] kullandığı N-gram benzerlik sıralayıcısı (N-gram similarity ranker) kullanılmıştır, ancak kullanıcı bağlamı yerine oluşturulmuş aday listesi kullanılmıştır. Giriş öneki q, Bölüm 3.3’te verilen etiketleme stratejisiyle çıkarılan bir karakter dizisidir, c ise oluşturulan aday listesinin (C_q) her bir adayıdır. QAC özelliğinin amacı, aday listesini en iyi eşleşen giriş öneki q girişine göre sıralamaktır. Etkili bir sıralama algoritması geliştirmek için gerçek sorgu sonekini yalnızca Q'_p olarak bulmak amacıyla gerçek sorgudan q öneki kaldırılmıştır. Benzer şekilde, C_q listesindeki her adaydan c' aday sonekini bulmak için öneki de kaldırılmıştır. Değiştirilmiş N-gram benzerlik sıralayıcı, her aday sonekinin skorunu gerçek sorgu sonekine göre hesaplamakta ve nihai adayların listesini daha yüksek bir skora göre hesaplamaktadır. Özellikle benzerlik yöntemi olarak Kondrak ve

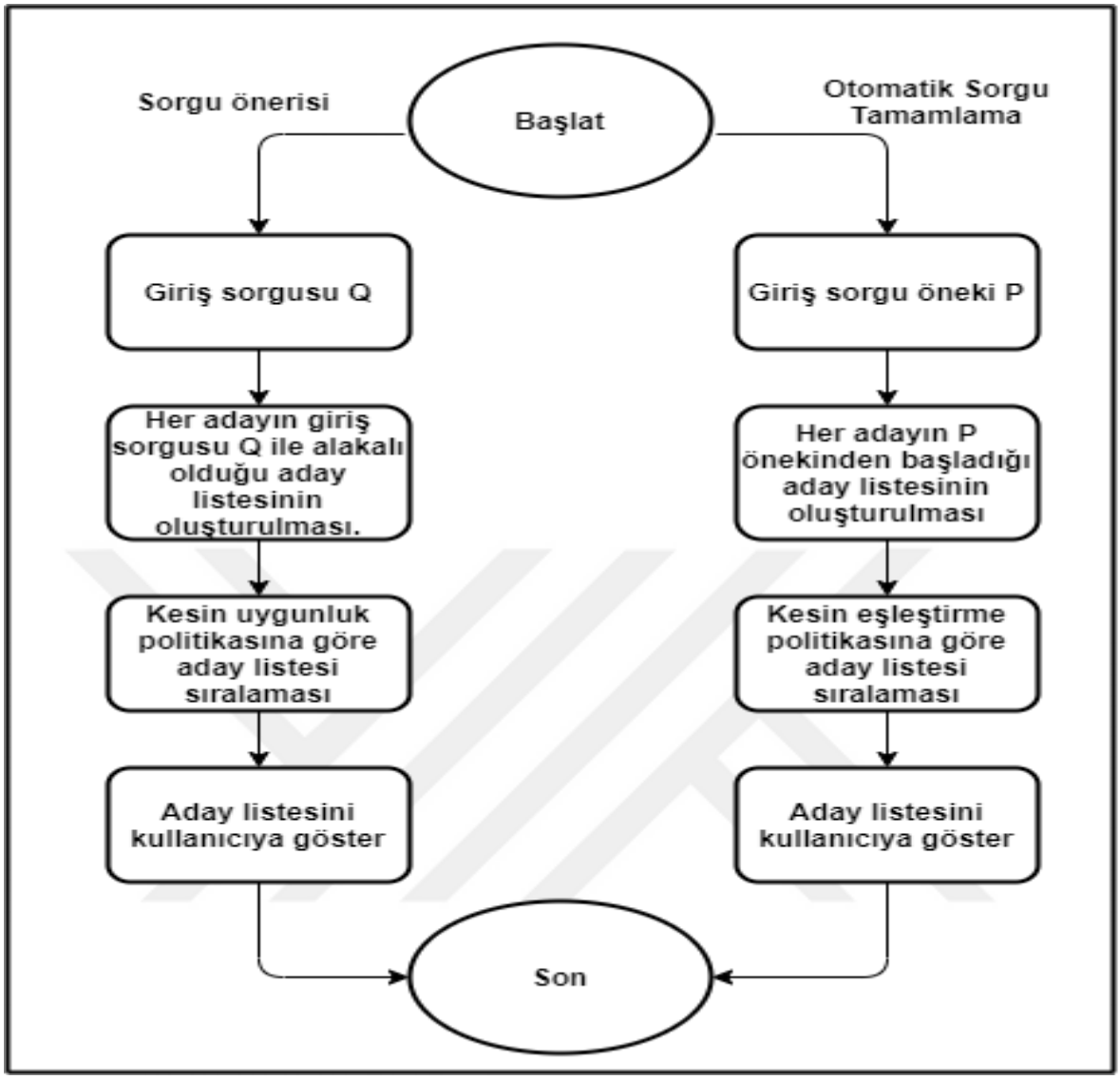
arkadaşlarının önerdiği [41] benzerlik yöntemi kullanılmıştır ve sıralama algoritması olarak konumsal n-gram benzerlik Eş. 3.1'deki gibi hesaplanmaktadır.

$$\text{SıralamaSkor} = \text{NGramBenzerlik}(Q_p^{'}, c^{'}) \tag{3.1}$$

Eş. 3.1'de, $Q_p^{'}$ gerçek sorgu son ekini gösterir ve $c^{'}$ aday son ekini gösterir.

3.6. Sorgu Önerileri

Sorgu önerileri sorgu tavsiyeleri olarak da bilinmektedir. Müşteri için sorguları tahmin etmek amacıyla öneri sisteminde sorgu geçmişinin nasıl kullanıldığını bilmek gereklidir. Sorgu önerisi yalnızca müşterinin bağlamsal bilgilerine değil, aynı zamanda müşterinin oturum geçmişine, mevcut eğilimlere, zamansal faktöre ve diğer birçok özelliğe de bağlıdır. Sorgu önerisi durumunda müşteri tanımlama mekanizması yeterli olmamaktadır. Sorgu önerisi bir kişiselleştirme sorunu değildir. Sorgu önerisi her orijinal girdiye karşı uygun sorguların bir listesini oluşturma işlemidir. Bu öneri kullanıcının ihtiyacını ifade etmekte güçlük çekmesi halinde onun için amaçlanan bir sorgunun oluşturulmasını hedeflemektedir. QAC, sorgu önerisinde katı bir eşleştirme politikası izleyebilir. Ayrıntılı biçimdeki bir sorgu önerisiyle QAC arasındaki fark Şekil 3.3'te gösterilmektedir.



Şekil 3.3. Sorgu önerisi ile QAC arasındaki fark

Kato vd. [42], QAC'nin giriş olarak önek aldığını ve bir sorgu kümesi oluşturduğunu ifade etmişlerdir. Ama sorgu önerisi, giriş olarak tek bir sorguyu almakta ve çıktı olarak bir dizi ilgili sorguyu üretmektedir. Baeza-Yates vd. [43], arama motorları için sorgu günlüklerini kullanarak sorgu önerileri yöntemini önermişlerdir. Önerilen yöntem bir girdi sorgusu alır ve sonuç olarak sorgu listesini tahmin eder. Anlamsal olarak benzer bir sorgu grubunu tanımlayan ve kümeler oluşturan kümeleme tekniği kullanılabilir. Kümeleme aynı kullanıcının arama motorunun sorgu geçmişinden önceki arama sorgularını kullanmaktadır. Wanyu Chen vd. [17], dikkat mekanizmalı hiyerarşik sinirsel ağa dayalı yeni sorgu öneri yöntemini önermiştir. Önerilen model sorgu öneri listesinin oluşturulması için kullanıcının kısa ve uzun vadeli geçmişini kullanmaktadır.

4. GELİŞTİRİLEN OTOMATİK SORGU TAMAMLAMA YÖNTEMİ

Bu bölümde sinirsel QAC, Uzun Kısa Süreli Bellek (Long Short Term Memory - LSTM) sinir ağları ve Tek Parçalı Kodlama (One Hot Encoding) yöntemleri ayrıntılı bir şekilde incelenmiştir. Daha sonra LSTM sinir ağını ve tek parçalı kodlama yöntemini kullanarak derin öğrenmeye dayalı bir QAC yöntemi önerilmiştir. Önerilen QAC sisteminin her bir alt bileşeni ayrıntılı olarak sunulmuştur.

4.1. Sinirsel QAC

Derin öğrenme tekniklerinin QAC uygulamalarında kullanımı son yıllarda önemli ölçüde artmıştır. QAC'nin farklı adımları için iki farklı temel yaklaşım vardır. Birincisi, Park ve Chiba [6] QAC sistemleri için yeni sinirsel dil modeli (Neural QAC) olarak geliştirilmiştir. Önerilen model geçmişte hiç gerçekleşmemiş yeni sorguların tamamlanması için popüler bir yaklaşım olmuştur. İkincisi ise, Mitra ve arkadaşları [20] tarafından kullanılan ve başlangıçta Shen ve arkadaşları [21] tarafından bilgi erişim işleri için önerilen QAC aday sıralamasına göre tasarlanmış anlamsal bir modeldir. QAC adayları oluşturma görevi için LSTM [6], GRU [10] gibi değiştirilmiş bir RNN türünün kullanıldığı QAC sıralama görevi [20] için ise CNN'in kullanıldığı görülmektedir.

4.2. Uzun Kısa Süreli Bellek Sinir Ağları

LSTM sinir ağı Tekrarlayan Sinir Ağlarının (Recurrent Neural Networks - RNN) değiştirilmiş bir versiyonudur. LSTM hızla artan (exploding) ve azalan gradyanların (vanishing gradients) RNN'deki sorunlarını çözmüştür. LSTM'nin metin analizi problemleri için en uygun sinir ağı olduğu değerlendirilmektedir. LSTM, son iki olay ve zamansal faktör arasındaki zaman gecikmesi durumunda hafızaya aldığı bilgileri hatırlamayı mümkün kılmaktadır [44]. Ayrıca, LSTM sinir ağları birçok zaman adımı boyunca bilgileri izlemek için kapalı bir hücreye sahiptir [45].

LSTM, RNN gibi bir zincir yapısına sahiptir ancak yinelenen modüler yapısı daha karmaşıktır. LSTM'de bu tekrar eden birimler, hücre içindeki bilgi akışını kontrol eden farklı etkileşim katmanları içermektedir. Olah, C. [46] LSTM kapılarının detaylı açıklamasını aşağıdaki gibi yapmıştır:

- İlk adım, ilgisiz geçmişi unutmak için hangi bilgilerin atılacağına karar verir,

- İkinci adım, yeni bilginin hangi bölümünün ilgili olduğuna ve hücre durumunda saklamak için kullanılacağına karar verir,
- Üçüncü adım, ilk iki adımın yardımıyla önceki bilgi parçalarını ve veri girişini alır ve bunları hücre durum değerlerini güncellemek için kullanır,
- Dördüncü adım, kodlanmış bilgiyi kontrol eden çıkış kapısı olarak bilinen çıktıyı bir sonraki zaman adımında girdi olarak ağa döndürür.

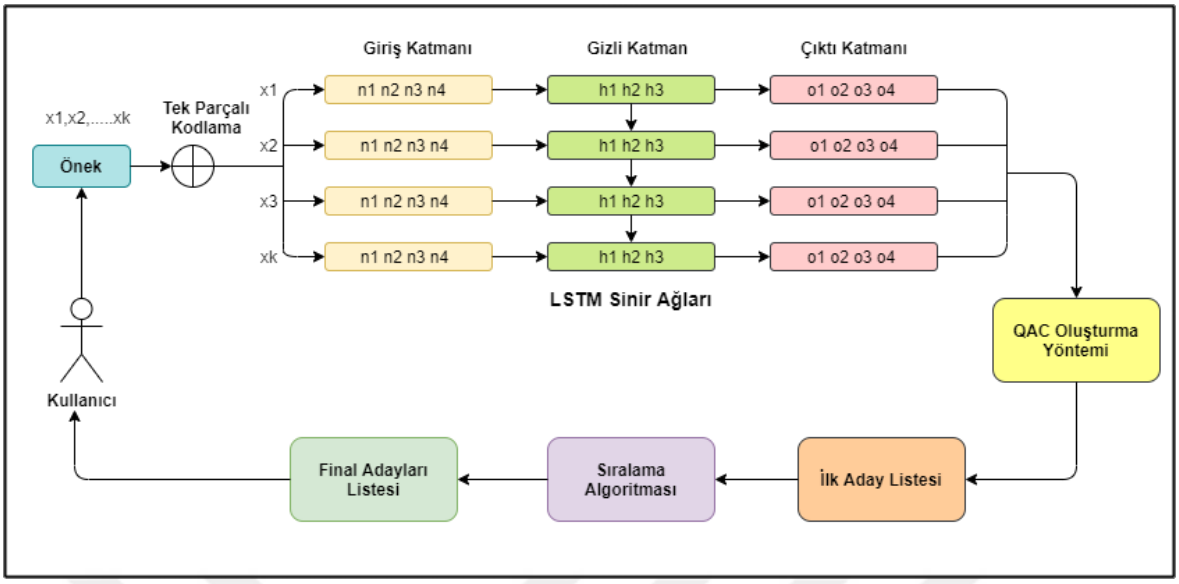
Ayrıca LSTM'nin diğer önemli bir özelliği de, zaman içinde yayılmaya izin veren kesintisiz bir gradyan akışına sahip olmasıdır.

4.3. Tek Parçalı Kodlama (One Hot Encoding)

Tek parçalı kodlama, makine öğrenimi ve Doğal Dil İşleme (Natural Language Processing - NLP) tabanlı uygulamalar için kategorik verilerle uğraşmak amacıyla yaygın olarak kullanılmaktadır. Özellikle NLP'de, kelime dağarcığını belirlemek için tek parçalı vektörden yararlanılır. Tek parçalı vektörün boyutu, karşılık gelen indeksin 1 değerine sahip olduğu ve indekslerin geri kalanının sıfır değerine sahip olduğu 1xN'dir [47,48]. QAC sisteminin ön işleme parçası olarak tek parçalı kodlama tekniği kullanılmaktadır.

4.4. Genel Mimari

Önerilen modelin mimarisi LSTM sinir ağına dayanmaktadır. Ağ, eğitim kümesinde karakter veri kümesi ile eğitilir ve ardından karakterlere göre yeni metin oluşturur. LSTM sinir ağı, her karakterin bir tamsayıya dönüştürülmesini gerektirir tek parçalı kodlanmış giriş alır. Bundan sonra karakter, karşılık gelen indeksin 1 değerine ve diğer indekslerin sıfır değerine sahip olduğu bir sütun vektörüne dönüştürülür. Model eğitildikten sonra her bir sorgunun girdi sorgu öneki Bölüm 3.3'te verilen etiketleme stratejisi kullanılarak çıkarılır. Ardından Bölüm 3.4'te verilen QAC listesi oluşturma tekniği kullanılarak her bir girdi sorgu önekine karşı tamamlama aday listesi oluşturulur. Daha sonra Bölüm 3.5'te gösterilen sıralama algoritması, yeniden sıralanan aday listesini almak için QAC aday listesine uygulanır. Bu tez çalışmasında önerilen modelin mimarisi Şekil 4.1'de verilmiştir.



Şekil 4.1. Önerilen QAC modeli

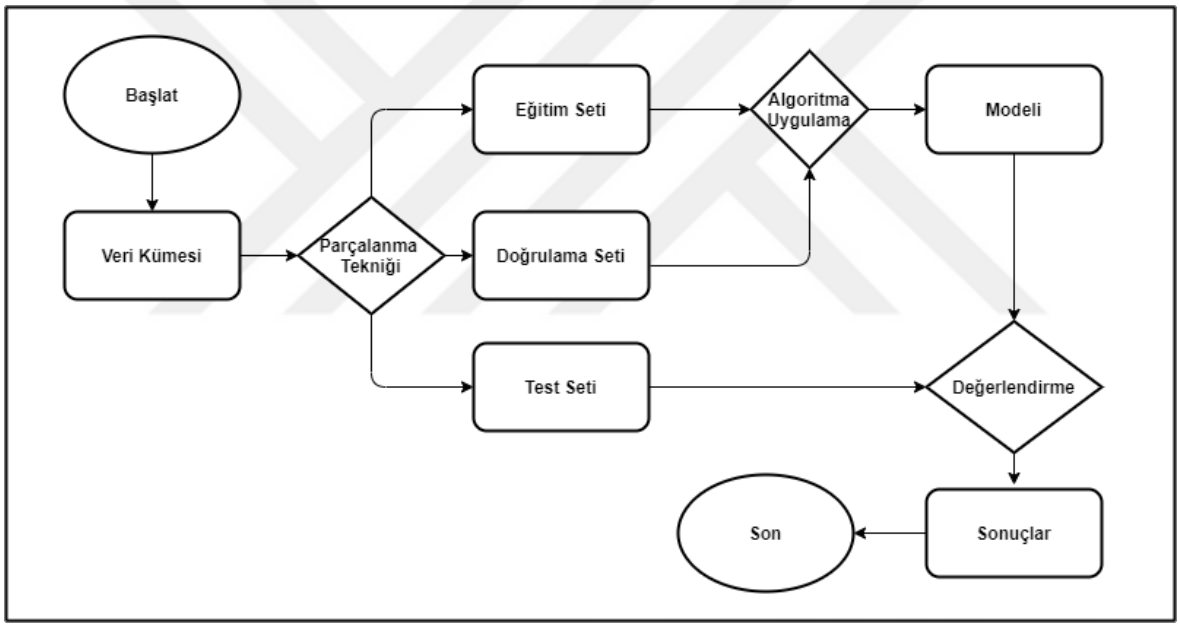
Şekil 4.1'de kullanıcı x1'den xk'ye kadar öneki girer (burada k doğal sayıdır). Daha sonra LSTM ağındaki besleme girişine tek parçalı kodlama tekniği uygulanır. LSTM sinir ağının üç ana katmanı, yani n değerli giriş katmanı, h değerlerine sahip gizli katman ve o değerlerinden oluşan çıktı katmanı vardır. QAC oluşturma yöntemi giriş önekine göre ilk aday listeyi almak için kullanılmıştır. Potansiyel kullanıcı için nihai aday listesini elde etmek amacıyla sıralama algoritması kullanılmıştır.

5. DENEYSEL SONUÇLAR

Bu bölümde, geliştirilen model için kullanılan veri kümeleri detaylı bir şekilde anlatılmıştır. Daha sonra, geliştirilen modelin test edilmesi amacıyla kullanılan ölçütler sunulmuştur. Son olarak, önerilen modelde kullanılan hiper parametreler hakkında bilgi verilmiştir.

5.1. Veri Kümeleri

QAC çalışmalarında literatürde kullanılan veri kümeleri detaylı bir şekilde incelenmiş ve AOL ve ORCAS veri kümelerinin kullanılmasına karar verilmiştir. Her iki veri kümesi hakkında ayrıntılı bilgiler sırasıyla Bölüm 5.1.1 ve Bölüm 5.1.2'de bulunmaktadır. Bu çalışmada veri kümesinin kullanımına ilişkin şematik gösterim Şekil 5.1'de verilmiştir.



Şekil 5.1. Veri kümesinin kullanımı

Şekil 5.1'de, önerilen modelin geliştirilmesi ve test edilmesi aşamalarında veri kümesinin kullanımı şematik olarak görülmektedir. İlk olarak, orijinal veri kümesi üç parçaya ayrılarak, eğitim, doğrulama ve test kümeleri oluşturulmuştur. Eğitim kümesi ve doğrulama kümesi önerilen modelin geliştirilmesinde kullanılmıştır. Test kümesi ise, modelin değerlendirilmesi ve ayarlanması için kullanılmıştır.

5.1.1. AOL veri kümesi

QAC tabanlı literatür çalışmalarının büyük bir çoğunluğu AOL veri kümesini [11] kullanmıştır [12]. Bu veri kümesi gerçek kullanıcılara dayalı gerçek bir sorgu geçmişi

sağlamaktadır. Buradaki gerçek sorgular istatistiksel öneme sahip olacak derecede büyük miktardadır. Bu veri kümesi yaklaşık 20 milyon sorgu içermektedir. Bu kümedeki nitelikler özellikleri: kullanıcı kimliği, sorgu, sorgu süresi, öge sıralaması ve tıklanan URL adresidir.

AOL veri kümesi, 01 Mart-31 Mayıs 2006 tarihleri arasındaki üç ayda toplanmıştır. Bu çalışmada, ASCII olmayan karakterlerin kaldırılması, 3 karakterden daha az sorgunun hariç tutulması [5] gibi bazı veri ön işleme adımları gerçekleştirilmiştir. Tüm karakterler küçük harfle değiştirilmiştir. Bir dizi farklı denemenin ardından AOL veri kümesini bölmek için bölümleme tekniği kullanılmıştır. Eğitim kümesi için 14 421 360 sorgu, doğrulama kümesi için 1 902 781 sorgu ve test kümesi için 4 036 493 sorgu alınmıştır. AOL veri kümesinin bölüm ayrıntıları Çizelge 5.1'de verilmiştir.

Çizelge 5.1. AOL veri kümesinin parçalanması

	Sorgu Sayısı	**Oran (%)**
Eğitim Seti	14 421 360	70,83
Doğrulama Seti	1 902 781	9,35
Test Seti	4 036 493	19,82
Toplam	20 360 634	100,00

5.1.2. ORCAS veri kümesi

Microsoft'un araştırma ekibi ORCAS veri kümesini bir QAC çalışması ile duyurmuştur [49,50]. Bu veri kümesi QAC ile ilgilidir ve özellikle tıklama özelliğine odaklanmıştır. Bu tez çalışmasında bu veri kümesinin sorgu özelliği ayrıntılı bir şekilde incelenmiştir. Ana veri kümesi dosyası dört nitelik içermektedir: sorgu kimliği, sorgu, belge kimliği ve URL. Bu veri kümesinde toplam 10,4 milyon arama sorgusu bulunmaktadır [50]. ORCAS veri kümesi QAC çalışmalarında en güncel veri kümesidir.

Bu çalışmada, ORCAS veri kümesi ayrıntılı olarak analiz edilmiştir ve bir dizi benzer sorgunun art arda bulunduğu tespit edilerek veri kümesinden çıkarılmıştır. Ardından, rastgele karıştırma stratejisi (random shuffling strategy) ORCAS veri kümesine uygulanmıştır. Bu strateji sistem içindeki önyargıyı önlemektedir. Ardından ASCII olmayan karakterlerin kaldırılması, 3 karakterden daha az sorgunun hariç tutulması [5] gibi bazı veri ön işleme adımları uygulanmıştır. Tüm karakterler küçük harfle değiştirilmiştir. Ardından bir veri kümesini parçalanarak eğitim kümesi için 7 277 061 sorgu, doğrulama kümesi için

1 039 435 sorgu ve test kümesi için 2 078 906 sorgu alınmıştır. ORCAS veri kümesinden elde edilen eğiti, doğrulama ve test veri kümelerine ait bilgiler Çizelge 5.2'de verilmiştir.

Çizelge 5.2. ORCAS veri kümesinin parçalanması

	Sorgu Sayısı	**Oran (%)**
Eğitim Seti	7 277 061	70
Doğrulama Seti	1 039 435	10
Test Seti	2 078 906	20
Toplam	10 395 402	100

5.2. Değerlendirme Ölçütleri

Derin öğrenmedeki en önemli adım, eğitim ve doğrulama tamamlandıktan sonra modeli değerlendirmektir. Modelin değerlendirilmesi, test veri kümesi kullanılarak gerçekleştirilmektedir. QAC ve bilgi erişim alanıyla ilgili detaylı bir literatür çalışması yapılarak kullanılan ölçütler incelenmiştir.

5.2.1. Ortalama karşılıklı sıra (Mean Reciprocal Rank - MRR)

Ortalama Karşılıklı Sıralama (Mean Reciprocal Rank - MRR) bilgi erişim çalışmalarında yaygın kullanılan bir değerlendirme ölçütüdür ve QAC çalışmalarında da yaygın kullanılmaktadır [4,5,6,7,8,9,13,14]. MRR Eş. 5.1'deki gibi hesaplanmaktadır.

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{q \in Q}\frac{1}{r_q} \tag{5.1}$$

Eş. 5.1'de, |Q| tüm öneklerin sayısını, r_q ise ilk ilgili elemanın sırasını göstermektedir.

5.2.2. Kısmi ortalama karşılıklı sıra (Partial Mean Reciprocal Rank - PMRR)

Park ve Chiba [6] tarafından kullanılan Kısmi Eşleşen Ortalama Karşılıklı Sıra (Partial-Matching Mean Reciprocal Rank - PMRR) değerlendirme ölçütü [5,10], QAC çalışmalarında yaygın kullanılmaktadır. PMRR Eş. 5.2'deki gibi hesaplanmaktadır.

$$\mathrm{PMRR} = \frac{1}{|Q|}\sum_{q \in Q}\frac{1}{pr_q} \tag{5.2}$$

Eş. 5.2'de, |Q| tüm öneklerin sayısını, pr_p ise ilk aday elemanın sırasını göstermektedir.

5.2.3. En üstteki ilk K elemanda başarı oranı

İlk K elemanda başarı oranı (Success Rate at Top-K or SR@K), tahmin edilen sorgu tamamlama adaylarının sıralama kalitesini değerlendirmek için kullanılan ve QAC

değerlendirme çalışmalarında kullanılan ölçüttür [9,27]. En iyi K tamamlama adaylarındaki ilgili adayların oranını hesaplamaktadır. Bu tez çalışmasında SR@1, SR@3 ve SR@5 kullanılmıştır.

5.2.4. Normalleştirilmiş azaltılmış toplam kazanç

Normalleştirilmiş Azaltılmış Toplam Kazanç (Normalized Discounted Cumulative Gain - NDCG), QAC için tamamlama adaylarının sıralamasını ölçmek için kullanılan ve QAC çalışmalarında yayın kullanılan değerlendirme ölçütüdür [9,15]. Ama öncelikle, adayların tamamlama listesinde sorgunun konumunu bularak sorgunun ilgililik düzeyini logaritmik olarak gösteren Azaltılmış Toplam Kazanç (Discounted Cumulative Gain - DCG) değerini bilmesi gerekmektedir. DCG Eş. 5.3'teki gibi hesaplanmaktadır.

$$\mathrm{DCG_k} = \sum_{q=1}^{Q} \frac{2^{\mathrm{uygunluk_p}} - 1}{\log_2(p+1)} \tag{5.3}$$

Eş. 5.3'te, p beklenen sorgunun sırasını göstermektedir. $\mathrm{Uygunluk_p}$ ($\mathrm{relevance_p}$), sorgunun p konumundaki uygunluğunu ifade etmektedir ve bu çalışmada 0 veya 1 olarak alınmıştır.

NDCG, DCG'nin normalleştirilmiş şeklidir ve Eş. 5.4'teki gibi hesaplanmaktadır.

$$\mathrm{NDCG_k} = \frac{\mathrm{DCG_k}}{\mathrm{IdealDCG_k}} \tag{5.4}$$

Eş. 5.4'te, IdealDCG, aday listesinin herhangi bir pozisyonunda mümkün olan maksimum değeri göstermektedir ve Eş. 5.5'teki gibi hesaplanmaktadır.

$$\mathrm{IdealDCG_k} = \sum_{q=1}^{Q} \frac{2^{\mathrm{revelance_p}} - 1}{\log_2(\mathrm{IdealRank}+1)} \tag{5.5}$$

Veri kümesindeki tüm sorguların genel NDCG kalite ölçüsü Eş. 5.6'daki gibi hesaplanmaktadır.

$$\mathrm{NDCG_Q} = \frac{\sum_{q=1}^{Q} \mathrm{NDCG_q}}{Q} \tag{5.6}$$

5.2.5. Ortalamaların ortalaması hassasiyet

Ortalamaların Ortalaması Hassasiyet (Mean Average Precision - MAP), bilgi erişim görevlerinin performansını ölçmek için kullanılan ve QAC çalışmalarında da kullanılan bir

değerlendirme ölçütüdür [9,51]. Bu ölçütten önce bilgi çıkarımı işlevinin hassasiyetini Eş. 5.7'deki gibi bilinmelidir.

$$\text{Precison} = \frac{|\{\text{ilgiliBelgeler}\} \cap \{\text{alinanBelgeler}\}|}{|\{\text{ilgiliBelgeler}\}|} \tag{5.7}$$

Daha sonra K konumundaki Ortalama Hassasiyet (Average Precision) Eş. 5.8'deki gibi tanımlanır.

$$\text{Ortalama Hassasiyet@K} = \frac{1}{\text{Zemin Gerçeği Pozitif}} \sum_{k}^{n} \text{Hassasiyeti@k} \times \text{uygunluk@k} \tag{5.8}$$

Bu çalışmada uygunluk@k 0 veya 1 olduğunda ve her bir sorgunun MAP skoru Eş. 5.9'deki şekilde tanımlanmıştır:

$$\text{MAP} = \sum_{r=1}^{R} \frac{\text{Ortalama Hassasiyet(r)}}{R} \tag{5.9}$$

Bu çalışmada sırasıyla MAP, MAP@1, MAP@3 ve MAP@5 kullanılmıştır.

5.2.6. Uygunluk (Relevancy)

Uygunluk, gerçek sorgu ile olası sorgu tamamlama adayları arasındaki benzerliği ölçmek için kullanılmaktadır. En az bir aday gerçek sorguyla kısmen veya tamamen aynı ise geliştirilen QAC modelinde değeri 1, aksi takdirde 0 olarak alınmıştır. Bu çalışmada, önce skor değerleri toplanmış, ardından uygunluk skorunu elde etmek için gerçek sorguların toplam sayısına bölünmüştür.

5.3. Geliştirilen Uygulama

Tamamlama listesinin olası adaylarını tahmin etmek amacıyla test verileri bölümünün giriş önekleri için Bölüm 3.1'de verilen etiketleme stratejisi kullanılmıştır. Bölüm 3.2'de verilen aday oluşturma yöntemi kullanılarak her bir giriş önekine karşı 10 sorgu adayı oluşturulmuştur. Bu çalışmada ilki AOL veri kümesi ve ikincisi ORCAS veri kümesi için olmak üzere iki model geliştirilmiştir. AOL veri kümesi yaklaşık 20 milyon sorgu içerirken, ORCAS veri kümesi yaklaşık 10,4 milyon sorgu içermektedir. Her iki veri kümesi hakkında detaylı bilgi sırasıyla Bölüm 5.1.1 ve Bölüm 5.1.2'de sunulmuştur.

LSTM ağı, modeli karakter karakter eğitmek ve doğrulamak için uygulanmıştır. Kodlanmış giriş vektörü LSTM modelinde kullanılmıştır. Giriş vektörü, AOL veri kümesi için 2 LSTM gizli katmanında 512 düğüm kullanılarak, ORCAS veri kümesi için ise 2 LSTM gizli katmanında 256 düğüm içermektedir. Çıkartma katmanında, AOL veri kümesi için 0,25 iken, ORCAS veri kümesi için 0,6 olarak alınmıştır.

Gradyan hızlı artma problemini önlemek için gradyanlar 5 değeriyle kesilmektedir. Sonraki karakterin tahmini için olasılık dağılımını veren SoftMax fonksiyonu yeniden sıralanan aday listesini almak için QAC aday listesine uygulanmıştır. Her iki veri kümesi için LSTM parametreleri Çizelge 5.3'te sunulmuştur.

Çizelge 5.3. LSTM hiper parametreleri

	AOL Veri Kümesi	**ORCAS Veri Kümesi**
Sorgu Sayısı (Number of Queries)	20 milyon	10,4 milyon
LSTM Katmanları (LSTM Layers)	2	2
Gizli Birimler (Hidden Units)	512	256
Çıkartma Değeri (Dropout Value)	0,25	0,6
Devir (Epoch)	4	4
Öğrenme Oranı (Learning Rate)	0,001	0,001
Gradyan Kırpma (Gradient Clipping)	5	5

6. DENEYSEL SONUÇLAR VE TARTIŞMA

Bu bölümde, önerilen modeller üzerinde kapsamlı deneyler yaptıktan sonra elde edilen sonuçlar ayrıntılı analiz sonuçları sunulmuştur. Elde edilen sonuçlar ile QAC sisteminin performansı ve kalitesi değerlendirilmiştir. Modellerin performansını değerlendirmek için uygunluk değerlendirme matrisi kullanılmıştır. Modellerin kalitesini değerlendirmek için kısmi veya tam eşleştirme stratejileri, başarı oranı, ortalamaların ortalaması hassasiyet ve normalize azaltılmış toplam kazanç kullanılmıştır. Her iki modeli popüler kelimeler üzerinden değerlendirilmiştir.

Test verileri bölümünde yer alan her bir sorgunun önekini elde etmek için Bölüm 3.3'te verilen etiketleme stratejisi kullanılmıştır. Ardından önerilen QAC modelinin performansı ve kalitesi belirlenen ölçütler kullanılarak değerlendirilmiştir. İlk olarak, önerilen modelin performansı Çizelge 6.1'de gösterildiği gibi sırasıyla 10K, 20K, 40K, 50K ve 100K olmak üzere beş farklı test sorgu kümesi üzerinde değerlendirilmiştir.

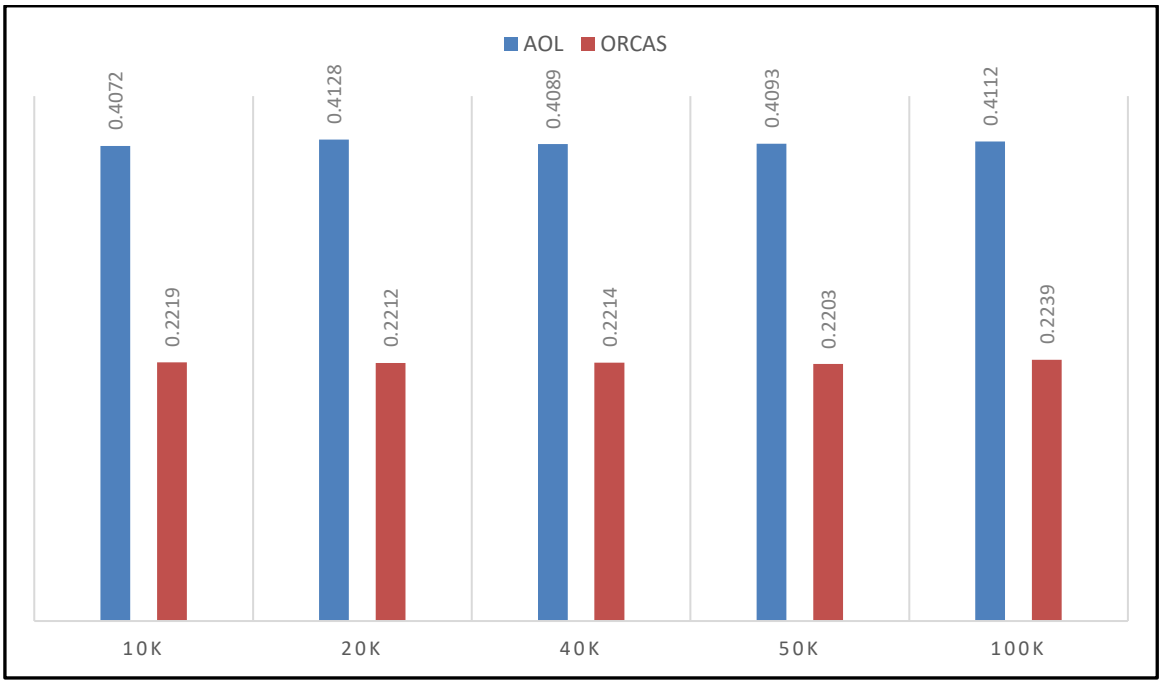
Çizelge 6.1. QAC modellerinin genel performansı

Örneklem	Veri Kümesi	Eşleştirme Stratejisi	Uygunluk Skoru
10K	AOL	PMRR	0,4072
		MRR	0,1724
	ORCAS	PMRR	0,2219
		MRR	0,0314
20K	AOL	PMRR	0,4128
		MRR	0,1836
	ORCAS	PMRR	0,2212
		MRR	0,0295
40K	AOL	PMRR	0,4089
		MRR	0,1859
	ORCAS	PMRR	0,2214
		MRR	0,0288
50K	AOL	PMRR	0,4093
		MRR	0,1882
	ORCAS	PMRR	0,2203
		MRR	0,0294
100K	AOL	PMRR	0,4112
		MRR	0,1878
	ORCAS	PMRR	0,2239
		MRR	0,0298

Çizelge 6.1'de her iki QAC modelinin test sorgu kümesinin beş farklı örneği üzerindeki genel performansı verilmiştir. Uygunluk skoru, her iki veri kümesinde de her bir eşleştirme stratejisi kullanılarak hesaplanmıştır. Kullanılan veriler test verileri olup daha önce model tarafından görülmemiş verilerdir.

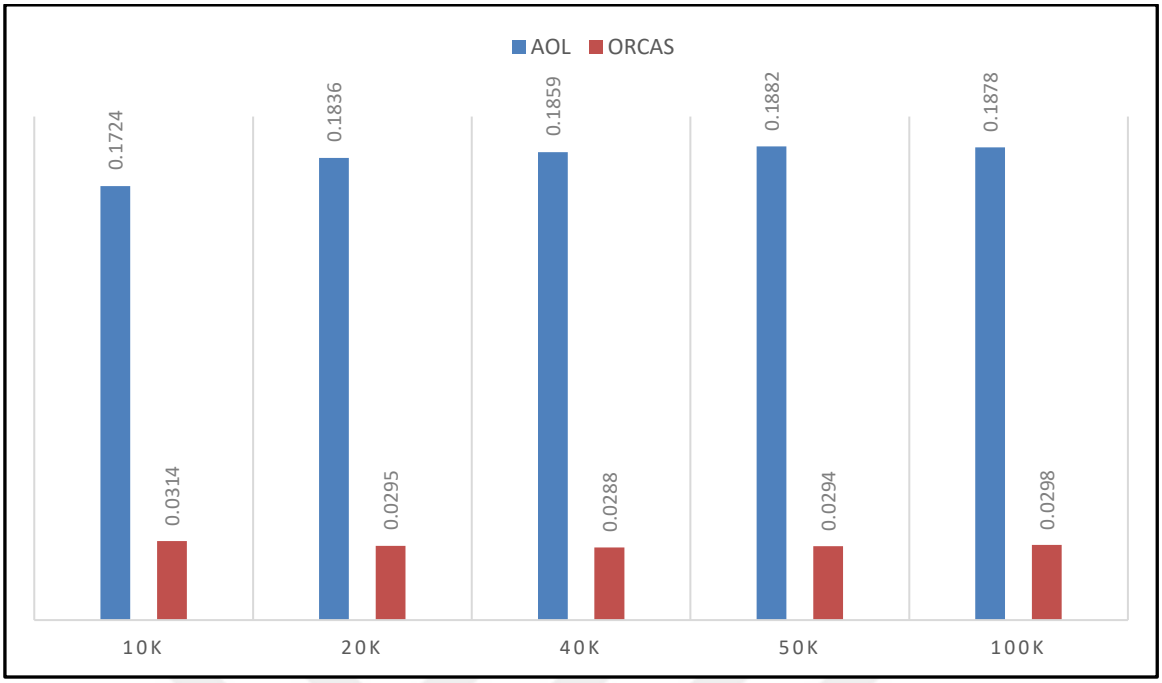
Elde edilen sonuçlardan, AOL veri kümesindeki genel uygunluk skorunun ORCAS veri kümesinden çok daha iyi olduğu görülmektedir. Bu, AOL model performansının ORCAS modelinden daha yüksek olduğunu ve AOL veri kümesinin QAC uygulamaları için daha uygun veri içeriğine sahip olduğunu ifade etmektedir. Kısmi ve tam eşleştirme stratejileri

kullanarak her iki modelin, uygunluk ölçütü üzerindeki performans karşılaştırması Şekil 6.1 ve Şekil 6.2'deki grafiklerde sunulmuştur. Önerilen yaklaşımın performansını, kısmi veya tam eşleştirme stratejilerine göre uygunluk skoru hesaplayan uygunluk matrisi kullanılarak değerlendirilmiştir. Şekil 6.1, kısmi eşleştirme stratejisindeki test sorgusu kümesinin beş farklı örneğinin ilgi düzeyi puanını, Şekil 6.2 ise, tam eşleştirme stratejisindeki test sorgu kümesinin beş farklı örneğinin ilgi düzeyi puanını göstermektedir.



Şekil 6.1. Kısmi eşleştirme stratejisi PMRR kullanılarak test sorgu kümesinin beş farklı örneği üzerinde her iki modelin performans analizi.

Şekil 6.1'de, AOL ve ORCAS modellerinde beş farklı test sorgusu kümesinin performansını analiz etmek için PMRR olan kısmi eşleştirme stratejisini kullanılmıştır. AOL modelinin en yüksek uygunluk skorunun test sorgu kümesinin 20K ve aynı modelin en düşük uygunluk skorunun 10K olduğu görülmektedir. Öte yandan, ORCAS modelinin 100 bin test sorgu kümesi örneğinde en yüksek uygunluk skoruna ve aynı modelin 50 bin örneğinde en düşük uygunluk skoruna ulaştığı görülmektedir. Bundan başka, AOL modelinin genel uygunluk skorunun, kısmi eşleştirme stratejisindeki ORCAS modelinin genel uygunluk skorundan daha iyi olduğunu görülmektedir.



Şekil 6.2. Tam eşleştirme stratejisi MRR kullanılarak test sorgu kümesinin beş farklı örneği üzerinde her iki modelin performans analizi.

Şekil 6.2'de, AOL ve ORCAS modellerinde beş farklı test sorgusu kümesinin performansını analiz etmek için tam eşleştirme stratejisi MRR kullanılmıştır. AOL modelinin en yüksek uygunluk skoruna test sorgu kümesinin 50K örneğinde ve aynı modelin en düşük uygunluk skoruna ise 10K örneğinde ulaşılmıştır. Öte yandan, ORCAS modelinin test sorgu kümesinin en yüksek uygunluk skorunun 10K örnekleminde, aynı modelin en düşük uygunluk skorunun ise, 40 bin örnekleminde olduğu görülmektedir. Bunun yanı sıra, AOL modelinin genel uygunluk skorunun, tam eşleştirme stratejisine ilişkin ORCAS modelinin genel uygunluk puanından daha iyi olduğu görülmektedir.

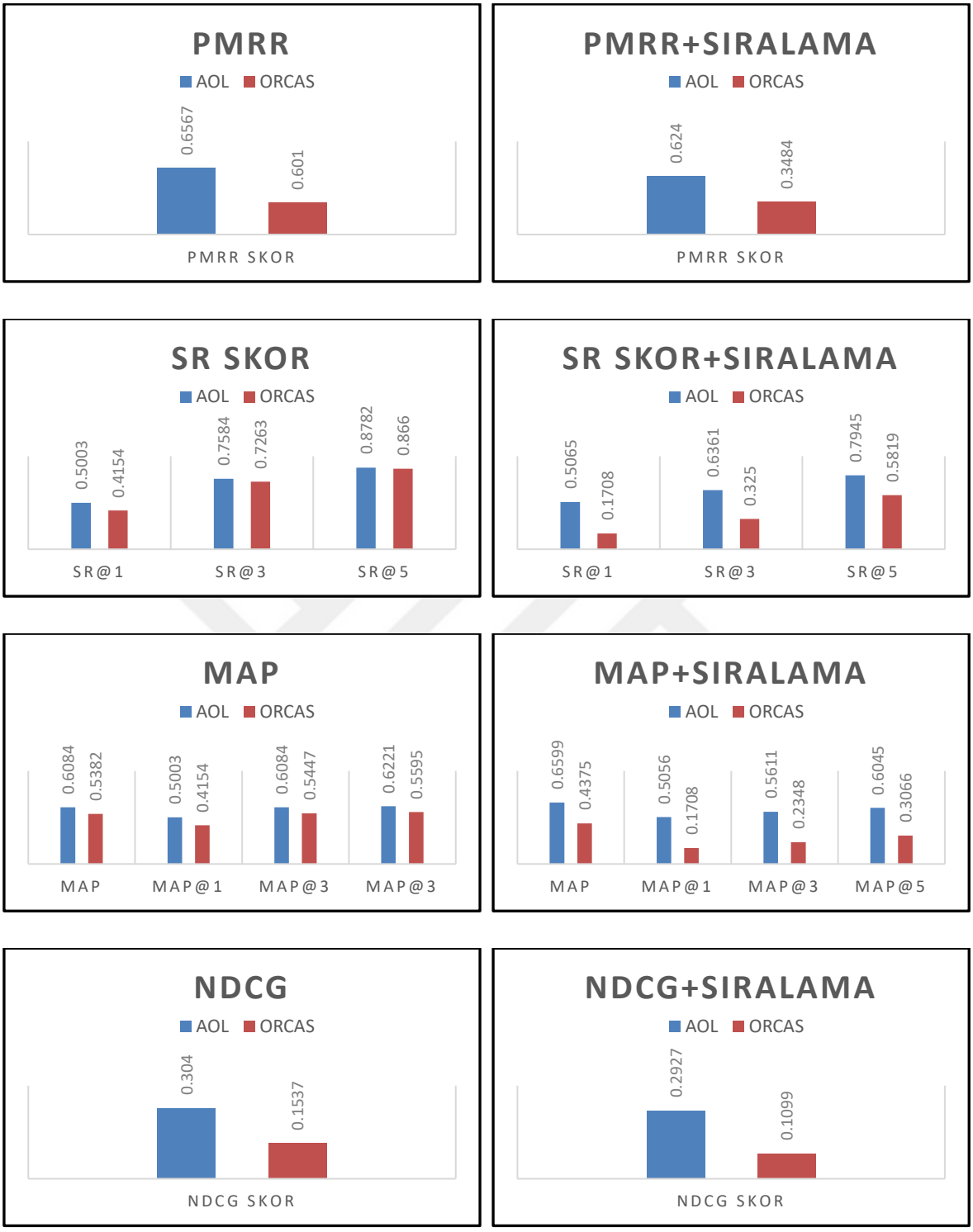
Bundan sonra, geliştirilen QAC modelinin kalitesini değerlendirmek için test setinden 100K örneği ele alınmıştır. QAC modelinin kalitesini değerlendirmek için kısmi veya tam eşleştirme stratejileri, başarı oranı, ortalamaların ortalaması hassasiyet ve normalleştirilmiş azaltılmış toplam kazanç ölçütleri kullanılmıştır. Önerilen modelimi analiz etmek için sonuçlar iki farklı şekilde kategorize edilmiştir.

İlk olarak, 100K test örneğinde, kısmi eşleştirme stratejisi olan PMRR ölçütünün 1, 3 ve 5 aday tamamlama listesi pozisyonundaki başarı oranı, 1, 3 ve 5 aday tamamlama listesi pozisyonundaki ortalama hassasiyet ve normalleştirilmiş azaltılmış toplam kazanç ile değerlendirilmiştir. Elde edilen sonuçlar Çizelge 6.2'de ve Şekil 6.3'te sunulmaktadır.

Çizelge 6.2. Önerilen modelin PMRR metriği kullanılarak değerlendirilmesi

	Veri Kümesi	**PMRR**	**SR@1**	**SR@3**	**SR@5**	**NDCG**	**MAP**	**MAP@1**	**MAP@3**	**MAP@5**
Model	AOL	0,656	0,500	0,758	0,878	0,304	0,608	0,500	0,608	0,622
Model+ Sıralama		0,624	0,506	0,636	0,794	0,292	0,659	0,5065	0,561	0,604
Model	ORCAS	0,601	0,415	0,726	0,866	0,153	0,538	0,415	0,544	0,559
Model+ Sıralama		0,348	0,170	0,325	0,581	0,109	0,437	0,170	0,234	0,306

Çizelge 6.2'de önerilen modelin PMRR metriği kullanılarak değerlendirilmesi. İlk olarak, 100K test örneğinin PMRR skoru hesaplanmıştır. Ardından bu skoru kullanarak diğer tüm değerlendirme metrikleri hesaplanmıştır. Sıralama yönteminin her iki veri kümesinin performansını düşürdüğü görülmüştür. Daha sonra Model+Sıralama olarak bir birlikte kullanıldığında test sonuçları elde edilmiştir. AOL veri kümesi, değerlendirme PMRR skoru kullanılarak gerçekleştirildiğinde ORCAS veri kümesinden daha iyi performans ve sorgu alma işlemi kalitesi sağlamaktadır.



Şekil 6.3. Bu çalışmada kısmi eşleştirme stratejisi PMRR kullanılmıştır.

Şekil 6.3'te kısmi eşleştirme stratejisi PMRR kullanılmıştır. Sıralayıcı kullanılmadan ölçüt skorları sol tarafta gösterilirken, sıralayıcı kullanılarak elde edilen skorlara ait veriler sağ tarafta gösterilmektedir.

Şekil 6.3'te PMRR ile PMRR+SIRALAMA karşılaştırıldığında, sıralama algoritmasının her iki modelin kalitesini düşürdüğünü, ancak sıralama algoritması kullanıldığında ORCAS modelinin kalitesinin AOL modelinden daha fazla etkilendiği görülmektedir. SR Skoru ile SR SKOR+SIRALAMA karşılaştırıldığında sıralama algoritmasının her iki modelin kalitesini düşürdüğü görülmüştür. Bununla birlikte, SR'nin pozisyonunun artmasıyla modellerin kalitesi artmıştır. Bu da SR@5'in kalitesinin SR@3'ten daha iyi olduğu ve SR@3'ün kalitesinin SR@1'den daha iyi olduğu anlamına gelmektedir. Aynı zamanda MAP Skoru ile MAP SKOR+SIRALAMA karşılaştırıldığında sıralamanın her iki modelin kalitesini düşürdüğü görülmektedir. Her iki modelin en düşük kalitesinin MAP@1'de ve en yüksek model kalitesinin ise MAP@5'te olduğunu görülmüştür. NDCG ile NDCG+SIRALAMA karşılaştırıldığında sıralama algoritmasının her iki modelin kalitesini düşürdüğü görülmektedir. Ancak, AOL modelinin NDCG skoru her iki durumda da ORCAS modelinin NDCG skorundan daha iyi olduğu görülmüştür.

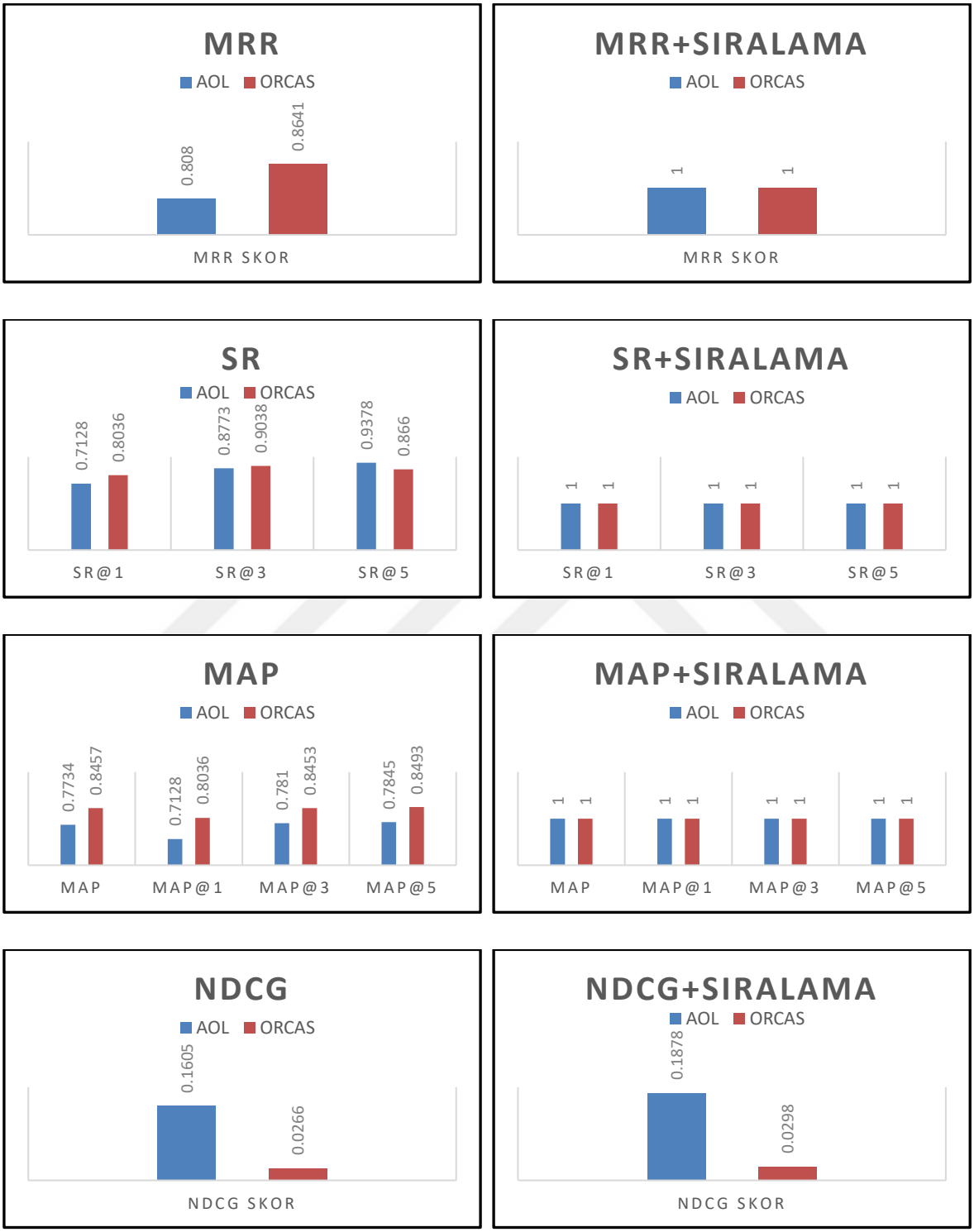
İlk olarak 100K test örneği, tam eşleştirme stratejisi olan MRR için 1, 3 ve 5 aday tamamlama listesi pozisyonundaki başarı oranı, 1, 3 ve 5 aday tamamlama listesi pozisyonundaki ortalama hassasiyet ve normalleştirilmiş azaltılmış toplam kazanç ile değerlendirilmiştir. Elde edilen sonuçlar Çizelge 6.3'te ve Şekil 6.4'te görülmektedir.

Çizelge 6.3. Önerilen modelin MRR metriği kullanılarak değerlendirilmesi

	Veri Kümesi	**MRR**	**SR@1**	**SR@3**	**SR@5**	**NDCG**	**MAP**	**MAP@1**	**MAP@3**	**MAP@5**
Model	AOL	0,808	0,712	0,877	0,937	0,160	0,773	0,712	0,781	0,784
Model+Sıralama		1,0	1,0	1,0	1,0	0,187	1,0	1,0	1,0	1,0
Model	ORCAS	0,864	0,803	0,903	0,945	0,026	0,845	0,803	0,845	0,849
Model+Sıralama		1,0	1,0	1,0	1,0	0,029	1,0	1,0	1,0	1,0

Çizelge 6.3'te önerilen modelin MRR metriği kullanılarak değerlendirilmesi. İlk olarak 100 bin test örneğinin MRR skoru hesaplanmıştır. Ardından bu puan kullanılarak diğer tüm değerlendirme metrikleri hesaplanmıştır. Modellerle bir sıralama yönteminin (Ranker) kullanılması QAC sisteminin performansını artırmıştır. Sıralama yöntemini (Model+Sıralama) içeren ayrı bir model kullanılarak testler yapılmıştır. ORCAS veri kümesi

değerlendirme MRR skoru kullanılarak gerçekleştirildiğinde AOL veri kümesinden daha iyi performans elde edilmiştir.



Şekil 6.4. Bu çalışmada tam eşleştirme stratejisi MRR kullanılmıştır.

Şekil 6.4'te tam eşleştirme stratejisi MRR kullanılmıştır. Sıralayıcı olmayan metrik skorları sol tarafta gösterilirken, sıralayıcı ile metrik skorları sağ tarafta gösterilmektedir.

Şekil 6.4'te MRR ile MRR+SIRALAMA karşılaştırılarak sıralama algoritmasının, mutlak 1 olan her iki modelin kalitesini artırdığı görülmüştür. ORCAS modelinin kalitesi, sıralama algoritması uygulanmadan AOL modelinden daha iyidir. SR Skoru ile SR SKOR+SIRALAMA karşılaştırıldığında sıralama algoritmasının her iki modelin kalitesini mutlak 1'e yükselttiği görülmektedir. Ayrıca SR SKOR durumunda SR'nin pozisyonunun artmasıyla modellerin kalitesi de artmıştır ki, bu da SR@5'in kalitesinin SR@3'ten daha iyi olduğu ve SR@3'ün kalitesinin SR@1'den daha iyi olduğu anlamına gelmektedir. SR SKOR+SIRALAMA durumunda ise, SR@1, SR@3 ve SR@5'in kalitesi mutlak 1'dir. MAP SKOR'u ile MAP SKOR+SIRALAMA karşılaştırıldığında sıralamanın her iki modelin de mutlak 1 olan kalitesini yükselttiği görülmüştür. Sıralayıcı olmadan, her iki modelin en düşük kalitesinin MAP@1'de ve en yüksek model kalitesinin ise MAP@5'te olduğu görülmektedir. NDCG ile NDCG +SIRALAMA karşılaştırıldığında sıralama algoritmasının her iki modelin kalitesini artırdığı görülmüştür ve AOL modelinin NDCG skoru sıralı veya sırasız her iki durumda da ORCAS modelinin NDCG skorundan daha iyidir.

Elde edilen bulgulardan yola çıkarak MRR skoru kullanılarak çıkarılan sonuçların PMRR skoru kullanılarak elde edilen sonuçlardan daha iyi olduğu görülmektedir. Aynı zamanda, MRR'nin uygunluk skoru PMRR uygunluk skoruna kıyasla daha düşüktür. Bunun ana nedeni, Bölüm 3.3'te verilen PMRR puanını kullanarak çıkardığımız öneklerin çıkarılması olarak değerlendirilmiştir. Ayrıca QAC adaylarının gerçek sorgu ile kısmi eşleşmesinin, AOL veri kümesinde iyi performans gösterdiği görülmüştür. Ayrıca, QAC adaylarının gerçek sorguyla tam olarak eşleştirilmesinin ORCAS veri kümesinde iyi çalıştığı da görülmektedir.

Popüler kelimelerin kullanılması, ticari arama motorlarının önemli bir özelliğidir. AOL ve ORCAS modellerinde ilk beş aday listesi dört farklı popüler kelime göz önünde bulundurarak oluşturuldu ve elde edilen sonuçlar Çizelge 6.4'te sunulmuştur.

Çizelge 6.4. Popüler kelime öneklerinde en iyi 5 aday listesini oluşturduk

Model	**Popüler Kelimelerin Önekleri (Prefixes Of Popular Words)**			
	www	transl	weat	gam
AOL	*www.google.com* *www.myspace.co* *www.chicagotourist.com* *www.myspace.com* *www.bigblood.com*	*translation* *translator* *translots* *translations.com* *translator*	*weather* *weather state university* *weather.com* *weather specials* *weather sales the parents*	*games.com* *game set* *game* *game tribune* *games*
ORCAS	*www.chestructures.com* *wwww.chologina.gov* *www.shares.com* *wwww.altichind* www.games.com	*translate* *translen chrome canada* *translatory* *translet for windows 10* *translage chees plant*	*weather from windows 10* *weather* *weather complaints* *weather format* *weather star*	*game personal calculator* *gaming settertinss changes* *game* *gam and tennies* *game test*

Çizelge 6.4'te önekli dört farklı popüler kelime (www.google.com, www), (translator, transl), (weather, weat) ve (game, gam) göz önünde bulundurarak AOL ve ORCAS modellerinde ilk beş aday listesi oluşturulduğunda elde edilen sonuçlar.

Çizelge 6.4'te, AOL ve ORCAS modelleri için dört popüler kelimenin ilk beş tamamlama aday listesini görülmektedir. Burada popüler kelimelerin önekleri kullanılmıştır. Bu popüler kelimeler Bölüm 3.3'te verildiği gibi, etiketleme stratejisinden veya önek çıkarma tekniğinden yararlanılarak elde edilmiştir. ORCAS modeli, "www" giriş önekindeki ilk beş aday tamamlama listesinde "www.google.com" popüler sorgusunu oluşturmadı ama AOL modeli, aynı girdideki ilk beş aday tamamlama listesinde "www.google.com" popüler sorgusunu oluşturmuştur. Benzer şekilde, ORCAS modeli bir "translator" popüler sorgu oluşturmadı fakat tamamlama listesinin adayları anlamsal olarak beklenen popüler sorguya çok yakın çıkmıştır. Kalan iki modeldeki aday tamamlama listeleri başarılı düzeyde oluşmuştur. Bu deneysel sonuçlar, AOL modelinin popüler kelimeler üzerindeki genel performansının ORCAS modelinden daha iyi olduğunu göstermektedir.

7. SONUÇ

Bu çalışmada, LSTM tabanlı QAC girdi önekini kullanarak bir sorgu tamamlama listesi oluşturmak için yeni bir model önerilmiştir. Önerilen LSTM tabanlı QAC modeli, AOL ve ORCAS veri kümeleri kullanılarak kapsamlı bir şekilde test edilmiştir. Önerilen modellerin performansı, hem AOL hem de ORCAS veri setlerinin 10K, 20K, 40K, 50K ve 100K test sorgu kümesinin beş farklı örneği kullanılarak kısmi eşleştirme ve tam eşleştirme stratejilerindeki sonuçları incelenmiştir.

Kısmi eşleştirme stratejisinde, AOL modelinin en yüksek performansını test sorgu kümesinin 20K örneğinde, en düşük performansını ise 10K örneğinde gösterdiği saptanmıştır. Öte yandan, ORCAS modelinin en yüksek performansını 100K test sorgu kümesi örneğinde, en düşük performansını ise 50k örneğinde gösterdiği tespit edilmiştir.

Tam eşleme stratejisinde, AOL modelinin en yüksek performansının test sorgu kümesinin 50K örneğinde, en düşük performansının ise 10K örneğinde olduğu görülmüştür. Aynı şekilde, ORCAS modelinin 10K test sorgu kümesi örneğinde en yüksek performansı, 40k örneğinde ise en düşük performansı gösterdiği görülmüştür. Her iki modelin de genel performansı kısmi eşleme stratejisinde daha iyi olmuştur. Önerilen modellerin kalitesi hem AOL hem de ORCAS veri kümelerinin 100 bin test sorgu kümesi örneği ile değerlendirilmiştir. Önerilen QAC modelinin başarısı tam eşleştirme stratejisinde daha iyi olmuştur.

Popüler kelimelerin kullanılması ticari arama motorlarının önemli bir özelliğidir. AOL ve ORCAS modellerinde popüler kelimeler üretilerek deneysel çalışmalar yapılmıştır. Deneysel bulgular, AOL modelinde oluşturulan popüler kelime tamamlama listesinin ORCAS modelinde aynı tamamlama listesinin oluşturulmasından daha etkili olduğunu göstermiştir.

Önerilen QAC modeli, PMRR ve MRR skorları kullanılarak değerlendirilmiştir. QAC modelinin performansı uygunluk skoru kullanılarak değerlendirilmiştir. Geliştirilen QAC modelinin kalitesi, kısmi ve tam eşleştirme stratejileri, başarı oranı, ortalamaların ortalaması hassasiyet ve normalleştirilmiş azaltılmış toplam kazanç kullanılarak değerlendirilmiştir. Elde edilen deneysel sonuçlar MRR skoru kullanılarak çıkarılan sonuçların PMRR skoru

kullanılarak elde edilen sonuçlardan daha iyi olduğunu göstermiştir. Ayrıca, MRR'nin uygunluk skorunun PMRR uygunluk skoruna kıyasla daha düşük olduğu görülmüştür.

KAYNAKLAR

1. İnternet: Jakob, Nielsen. (2011). *How long do users stay on web pages*. Nielsen norman group. Web: https://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/. Son Erişim Tarihi: 03.03.2021.
2. Fei, Cai. and Rijke, M. D (2016). A survey of query auto-completion in information retrieval. *Foundations and Trends in Information Retrieval*, 10(4), 273-363.
3. Yin, D., Hu, Y., Tang, J., Daly, T., Zhou, M., Ouyang, H., and Chang, Y. (2016). *Ranking relevance in yahoo search*. Paper presented at the 2016 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA, 323-332.
4. Bar-Yossef, Z., and Kraus, N. (2011). *Context-sensitive query auto-completion*. Paper presented at the 2011 Proceedings of the 20th International Conference on World Wide Web, Hyderabad, India, 107-116.
5. Kim, G. (2019). *Subword language model for query auto-completion*. Paper presented at the 2019 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 5022–5032.
6. Park, D. H., and Chiba, R. (2017). *A neural language model for query auto-completion*. Paper presented at the 2017 Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 1189-1192.
7. Shokouhi, M., and Radinsky, K. (2012). *Time-sensitive query auto-completion*. Paper presented at the 2012 Proceedings of the 35th International ACM SIGIR Donference on Research and Development in Information Retrieval, Oregon, USA, 601-610.
8. Shokouhi, M. (2013). *Learning to personalize query auto-completion*. Paper presented at the 2013 Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 103-112.
9. İnternet: Kannadasan, M. R., and Aslanyan, G. (2019). *Personalized query auto-completion through a lightweight representation of the user context.* Web: https://arxiv.org/pdf/1905.01386.pdf. Son Erişim Tarihi: 02.09.2021.

10. Fiorini, N., and Lu, Z. (2018). *Personalized neural language models for real-world query auto-completion.* Paper presented at the 2018 Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Louisiana, US, 208–215.
11. Pass, G., Chowdhury, A., and Torgeson, C. (2006). *A picture of search*. Paper presented at the 2006 Proceedings of the 1st International Conference on Scalable Information Systems, Hong Kong, 1-7.
12. İnternet: Pass. G. (2006). *AOL user collections*. Web: http://www.cim.mcgill.ca/~dudek/206/Logs/AOL-user-ct-collection/. Son Erişim Tarihi: 02.09.2020.
13. Di Santo, G., McCreadie, R., Macdonald, C., and Ounis, I. (2015). *Comparing approaches for query autocompletion*. Paper presented at the 2015 Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 775-778.
14. Jiang, D., Chen, H., and Cai, F. (2017). Exploiting query's temporal patterns for query autocompletion. *Mathematical Problems in Engineering*, 2017(7490879), 1-8.
15. Addepalli, B., Li, H., and Dueck, D. (2019). *Model selection and hyperparameter tuning in maps query auto-completion ranking.* Paper presented at the 2019 28th ACM International Conference on Information and Knowledge Management, Beijing, China.
16. Rossiello, G., Caputo, A., Basile, P., and Semeraro, G. (2019). Modeling concepts and their relationships for corpus-based query auto-completion. *Open Computer Science*, 9(1), 212-225.
17. Chen, W., Cai, F., Chen, H., and Rijke, M. D. (2020). Hierarchical neural query suggestion with an attention mechanism. *Information Processing and Management*, 57(6).
18. Jaech, A., and Ostendorf, M. (2018). *Personalized language model for query auto-completion.* Paper presented at the 2018 Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 700–705.
19. Whiting, S., and Jose, J. M. (2014). *Recent and robust query auto-completion*. Paper presented at the 2014 Proceedings of the 23rd International Conference on World Wide Web, Seoul, Korea, 971-982.

20. Mitra, B., and Craswell, N. (2015). *Query auto-completion for rare prefixes*. Paper presented at the 2015 Proceedings of the 24th ACM International Conference on Information and Knowledge Management, Melbourne, Australia, 1755-1758.
21. Shen, Y., He, X., Gao, J., Deng, L., and Mesnil, G. (2014). *Learning semantic representations using convolutional neural networks for web search*. Paper presented at the 2014 Proceedings of the 23rd International Conference on World Wide Web, Seoul, Korea, 373-374.
22. Wang, S., Guo, W., Gao, H., and Long, B. (2020). *Efficient neural query auto-completion.* Paper presented at the 2020 Proceedings of the 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, 2797-2804.
23. Gog, S., Pibiri, G. E., and Venturini, R. (2020). *Efficient and effective query auto-completion*. Paper presented at the 2020 Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, China, 2271-2280.
24. Li, Y., Dong, A., Wang, H., Deng, H., Chang, Y., and Zhai, C. (2014). *A two-dimensional click model for query auto-completion*. Paper presented at the 2014 Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, Gold Coast Queensland, Australia, 455-464.
25. Cai, F., Reinanda, R., and Rijke, M. D. (2016). Diversifying query auto-completion. *ACM Transactions on Information Systems*, 34(4), 1-33.
26. Shao, T., Chen, H., and Chen, W. (2018). Query auto-completion based on word2vec semantic similarity. *In Journal of Physics: Conference Series*, 1004(1).
27. Jiang, J. Y., Ke, Y. Y., Chien, P. Y., and Cheng, P. J. (2014). *Learning user reformulation behavior for query auto-completion*. Paper presented at the 2014 Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, Gold Coast Queensland, Australia, 445-454.
28. Vargas, S., Blanco, R., and Mika, P. (2016). *Term-by-term query auto-completion for mobile search.* Paper presented at the 2016 Proceedings of the 9th ACM International Conference on Web Search and Data Mining, San Francisco California, USA, 143-152.
29. Krishnan, U., Moffat, A., and Zobel, J. (2017). *A taxonomy of query auto completion modes*. Paper presented at the 2017 Proceedings of the 22nd Australasian Document Computing Symposium, Brisbane QLD, Australia, 1-8.

30. Arkoudas, K., and Yahya, M. (2019). *Semantically Driven Auto-completion*. Paper presented at the 2019 Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 2693-2701.
31. Tahery, S., and Farzi, S. (2020). Customized query auto-completion and suggestion: A review. *Information Systems,* 87(101415).
32. Dehghani, M., Rothe, S., Alfonseca, E., and Fleury, P. (2017). *Learning to attend, copy, and generate for session-based query suggestion*. Paper presented at the 2017 Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 1747-1756.
33. Li, L., Deng, H., Chen, J., and Chang, Y. (2017). *Learning parametric models for context-aware query auto-completion via hawkes processes*. Paper presented at the 2017 Proceedings of the 10th ACM International Conference on Web Search and Data Mining, Cambridge, UK, 131-139.
34. Hu, Y., Xiao, C., and Ishikawa, Y. (2018). *Context-sensitive query auto-completion with knowledge base.* Paper presented at the 2018 The 10th Forum on Data Engineering and Information Management, Awara, Japan, 4-6.
35. Wang, Y., Ouyang, H., Deng, H., and Chang, Y. (2017). Learning online trends for interactive query auto-completion. *IEEE Transactions on Knowledge and Data Engineering,* 29(11), 2442-2454.
36. Hu, S., Xiao, C., and Ishikawa, Y. (2018). An efficient algorithm for location-aware query auto-completion. *IEICE Transactions on Information and Systems,* 101(1), 181-192.
37. Qin, J., Xiao, C., Hu, S., Zhang, J., Wang, W., Ishikawa, Y., and Sadakane, K. (2019). Efficient query auto-completion with edit distance-based error tolerance. *The VLDB Journal,* 29(1), 919–943.
38. Jiang, D., Cai, F., and Chen, H. (2018). *Personalizing query auto-completion for multi-session tasks.* Paper presented at the 2018 IEEE International Conference on Computer and Communication Engineering Technology, Beijing, China, 203-207.
39. Mitra, B., Shokouhi, M., Radlinski, F., and Hofmann, K. (2014). *On user interactions with query auto-completion*. Paper presented at the 2014 Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, Queensland, Australia, 1055-1058.

40. Krishnan U., Moffat A., Zobel J., and Billerbeck B. (2020) Generation of synthetic query auto-completion logs. *Advances in Information Retrieval. ECIR 2020. Lecture Notes in Computer Science*, 12035(1), 621-635.
41. Kondrak, G. (2005). *N-gram similarity and distance*. Paper presented at the 2005 International Symposium on String Processing and Information Retrieval, Berlin, Germany, 115-126.
42. Kato, M. P., Sakai, T., and Anaka, K. (2013). When do people use query suggestion? A query suggestion log analysis. *Information Retrieval*, 16(6), 725-746.
43. Baeza-Yates, R., Hurtado, C., and Mendoza, M. (2004). *Query recommendation using query logs in search engines*. Paper presented at the 2004 International Conference on Extending Database Technology, Berlin, Germany, 588-596.
44. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
45. İnternet: Andrej. Karpathy. (2015). *The unreasonable effectiveness of recurrent neural networks*. Andrej Karpathy Blog. Web: http://karpathy.github.io/2015/05/21/rnn-effectiveness/. Son Erişim Tarihi: 28.10.2020.
46. İnternet: Christopher. Olah. (2015). *Understanding LSTM networks*. Colah's blog. Web: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Son Erişim Tarihi: 19.01.2021.
47. İnternet: Jason Brownlee. (2017). *Why One-Hot Encode data in machine learning*. machine learning mastery. Web: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/. Son Erişim Tarihi: 01.01.2021.
48. İnternet: Wikipedia *One-hot Encoding*. Web: https://en.wikipedia.org/wiki/One-hot. Son Erişim Tarihi: 01.01.2021.
49. Craswell, N., Campos, D., Mitra, B., Yilmaz, E., and Billerbeck, B. (2020). ORCAS: 20 million clicked query-document pairs for analyzing search. Paper presented at the 2020 Proceedings of the 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, 2983-2989.
50. İnternet: Nick. Craswell. (2020). *ORCAS: Open resource for click analysis in search*. Msmarco. Web: https://microsoft.github.io/msmarco/ORCAS. Son Erişim Tarihi: 01.01.2021.

51. İnternet: Rohatgi, S., Zhong, W., Zanibbi, R., Wu, J., and Giles, C. L. (2019). *Query auto-completion for math formula search.* Web: https://arxiv.org/pdf/1912.04115v1.pdf. Son Erişim Tarihi: 05.01.2021.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : QURESHI, Abdur Rehman Anwar

Uyruğu : PAKİSTAN

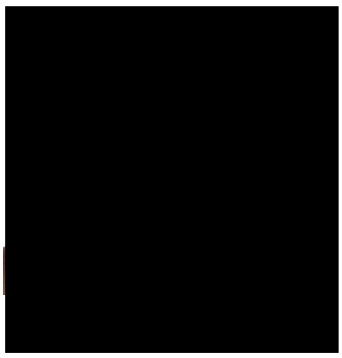
Doğum tarihi ve yeri : [redacted]

Medeni hali : [redacted]

Telefon : [redacted]

Faks : [redacted]

e-mail : [redacted]



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi / Bilgisayar Bilimleri	Devam ediyor
Lisans	Bahria Üniversitesi / Bilgisayar Bilimleri	2018
Lise	PAKİSTAN F.G KOLEJİ / Genel Bilim	2014

İş Deneyimi

Yıl	Yer	Görev
2020	Pivony, İstanbul	Veri Bilimi Stajyeri
2018	Arkitech Software Firm	Yazılım Mühendisi
2017	E-Learning Solutions	Unity 3D Geliştirici
2016	Sapphire Consulting Limited	Oracle Veritabanı Stajyeri

Yabancı Dil

Türkçe, İngilizce

Yayınlar

A.R.A Qureshi, and M.A. Akcayol, "Long Short-Term Memory Based Query Auto-Completion", in 8th International Conference on Electrical and Electronics Engineering, Antalya Turkey, April. 2021.

Hobiler

Kriket ve Futbol

GAZİLİ OLMAK AYRICALIKTIR,…