T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

ARAMA MOTORLARI İÇİN DERİN ÖĞRENME TABANLI SORGU ÖNERİSİ ÇERÇEVESİ

Fatih ÇELİK

DANIŞMAN
Doç. Dr. Sibel SENAN

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği, Tezli Yüksek Lisans Programı

Ocak, 2024

TEZ KABUL VE ONAYI

Fatih ÇELİK tarafından, **Doç. Dr. Sibel SENAN** danışmanlığında hazırlanan **"ARAMA MOTORLARI İÇİN DERİN ÖĞRENME TABANLI SORGU ÖNERİSİ ÇERÇEVESİ"** başlıklı bu çalışma, jürimiz tarafından **18/01/2024** tarihinde yapılan sınav sonucunda **oy birliği** ile başarılı bulunarak **Yüksek Lisans Tezi** olarak kabul edilmiştir.

Tez Jürisi

		İmza	**Sonuç**
DANIŞMAN	Doç. Dr. Sibel SENAN İstanbulÜniversitesi-Cerrahpaşa Bilgisayar Mühendisliği Anabilim Dalı		☐ Kabul ☐ Ret
ÜYE	Doç. Dr. Selçuk SEVGEN İstanbulÜniversitesi-Cerrahpaşa Bilgisayar Mühendisliği Anabilim Dalı		☐ Kabul ☐ Ret
ÜYE	Dr. Öğr. Üyesi Zeynep TURGUT AKGÜN İstanbul Medeniyet Üniversitesi Bilgisayar Donanımı Anabilim Dalı		☐ Kabul ☐ Ret

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve bilimsel etik kuralları içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını ve her türlü hukuki sorumluluğu aldığımı kabul ederim.

Fatih ÇELİK

Aileme ve tüm sevdiklerime ithaf ediyorum…

BÜTÇE DESTEKLERİ

ARAMA MOTORLARI İÇİN DERİN ÖĞRENME TABANLI SORGU ÖNERİSİ ÇERÇEVESİ

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.

TEŞEKKÜR

Yüksek Lisans tez çalışma süresince beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan tez danışmanım değerli Doç. Dr. Sibel SENAN'a teşekkürlerimi sunarım.

Hayatım boyunca her zaman yanımda olan, maddi ve manevi desteklerini hiçbir zaman esirgemeyen aileme ve tüm sevdiklerime teşekkürlerimi, sevgi ve saygılarımı sunarım.

Ocak 2024 Fatih ÇELİK

İÇİNDEKİLER

Sayfa No

TEZ KABUL VE ONAYI ..... **ii**
BEYAN ..... **iii**
İTHAF ..... **iv**
BÜTÇE DESTEKLERİ ..... **v**
TEŞEKKÜR ..... **vi**
İÇİNDEKİLER ..... **vii**
ŞEKİL LİSTESİ ..... **ix**
TABLO LİSTESİ ..... **xi**
KISALTMA LİSTESİ ..... **xii**
ÖZET ..... **xiii**
ABSTRACT ..... **xiv**
1. GİRİŞ **1**
2. KAVRAMSAL ÇERÇEVE **4**
2.1. ÇİZGE TABANLI YAKLAŞIMLAR 6
2.2. ANLAM TABANLI YAKLAŞIMLAR 8
2.3. MAKİNE ÖĞRENMESİ TABANLI YAKLAŞIMLAR 10
3. YÖNTEM **16**
3.1. VERİ KÜMESİ 17
3.1.1. AOL Veri Kümesi 17
3.1.2. İşlenmiş AOL Veri Kümesi 20
3.2. VERİ ÖN İŞLEME 21
3.3. DOĞAL DİL İŞLEME 23
3.4. TEMEL GERÇEKLER 26
3.5. DERİN ÖĞRENME 30
3.6. ÖNERİLEN DERİN ÖĞRENME MİMARİSİ 34
3.7. TQS 37
4. BULGULAR **48**
5. TARTIŞMA **55**

6. SONUÇ VE ÖNERİLER 57
KAYNAKLAR 58
İNTİHAL RAPORU İLK SAYFASI 65
ETİK KURUL İZİN YAZISI 66
KURUM İZNİ YAZILARI 67
ÖZGEÇMİŞ 68

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: Bazı arama motorlarına ait QS örnekleri. 5

Şekil 2.2: Ağırlıklı çizge örneği. 6

Şekil 2.3: Adlandırılmış varlık tanıma örneği. 8

Şekil 2.4: Temel makine öğrenmesi akışı. 11

Şekil 3.1: TQS çerçevesini oluşturan aşamaların diyagramı. 16

Şekil 3.2: En sık kullanılan sorgu cümlecikleri. 19

Şekil 3.3: Her bir kullanıcı için oturumların belirlenme durumu. 20

Şekil 3.4: Veri kümesine ait çeyreklikler ve aykırı değer aralıkları. 23

Şekil 3.5: Bir cümleye ait unigram, bigram ve trigram örnekleri. 25

Şekil 3.6: Örnek bir sorgu cümleciğinin veri ön işleme ve doğal dil işleme süreci. 26

Şekil 3.7: Yapay sinir ağları içerisinde bulunan bir nöron yapısı. 31

Şekil 3.8: Birden fazla gizli katman içeren bir derin öğrenme mimarisi. 32

Şekil 3.9: RNN ve LSTM mimarilerinde kelimeler arası ilişkilerin yakalanması. 35

Şekil 3.10: TA modellerinde kelimeler arası ilişkilerin yakalanması. 35

Şekil 3.11: Derin öğrenmeyi kullanan TA modeli. 36

Şekil 3.12: Kelime dağarcığında bulunan bazı sorgular ve sayısal gösterimleri. 40

Şekil 3.13: Yerleştirme ve konumsal kodlama adımları. 41

Şekil 3.14: TQS'ye ait kodlayıcı katmanı. 43

Şekil 3.15: Öz dikkat mekanizması ve dinamik evrişim mekanizmalarının gösterimi [71]. 44

Şekil 3.16: Kod çözücüye ait maskeleme. 45

Şekil 3.17: TQS'ye ait kod çözücü katmanı. 46

Şekil 3.18: TQS modelinin giriş anından tahmin anına kadar olan tüm adımları. 47

Şekil 4.1: Değerlendirilen temel çizgilere ve TQS'ye ait BLEU puanları. 53

Şekil 4.2: Değerlendirilen temel çizgilere ve TQS'ye ait kafa karışıklığı değerleri..... 53

TABLO LİSTESİ

Sayfa No

Tablo 3.1: AOL sorgu günlüklerine ait sütunlar ve anlamları. 18

Tablo 3.2: AOL sorgu günlüklerine ait çeşitli bilgiler. 19

Tablo 3.3: Bazı sorgular için 3 farklı yöntem ile oluşturulan temel gerçekler. 29

Tablo 4.1: Temel çizgilere ve TQS'ye ait BLEU puanları. 50

Tablo 4.2: Değerlendirilen temel çizgilere ve TQS'ye ait kafa karışıklığı değerleri. 51

Tablo 4.3: TQS modelini çeşitli oturum büyüklüklerinde eğitmek için geçen süreler. 54

Tablo 4.4: Temel çizgiler ve TQS'ye ait bazı QS örnekleri. 54

KISALTMA LİSTESİ

Kısaltmalar	Açıklama
TQS	: Dönüştürücü tabanlı sorgu önerisi (Transformer-based query suggestion)
RNN	: Tekrarlayan sinir ağları (Recurrent neural network)
CNN	: Evrişimli sinir ağları (Convolutional neural network)
QS	: Sorgu önerisi (Query suggestion)
TA	: Dönüştürücü mimarisi (Transformer architecture)
RELU	: Doğrultulmuş lineer birim (Rectified linear unit)
LSTM	: Uzun kısa süreli bellek (Long short-term memory)
GLU	: Geçitli lineer birim (Gated linear unit)
LConv	: Hafif evrişim (Lightweight convolution)

ÖZET

[YÜKSEK LİSANS TEZİ]

[ARAMA MOTORLARI İÇİN DERİN ÖĞRENME TABANLI SORGU ÖNERİSİ ÇERÇEVESİ]

[Fatih ÇELİK]

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği, Tezli Yüksek Lisans Programı

[Danışman: Doç. Dr. Sibel SENAN]

[Çevrimiçi dünyada yer alan bilgi sayısında meydana gelen büyüme ve kullanıcı sayısının artışıyla birlikte bilgiye erişim için arama motorlarının kullanımı giderek yaygınlaşmaya başlamıştır. Bu artış daha fazla kullanıcıyı arama motorlarını kullanmaya teşvik etmiştir. Bu durumla birlikte kullanıcıların sorgu karmaşıklıkları artmaya başlamıştır. Aynı zamanda arama motorlarını kullanıcıların deneyimini iyileştirmek için sorgu önerilerinde bulunmaya ve bu önerileri iyileştirmeye teşvik etmiştir. Günümüze dek sorgu önerisi görevine özgü olarak pek çok farklı yöntem oluşturularak kullanılmıştır. Fakat kullanılan çoğu yöntem, kullanıcıların isteklerini ve aramanın bağlamını tam olarak anlamaktan uzaktır. Hedef sorguların belirlenmesi sorgu önerilerinin doğası gereği oldukça zor bir konudur. Kullanıcı aramalarının altındaki gerçek anlamların yakalanması kullanıcılara doğru ve etkili sorgu önerileri sunmak için oldukça önemlidir. Dönüştürücü mimarileri, çok çeşitli doğal dil işleme görevi için oldukça iyi performans sergilemektedir ve dil ile ilgili problemlerdeki derin anlamları yakalamak için oldukça kullanışlıdır. Dönüştürücüler, sıralı veriler içindeki bağlamsal ilişkileri yakalamak için kişisel dikkat mekanizmasını kullanan derin sinir ağlarıdır. Bu tez çalışmasında, dönüştürücü tabanlı derin öğrenme mimarilerinin gücünden yararlanılarak sorgu önerisi için kullanıcının aramalarının altında yatan anlamlar ve oturumların bağlamı tespit edilmeye çalışılmıştır. Çalışmada sunulan "dönüştürücü tabanlı sorgu önerisi (TQS)" modeli, çeşitli geçmiş sorgu önerisi modelleri ile karşılaştırılmış ve sonuçlar sunulmuştur. Ayrıca elde edilen performans sonuçları, önerilen modelin eğitim aşamasının oldukça hızlı olduğunu göstermektedir.]

Ocak 2024 , [79] sayfa.

Anahtar kelimeler: [Derin Öğrenme, Dönüştürücü, Sorgu Önerisi, Dikkat Mekanizması

ABSTRACT

M.Sc. THESIS

DEEP LEARNING BASED QUERY SUGGESTION FRAMEWORK FOR SEARCH ENGINES

Fatih ÇELIK

İstanbul University-Cerrahpaşa

Institute of Graduate Studies

Department of Computer Engineering

Computer Engineering, Thesis Master Program

Supervisor: Assoc. Prof. Dr. Sibel SENAN

With the growth in the number of information in the online world and the increase in the number of users, the use of search engines to access information has become increasingly widespread. This increase has encouraged more users to use search engines. With this situation, the query complexity of users has started to increase. It has also encouraged search engines to make and improve query suggestions to improve users' experience. So far, many different methods have been used for the query suggestion task. However, most of the methods used are far from fully understanding users' wishes and the context of the search. Determining target queries is a very difficult issue due to the nature of query suggestions. Capturing the real meanings of user searches is very important to provide accurate and effective query suggestions to users. Transformer architectures perform very well for a wide range of natural language processing tasks and are useful for capturing deep meanings in language-related problems. Transformers are deep neural networks that use the self-attention mechanism to capture contextual relationships within sequential data. In this thesis study, we tried to determine the underlying meanings of the user's searches and the context of the sessions for query suggestion by utilizing the power of transformer-based deep learning architectures. The "transformer-based query suggestion (TQS)" model presented in the study was compared with various historical query suggestion models and analyzes were presented. In addition, the performance results obtained show that the training phase of the proposed model is quite fast.

January 2024, 79 pages.

Keywords: Deep Learning, Transformer, Query Suggestion, Attention Mechanism

1. GİRİŞ

Her gün milyonlarca kullanıcı çevrimiçi dünyada bilgiye ulaşmak için arama motorlarını kullanmaktadır. Artan akıllı cihaz kullanımı ile dijital ortamlarda, arama motorları kullanımı yaygınlaşmıştır. Bu yüzden arama motorlarının bazı zorlu görevleri de yerine getirme zorunluluğu doğmuştur. Bu görevlerden biri, kullanıcılara sorguda ifade ettikleriyle ilgili en alakalı bilgileri sunmaktır. Kullanıcılar, arama çubuklarına ulaşmak istedikleri bilgi ile ilgili birkaç kelime girerek çevrimiçi dünyadaki keşif ve bilgiye ulaşım deneyimini yaşamaktadır. Ancak burada kullanıcıların talep ettikleri bilgileri arama motorlarına ne şekilde ifade etmeleri gerektiği ile ilgili bir zorluk ortaya çıkmaktadır.

Arama motorları, günümüzde çevrimiçi dünyanın genişleyen bilgi ağında kullanıcıların istediklerini bulabilmeleri için bir geçit görevi üstlenmektedir. Ayrıca kullanıcılara zaman ve hız açısından verimlilik sunarak çevrimiçi dünyadaki gezintilerine yardımcı olmaktadır. Sunulan arama motorları, kullanıcıların çevrimiçi dünya deneyimini hızlandırırken isteklerini anlamak ve karşılamak konusunda da oldukça verimli olmalıdır. Bu durum ile ilgili belki de en önemli problem kullanıcıların sorgularını bir makinenin anlamasını sağlamak ve sorgunun arkasındaki anlamları yakalamaktır. Arama motorları için bu problem, kullanıcı deneyimini ve verimliliğini doğrudan tanımladığı için önemi yadsınamaz bir gerçektir. Bu problemi çözmek için sorgu önerileri (QS) arama motorlarında yaygın olarak kullanılmaktadır. QS, aynı zamanda arama motorlarının performanslarını optimize etmede önemli bir yer tutar [1]. Bu çalışma, arama motorlarındaki bu önemli görevi yerine getirecek kaliteli bir yöntem sunmaya odaklanmaktadır. Bunun altında yatan neden ise gayet açıktır: mevcut sorgular ışığında kullanıcıların arama deneyimlerini yeni bir boyuta taşımak. Bu hedef, aynı zamanda arama motorlarının daha kullanıcı dostu ve vazgeçilmez hale gelmesinde önemli rol oynamaktadır.

Kullanıcıların, aradığı bilgiyi ifade etmekte yaşadığı güçlükleri ve makinelerin bu ifadelerinin arkasında yatan gerçek anlamları bulmaktaki zorluklar literatürde incelenmiştir [2]. Yaşanan bu zorlukların üstesinden gelmek ve bilgiye erişim hızının oldukça önemli olduğu bu dönemde arama motorlarının QS alanındaki başarısını artırarak kullanıcı deneyimini her açıdan

iyileştirmek için tasarlanan “dönüştürücü tabanlı sorgu önerisi (TQS)” çerçevesinin ortaya çıkmasını sağlamıştır.

Arama motorları için kullanıcıların dijital dünyadaki serüveni bir arama sorgusuyla başlar. Ancak bu birkaç tuş vuruşundan ibaret basit bir süreç değildir. Tuş vuruşlarından doğru bilgiye giden yolu belirlemek ve ilerlemek makineler için oldukça zordur. Bu zorluklarla dolu süreçte makinelerin, kullanıcı isteklerini anlamalarının en başarılı yollarından biri derin öğrenme yöntemleridir. TQS, tüm bu yolculukta kullanıcıya sunulan bilginin kalitesini ve verimliliğini artırmak için derin öğrenme yöntemlerinin gücüne güvenerek tasarlanmış bir araçtır.

Çevrimiçi aramalarda önemli bir zorluk hala devam etmektedir. Arama motorları, kullanıcı sorgularının ardındaki anlam ve amacı nasıl anlayabilir ve bunlara uygun yanıtları nasıl verebilir? Geleneksel yöntemler, sorguların arkasında yatan bağlam ve anlamları yakalamada yetersiz kalmaktadır. Bu durum, kullanıcının arama sonuçlarının beklentilerinden uzaklaşmasına ve belki de ilgili arama motorunu kullanmamasına yol açabilir. Bu problemi ortadan kaldırarak kullanıcılara daha kaliteli ve verimli sorgu önerilerinde bulunan TQS, QS için metinlerde bağlamı ve anlamsal anlamları yakalamakta başarısı kanıtlanan derin öğrenme tabanlı dönüştürücü mimarilerini (TA) kullanmaktadır [2, 3]. TQS, önerilerin başarısını artırmak için QS alanında popüler bir bölüm olan kullanıcı oturumlarını ve kullanıcı benzerliklerini kullanmaktadır [4]. Çevrimiçi aramaların gelişen ve büyüyen doğası gereği verimlilik ve alaka düzeyi en önemli konu olmaya devam etmektedir. Bu çalışma, bu alanda son teknoloji TQS çerçevesini sunmaktadır. Bu kapsamda QS için iki ana amaç üzerinde yoğunlaşılmıştır. Verimliliği yükseltmek ve kullanıcı deneyimini iyileştirmek için doğruluğu artırmak. Çeşitli derin öğrenme [5, 6] ve TA modelleri araştırılmıştır. Belirtilen iki ana amaca uygun olarak, metinlerin bağlamlarını ve anlamsal anlamlarını yakalamakta tekrarlayan sinir ağları (RNN) ve evrişimli sinir ağlarına göre (CNN) daha başarılı olduğunu kanıtlanan dönüştürücü mimarisi (TA) modellerinden yararlanılmıştır. Çalışma TA modellerinde, bazı hız ve performans iyileştirmelerini sunmaktadır.

Bu tez çalışmasının amacı, arama motorları için QS problemini yeniden şekillendiren *Dönüştürücü Tabanlı Sorgu Öneri Çerçevesi* sunulmasıdır. QS alanındaki karmaşıklıklar incelenirken, sorguların altında yatan bağlamlar, sorguların anlamsal anlamları, sorguların benzerlikleri, kullanıcıların oturumları ve kullanıcılar arasındaki benzerlikler göz önünde

bulundurulmuştur. Doğal dil işleme, sorgu genişletme, metin benzerliği, çizge tabanlı benzerlikler, derin öğrenme teknikleri ve TA modelleri tez çalışmasında önerilen çerçevenin temellerini oluşturmaktadır. Bu sayede, özgün bir dönüştürücü tabanlı sorgu önerisi çerçevesi sunulmuştur.

TQS'nin ortaya çıkışı ve yapısı oldukça önemlidir. Tez çalışması kapsamında kullanılan veri kümesi, TA modelleri, kullanıcı oturumlarının belirlenmesi, önerilen modele ait bilgiler ve değerlendirme metrikleri ileriki bölümlerde sunulmaktadır. Ayrıca, modelin en iyi performansa ulaşması için kullanılan parametrelerden, veriyi modele uygun hale getirmek için uygulanan yöntemlerden ve sorguların benzerliklerini yakalamak için kullanılan tekniklerden detaylı olarak bahsedilecektir.

Her problemde olduğu gibi QS alanında da bazı sınırlamalar mevcuttur. Bu sınırlamalardan bazıları tez çalışmasında önerilen çerçeve içinde geçerliliğini korumaktadır. Çalışma, arama motorları için büyük bir öneme sahiptir ve sonuçlar ümit vericidir. Ayrıca gelişime açıktır fakat QS görevinin doğası gereği bazı sınırlarının olması kaçınılmazdır. Çalışma ile derin öğrenme tabanlı QS yöntemlerine bir yenisi eklenmekte ve TA modellerinin QS alanında başarısını ispatlamaktadır.

2. KAVRAMSAL ÇERÇEVE

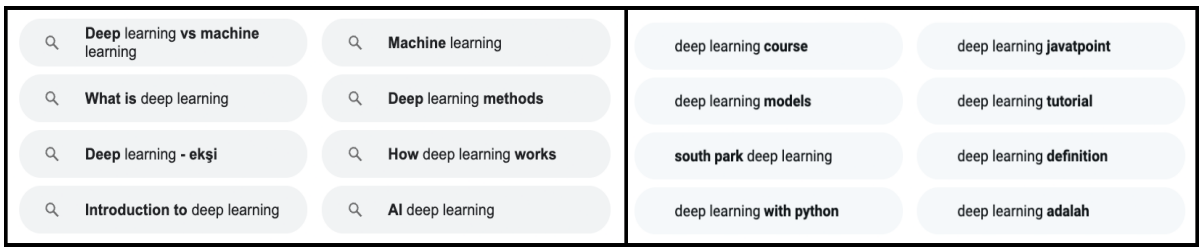
Dijital dünyada meydana gelen önlenemez gelişim dijital bilginin ciddi ölçüde genişlemesine neden olmuştur. Meydana gelen bu büyümeyle kullanıcılar, bilgiye daha hızlı erişmek için arama motorlarını tercih etmeye başlamıştır. Bu durum, araştırmacıları arama motorlarını kullanıcılara daha doğru bilgiler sunmak için çalışmalara yönlendirmiştir. Bu nedenle QS, arama motorları için kullanıcıların aradıklarını verimli ve doğru olarak sunabilmek adına büyük önem taşımaktadır. Bu bölümde, QS alanına genel bir giriş yapılarak, QS görevinden, QS için kullanılan teknik ve yöntemlerden bahsedilecektir. Ardından literatürde QS ile ilgili yapılan çalışmalardan bahsedilecektir. Geçmişten bugüne dek QS alanında; kural tabanlı, çizge tabanlı, anlam tabanlı ve son yıllarda yapay zekânın da gelişimiyle birlikte makine öğrenmesi ile derin öğrenme tabanlı yaklaşımlar kullanılmıştır. Burada belirtilen teknikler ve yöntemler tez çalışmasında önerilen sorgı önerisi çerçevesinin temelini oluşturmaktadır.

QS ya da diğer bir adıyla sorgu önerileri, günümüzde arama motorlarında yaygın olarak kullanılan bir kavramdır. Kullanıcılara, arama konusunda daha kaliteli bir deneyim sunmayı amaçlar. Google[1], Yahoo[2] ve Bing[3] gibi birçok arama motoru, QS dizilerini kullanarak kullanıcılara aradıklarını daha kesin olarak tamamlama ve sonuca ulaşma şansı tanımaktadır. QS görevinin ana amacı, kullanıcıya aradığı bilgiyi kesin ve hızlı olarak sunmaktır.

Arama motorları, internetin gelişimi bile birlikte kullanıcılar tarafından oldukça sık olarak kullanılmaya başlanmıştır. Kullanıcılar için, aradıklarını bir makineye anlatmak zor olabilir. Örneğin kullanıcının tam olarak ne aradığını bilmediği durumlarda bunu arama motoruna aktararak istediği sonuca ulaşması oldukça güçtür. Arama motorları, bu yüzden QS özelliklerini kullanır. Arama motorları, QS kullanarak kullanıcının aramasının altında yatan anlamı yakalar ve kullanıcıya aramak istediği kelimeyi ya da cümleciği önermeye çalışır. Bu sayede kullanıcıların aradığı şeyin ne olduğunu tam olarak bilmesine gerek kalmadan aradığı şeyi en doğru şekilde kullanıcıya gösterebilir. Bu arama motorları için kullanıcı kazanımı

[1] https://www.google.com/
[2] https://search.yahoo.com/
[3] https://www.bing.com/

açısından önemli bir gelişim alanıdır. Farklı arama motorlarına ait QS örnekleri Şekil 2.1'de gösterilmiştir.



Şekil 2.1: Bazı arama motorlarına ait QS örnekleri.

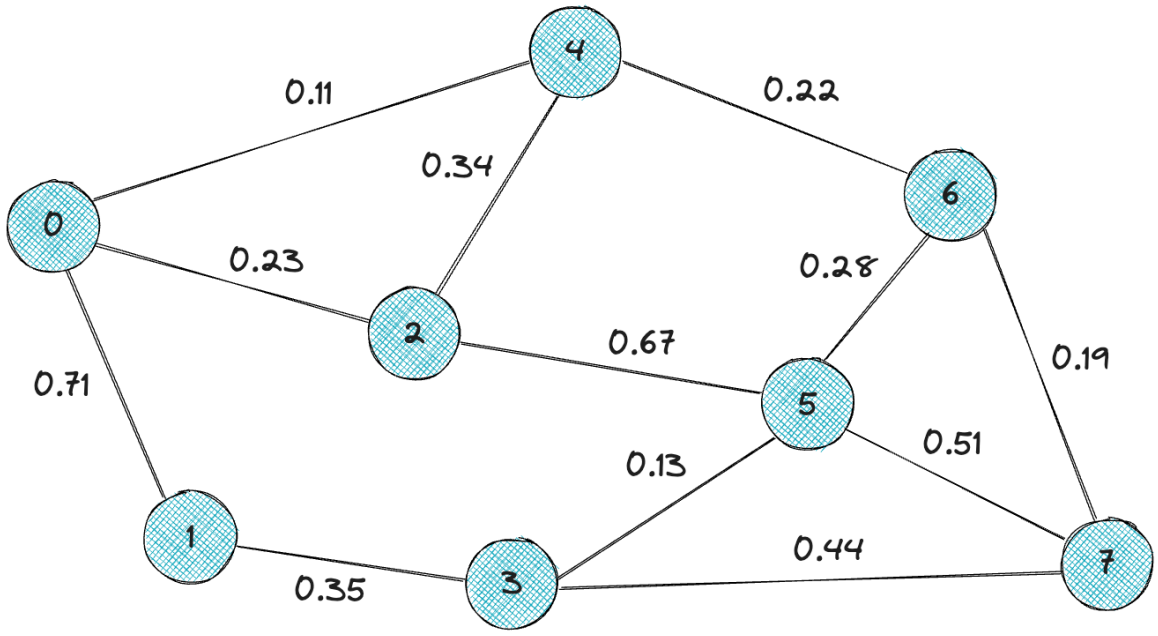
Son yıllarda, birçok farklı alanda yapay zekâ kullanılmaya başlanmıştır. Makine öğrenmesi ve derin öğrenme tabanlı yöntemler bu alanlarda önemli gelişmelere katkıda bulunmuştur. Bu yaklaşımlar, büyük miktarlarda veri kullanarak bu verilerin anlamlandırılmasını ve bu anlamlı bilgilerin kullanılmasını içerir. Arama motoru görevlerinin ve problemlerinin özelinde metin verileri büyük önem taşır. Metin verileri, doğal dil işleme ve makine çevirisi gibi alt görevler ile kullanılır. QS de bu iki alt görev kullanılarak iyileştirilebilir. 2006 yılında AOL[4] tarafından yayımlanan sorgu günlükleri özellikle QS alanındaki çeşitli çalışmalarda kullanılmıştır. QS için geçmişte kural tabanlı sistemler kullanılmıştır. Yapay zekanın gelişimi ve endüstride kullanılmaya başlanması, arama görevlerinde kullanımını da beraberinde getirmiştir.

Doğal dil işleme ve metin verilerini kullanan diğer alt görevlerde farklı alanlarda birçok çalışma yapılmıştır. Doğal dil işleme görevlerinde, yinelemeli sinir ağları (RNN) ve evrişimli sinir ağları (CNN) gibi yaklaşımlar yaygın olarak kullanılmaktadır. Fakat bu yöntemlerde bazı doğal problemler bulunur. RNN ve CNN, yapıları gereği ardışık kelime dizilerinde o an ele alınan kelimeyi sadece bir önceki kelime ile değerlendirebilir. Bu yüzden ardışık kelime dizilerinde, değerlendirilen kelime ile diğer kelimeler arasındaki mesafe arttığı için kelimelerin birbiriyle ilişkisini yakalamak zorlaşır. 2017 yılında doğal dil işleme alanında devrim olarak nitelendirilen bir yöntem ortaya çıkmıştır [7]. Çalışmada sunulan yöntem, kelime dizilerinin içerisindeki derin anlamları ve kelimelerinin birbirleri ile olan ilişkileri anlayarak öğrenebilmektedir. Bu da RNN ve CNN gibi yöntemlere göre daha başarılı bir yöntem olduğunu kanıtlamıştır. Bu yöntem, sadece doğal dil işleme görevleri değil, makine çevirisi, büyük dil modelleri ve görüntü işleme gibi bazı alanlarda da kullanılmaya başlamıştır.

[4] https://www.aol.com/

2.1. ÇİZGE TABANLI YAKLAŞIMLAR

Çizge, köşe ve kenarlardan oluşan, bilgisayar bilimlerinde oldukça yaygın kullanıma sahip matematiksel kavramdır. Pek çok farklı çizge çeşidi vardır. Yönlü, yönsüz ve ağırlıklı gibi çeşitli çizge türleri bulunur ve her biri farklı kullanım alanlarına sahip olabilir. Bir çizge, basitçe kenarlardan ve düğümlerden oluşur. Düğümler sonlu veya sonsuz değerler alabilir. Bir kenar iki düğüm arasındaki bağlantıyı sağlar. Basit bir ağırlıklı çizge Şekil 2.2'de verilmiştir.



Şekil 2.2: Ağırlıklı çizge örneği.

Bilgi çizgeleri, varlıklar, varlıklara ait özellikler ve aralarındaki ilişkileri saklayan bir tür çizgedir. Varlıklar düğümler ile aralarındaki ilişkiler ve özellikler ise kenarlar ile ifade edilir. Bilginin yapısal bir temsilidir. Büyük miktarda veriyi makinelerin anlayabileceği bir formata dönüştürür. Günümüzde kullanılabilir durumda farklı bilgi çizgesi tabanları mevcuttur. Bu çizgeler, QS ve makine anlama gibi görevleri yerine getirmek içinde kullanılmaktadır.

QS alanında, sorguların birbiri ile ilişkisini anlamak ve bağlamsallığı geliştirmek için bilgi çizgeleri kullanılmaktadır. Kullanıcı sorguları çizgedeki bir düğüm ile temsil edilirken, çizge üzerindeki başka bir varlığa bağlanarak belirsizlikler giderilebilir. Ayrıca çeşitli bilgiler çizge üzerinde ifade edilerek kişiselleştirme gibi görevler yerine getirilebilir. Bu tür yaklaşımlar, genellikle kullanıcılara alakalı önerilerde bulunmak için çizgelerden yararlanır.

TaSQS'ın önerildiği çalışmada, belirli bir zamandaki popülariteye göre sorgu ve tıklama verilerini kullanarak bir çizge oluşturulmuştur. Bu çizgeyi kullanarak QS, bir zaman çizelgesinde gösterilmiştir. Geçmiş tarihlerdeki önerilere göre bu QS sonuçları, *Hierarchical Agglomerartive Clustering (HAC)* algoritması kullanılarak kümelenmiştir. Google aramaları da dahil çeşitli basit referans değerlerinde, kıyaslanan diğer çalışmalara göre daha iyi sonuçlar elde edildiği belirtilmiştir [8].

Web aramalarında çeşitlendirme ve kişiselleştirmeyi birleştiren QS-DP'de, QS için çok parçalı çizgeler kullanılmıştır. Bu çizgeleri kullanarak QS için çeşitlendirmeler yapılmıştır. Çalışmanın kişiselleştirme bölümü için, *User Profiling Model (UPM)* kullanılmıştır. Bu iki modeli birleştirerek tek bir modelde sunan QS-DP, ticari bir arama motoruna ait verilerle yapılan testlerde birlikteliğin birbirini geliştirebileceğini göstermiştir [9].

Bir diğer çalışmada, keşif amaçlı aramayı desteklemek için hedef kaydırma incelenmiş ve bunu desteklemek için hedef kaydırmaya dayalı bir öneri sistemi önerilmiştir. Önerilen sistem; hedef kaydırma çizgesi oluşturarak, oluşturulan bu çizgede *Random Walk Algorithm (RWA)* ile QS dizileri oluşturulmuştur. Diğer yöntemlerle yapılan karşılaştırmalı analizler önerilen yöntemin etkinliğini kanıtlamıştır [10].

İki popüler QS yaklaşımı olan QFG ve TQGraph'ı genişleten bir başka çalışma, arama sorgusu ile kullanıcının bulunduğu konum arasındaki uzamsal yakınlığı ölçerek kullanıcıya konuma uygun QS sunmaya çalışmıştır. Ayrıca *Bookmark-Coloring Algorithm (BCA)*'yı da genişleterek yararlanmıştır. Yapılan diğer benzer çalışmalara göre daha iyi sonuçlar verdiğini belirtilmiştir [11].

QS için varlık tabanlı melez bir yaklaşım sunan KB-QRec ve D-QRec adındaki iki çizge yöntemini uygulayan çalışma, giriş sorgularından varlıkları çıkarmakta ve yenilerini keşfetmek için bu varlıkları kullanmaktadır. Elde edilen sonuçlar uzun kuyruklu sorgularda başarısını ve buna ek olarak hızın önemli olduğu durumlar için 100 milisaniyenin altında QS sunabildiği için kullanılabilirliğini de kanıtlamıştır [12].

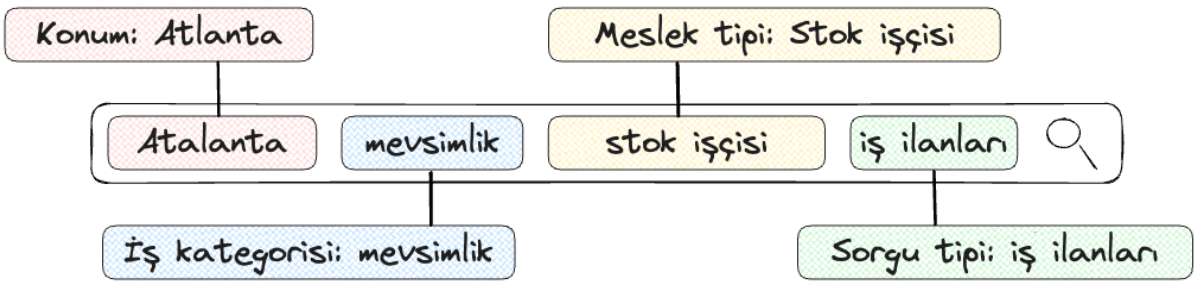
QCG, kullanıcı sorgularını ve bu sorgulara ait benzerlikleri içeren çizgedir. QFG ise kullanıcıların oturum bilgilerini ve ardışık aramalarından oluşan bilgileri içeren çizgedir. Yapılan çalışmada belirtilen bu iki çizge birleştirilerek tek bir melez çizge sunulmuştur. Bu çizge kullanılarak QS üretilmektedir. Ayrıca QCG içerisindeki sorgularda, aynı özellikleri

taşıyan sorguları azaltarak düzelten *Locality-Sensitive Hashing (LSH)* yöntemi kullanılmıştır. Üç katmanlı bir mimariden oluşan bu yöntem, QFG'yi kullanarak, kullanıcıların geçmiş sorgularındaki anlamsal benzerlikleri incelemektedir. İki farklı veri kümesi ile yapılan testlerin, diğer benzer yöntemlere göre 14% daha başarılı olduğu belirtilmiştir [13].

Çeşitlendirme ve kişiselleştirme kavramları genellikle zıt kavramlar olarak değerlendirilmektedir. Çalışmada, bu zıt görüş ortadan kaldırılarak, çeşitlendirme ile kişiselleştirmeyi tek bir birleşik çerçevede sunan çeşitlilik farkında kişiselleştirilmiş bir QS modeli önerilmiştir. Model, arama sorgularının farklı potansiyel yönlerini kapsayarak etkili bir şekilde çeşitlendirilirken, her bir kullanıcı için QS dizilerinin sıralamasını kişiselleştirmek için *User Profiling Model (UPM)* kullanmaktadır. Sonuçlar, çeşitlendirme ve kişiselleştirmenin etkili bir şekilde entegre edilebileceğini göstermiştir [14].

2.2. ANLAM TABANLI YAKLAŞIMLAR

Metin işleme alanında anlamsal yaklaşımlar, metinlerin yapısal özelliklerine değil anlamını anlamaya odaklanır. Bu tür yaklaşımlar, doğal dile ait kelimeler, cümleler ve metinlerin bağlamlarını ve arkasındaki derin anlamları yakalamaya odaklanır. QS alanında da aynı şekilde sorguların altında yatan derin anlamları yakalayarak QS dizilerini sunmak için bu yöntem kullanılmaktadır. Anlam tabanlı yaklaşımlar ile QS sunulurken; adlandırılmış varlık tanıma, sorgu genişletme, *Kelime yerleştirme (Word embedding)*, bağlam yerleştirme ve bilgi çizgeleri gibi çok çeşitli yöntemler kullanılır. Anlam tabanlı yöntemler içerisinde oldukça sık kullanılan adlandırılmış varlık tanıma ile ilgili bir örnek Şekil 2.3'te verilmiştir. Bu yaklaşımlar, sorgu metinlerindeki anlamları, ilişkileri ve bağlama dikkat ederek QS kalitesini artırmaya ve kullanıcı deneyimini artırmaya yardımcı olur.



Şekil 2.3: Adlandırılmış varlık tanıma örneği.

X (önceki adı ile Twitter) ile farklı birkaç veri kaynağından gelen verileri kullanan ve QS için sosyal medyanın dikkate alınması gerektiğini belirten sosyal ağlara dayalı bir öneri sistemi sunulmuştur. Kullanıcıya, aramalarına göre X'te yüksek erişime ulaşmış varlıklardan (Etiket, bahsetme, web adresi vb.) oluşan dinamik bir öneri kümesi sunulmuştur. X'teki yüksek erişime ulaşmış ve güncel varlıkların önerilmesiyle, içerikler hızlıca yayılmış, yeniden kullanılması sağlanmış ve gündemde kalması sağlanarak dinamik bir ağ oluşturulmuştur. Değerlendirmeler, önerilerin doğruluğunu ve eğilimlerin ne kadar hızlı yansıtıldığını araştırmıştır [15].

Arama kelimelerinin gizli anlamlarından yararlanan QS modeli, tam kelime eşleştirmeye dayalı modellerin karmaşıklığın üstesinden gelmeye odaklanarak bu sınırın ortadan kaldırılmasını sağlamıştır. Bunu yapmak için *Latent Dirichlet Allocation (LDA)* kullanılmıştır. Bu strateji, modelin kullanıcıların sorgularının altındaki anlamı yakalamasına yardımcı olmuştur. Önerilen model hem kelime hem de konu tabanlı sistemlerin güçlü yönlerinden yararlanırken sunulan sonuçlar, sorgu günlüklerindeki gizli anlamları kullanmanın QS kabiliyetini artırdığını göstermektedir [16].

Single Objective Genetic Approach (SGA) tabanlı kişiselleştirilmiş öneri sisteminde, önerilen sorgu kümesindeki görünüm benzerliğini ve gizli anlam alaka düzeyini kontrol eden iki ayrı hedef belirlenmiştir. Bu iki amacın ağırlıklı toplamı ile nihai hedefi oluşturulmuştur. Hedef ağırlıkları, kişisel tercihlere göre ayarlanmıştır. Sonuçlar, sunulan yöntemin üretilmesi çok zor gizli anlamsal bağlarla QS alanında bulunabileceğini göstermiştir. Ayrıca ağırlıkların periyodik olarak değiştirilmesi gerektiği bildirilmiştir [17].

Geri bildirim oturumları kullanıcıların arama sorgularına sunulan öneriler için yardımcı kaynak olarak kullanılabilmektedir. Çalışmada, geri bildirim oturumunda tıklanan ve tıklanmayan belgelerde sunulan anahtar kelimeleri belirleyen ilgili arama önerisi çerçevesi sunulmuştur. Geri bildirim oturumları, zenginleştirilmiş belgelere çevrilmiştir. Tüm bu çevrilmiş belgeler, sözde belgeleri oluşturmak için kullanılmıştır. Bu sözde belgeler birleştirilerek optimize edilmiş ve optimize edilmiş sözde belgeler oluşturulmuştur. Optimize edilmiş belgelerde, çeşitli benzerlik yöntemleri kullanılarak QS dizileri oluşturulmuştur. Model, çeşitli yöntemlerle kıyaslanmış, performansı değerlendirmek *için Click-Through Rate (CTR)* kullanılmıştır. Önerilen model değerlendirilen diğer QS modellerine göre daha yüksek *CTR* sağlamıştır [18].

Bilgi tabanları (Knowledge-Base), gerçek dünya varlıklarının birbirine nasıl bağlandığını açıklayan zengin kaynaklardır. Uzun kuyruklu sorgular için sunulan çoğu QS sistemi yetersizdir. Bu yetersizliğin üstesinden gelmek adına uzun kuyruklu sorguların işlenmesi için *YAGO* ve *Freebase* gibi bilgi tabanlarından yararlanan bir model önerilmiştir. Model, uzun kuyruklu sorgulardan varlıkları çıkararak bunu bilgi tabanında yenilerini keşfetmek için kullanmıştır. Önerilen modelin, uzun kuyruklu sorgular için mevcut çözümlerden daha iyi sonuç verdiği belirtilmiştir [19].

Çoğu zaman kullanıcıların ilk aramaları, bir bilgi tabanındaki varlıklara benzersiz olarak bağlanabilir. Gerçek dünya bilgi tabanlarında bu bağlantıya sahip varlıkların sayısı çok fazla olabilir ve ilgili varlıklar eşit derecede ilgili olmayabilir. Bu varlıklar sıralanmalıdır. Yapılan çalışmada, Spark olarak isimlendirilen anlamsal arama yardım aracı tanıtılmıştır. Bu araç, kullanıcının ilk sorgusunu bilgi tabanındaki bir varlığa bağlayan ve ilgili varlıkların bir sıralamasını sağlayan öneri motoru olarak açıklanmıştır. Araç, çeşitli kaynaklardan sinyaller çıkarak, bu sinyalleri kullanıcı QS dizileri oluşturmak için kullanmıştır [20].

Web arama sorgularındaki bağlam ile ilgili sorunlardan biri, ciddi zaman kısıtlamaları altında çalışmasıdır. Çalışmada önerilen yöntem, dinamik programlama ve anlamsal vektörleri kullanmıştır. Anlamsal vektörler, QS dizilerinin sıralanışına sorgu bağlamı ile ilgili bilgi eklemek için kelime dağılım anlamını kodlar. Model, bu iki yöntemi kullanan bir olasılık yöntemini sunmuştur. Algoritmayı hızlı ve tasarruflu yapan üç bileşen vardır. Bunlar; bağlama işleminde yok sayma, belge ayak izi azaltma için sıkıştırma, sorgu kelimeleri ve varlık anlamları arasındaki mesafeyi hesaplamadır. Sonuçlar, önerilen sistemin mevcut sistemlerden iki kat hızlı işleme yapabildiğini göstermektedir [21].

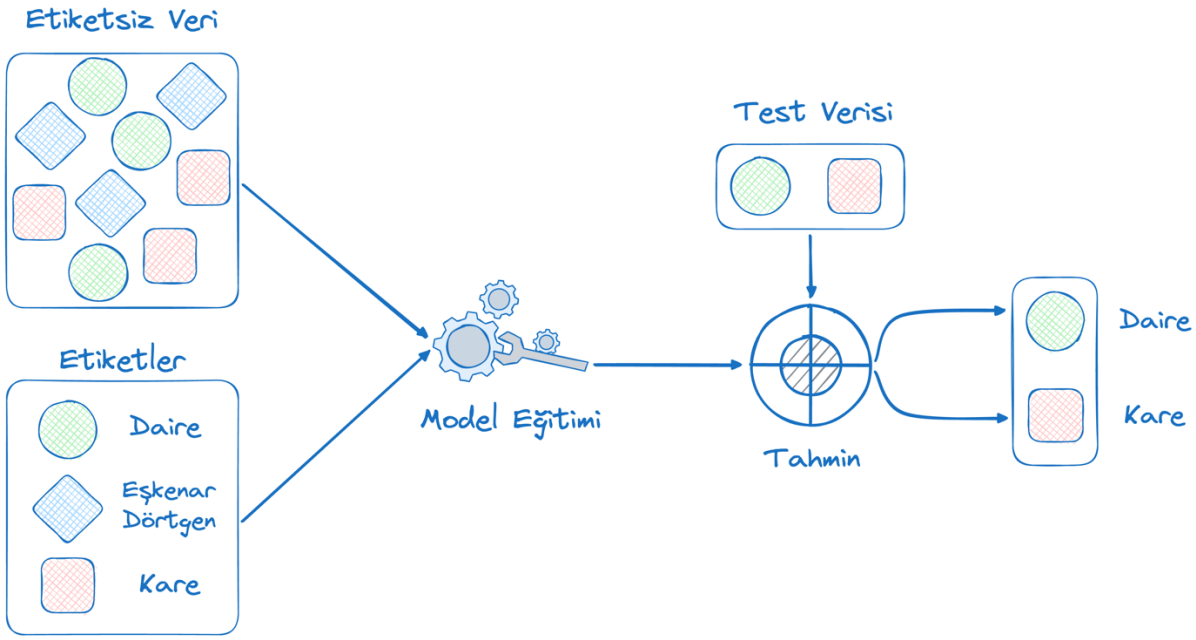
2.3. MAKİNE ÖĞRENMESİ TABANLI YAKLAŞIMLAR

Makine öğrenmesi, makinelerin verilere dayalı olarak öğrenmesini ve tahminde bulunmasını veya karar almasını sağlayan algoritmaları ya da istatistik modellerini geliştirmeyi sağlayan bir yapay zekâ alt alanıdır. Makine öğrenmesini temel düzeyde açıklayan görsel Şekil 2.4'te gösterilmiştir. Mevcut sistemlerin içerisindeki geçmişe yönelik veriler ya da farklı bilgi kaynakları kullanılarak karar alma ve geleceğe dair öngörülerde bulunmak için kullanılır. Bunu yaparken geçmiş bilgi kaynaklarının birbirleri ile olan ilişkilerini, bilgi kaynaklarındaki kalıpları ve örüntüleri kullanır. Birçok sektördeki karmaşık problemlerin çözümünde makine

öğrenmesinden destek alınmaktadır. Bu problemlerin çözümünde gösterdiği başarılar nedeniyle günümüzde yaygın bir kullanım alanına sahiptir.

QS alanında, kullanıcı girdilerine ve geçmiş arama verilerine bakarak gelecek sorgu tahminlerini yapmak için kullanılır. Makine öğrenmesi, kullanıcıların geçmiş verilerini inceleyerek sorgularındaki anlamlarını, sorgularının amaçlarını ve kişiye özgü durumlarını tespit etmek için oldukça ideal bir yöntemdir. Makine öğrenmesi, QS alanında, kullanıcılara ait sorguların kalitesini ve kullanıcı ile etkileşim düzeyini artırmayı ve kullanıcı deneyimini iyileştirmeyi hedefler. Günümüzde pek çok makine öğrenmesi yöntemi QS alanında kullanılmaktadır. Doğal dil işleme, derin öğrenme ve son zamanlarda yaygınlaşmaya başlayan büyük dil modeller QS alanında birçok çalışmada kullanılmıştır.

Derin öğrenme tabanlı yaklaşımlar da QS için kullanılmaktadır. Derin öğrenme yöntemleri, QS oluşturmak için kullanıcı davranışlarını, sorguların anlamsal anlamlarını, bağlamları ve kullanıcıların birbirlerine olan benzerliklerini yakalar. QS için kullanılan bazı derin öğrenme yöntemleri içerisinde RNN ve CNN ile *BERT* ve *BART* gibi dönüştürücü mimari (TA) modelleri sayılabilir. Bu yönteme dayalı çalışmalar, bahsedilen diğer yaklaşımlara göre daha doğru ve kişiselleştirilmiş öneriler sunma avantajına sahiptir.



Şekil 2.4: Temel makine öğrenmesi akışı.

Derin öğrenme tabanlı sorgu önerileri çalışmaları şu şekilde verilebilir: Homomorfik şifrelemeden esinlenen *Homomorfik Yerleştirme (Homomorphic Embeddings)*, anlamsal terim

yerleştirmelerine dayanan hesaplamalarla sorguyu yeniden ifade etmek için kullanılır. Yeniden ifade edilen sorguları ve yerleştirmeleri kullanarak sorguların yeniden ifade edilişini çıkarmak için *Dikkat Mekanizmasına (Attention Mechanism)* sahip bir RNN mimarisi kullanılmıştır [22]. Mimari, oturum bağlamının yapısını yakalamaya çalışmıştır. Ayrıca mimarinin hem sorguları yeniden ifade etmek hem de QS sunmak için optimize edilebileceği gösterilmiştir. Ortaya çıkan sonuçlar, önerilen RNN mimarisisin rekabetçi diğer temel çizgilere göre daha iyi sonuçlar verdiğini göstermiştir [22].

QS için kullanıcı etkileşimlerini modelleyen geri bildirime dayalı bellek ağları, kullanıcıya ait tıklanan ve tıklanmayan (atlanan) belgeleri kullanıcının tercihlerini temsil için olumlu ve olumsuz geri bildirim olarak kullanmıştır. Bu yaklaşım kullanılarak sorguların altında yatan bilgi ihtiyaçları daha iyi modellenmiştir. Önerilen model, *Hiyerarşik Diziden-diziye Sinir Ağı Sorgu Öneri Modeli (Hierarchical Sequence-to-Sequence Neural Query Suggestion Model)* ile bütünleştirilmiştir. Sonuçlar, önerilen yöntemin daha zor şartlarda daha dikkat çekici sonuçlar verdiğini göstermiştir [23].

Tek bir kullanıcı oturumu içerisinde, daha önce aranan sorgulardaki terimlerin büyük bir kısmının yeri korunur. QS için oturum bazlı özelleştirilmiş *Diziden-diziye (Sequence-to-sequence)* model, oturumlardaki anlamı ve bağlantıyı yakalamak için bir dikkat mekanizmasını kullanmıştır. Bu yöntemin oturumun sınırlarını da belirlemekte fayda sağladığı belirtilmiştir. Çalışmada ayrıca, kullanıcı oturumundaki sorguların büyük bir çoğunluğunun kendisinden önceki sorgu cümleciklerinden bağımsız olduğu bildirilmiştir. Bundan dolayı önerilen modelin *Kod Çözücüsüne (Decoder)* bir dikkat mekanizması dahil edilmiştir. Bu dikkat mekanizması, kullanıcının tahmin edilen sorgu öncesindeki sorgularına tam erişim sağlayarak kullanıcının oturumundaki anlamları tam olarak anlamasına ve oturum farkında olmasına yardımcı olmuştur. Ek olarak çalışma, bir dizi değerlendirme metriğini QS görevine özel olarak önermiştir. Önerilen model ile yapılan değerlendirmeler, QS için Diziden-diziye modellerin performansı artırdığını ve çeşitli değerlendirme metriklerinde daha iyi performans gösterdiğini ortaya koymuştur [24].

Veri seyrekliği nedeniyle sorguların bağlamını tespit etmek zordur. Çalışma, hiyerarşik bir sinir ağı mimarisi kullanılarak üretken, bağlam farkında ve nadir sorguları işleyebilen QS sistemi sunmuştur. Önerilen sistem, nadir ve uzun kuyruklu sorgulardan bağlamları çıkararak QS sunmaktadır. Değerlendirmelere göre QS dizilerinin sıralanmasında *Ortalama Karşılıklı*

Sıralama (Mean Reciprocal Rank)'yı geliştirmeye yardımcı olduğu gözlemlenmiştir [25]. Yapılan çoğu çalışma sorgular ve kullanıcılar arasındaki gizli bağımlılıkları ve anlamları göz ardı etmektedir. Çalışmada önerilen dikkat tabanlı hiyerarşik sinir QS yöntemi, kullanıcıların kısa ve uzun vadeli oturumlarını modelleyebilmektedir. Kullanıcı oturumlarını ve RNN'yi bir dikkat mekanizmasıyla birleştirmiştir. Bu dikkat mekanizması kullanıcı tercihlerini yakalamak için kullanılmıştır. Sonuçlar, kısa oturumlardaki iyileştirmelerin daha büyük olduğunu göstermiştir [26].

Bağlama duyarlı bir sinirsel sıralama modeli; arama bağlamını, devam eden bir arama görevinden önceki sorguları ve tıklamaları kullanarak hazırlanmıştır. Belge sıralaması ve sorgu üretimi için temsilleri ve bağlamları öğrenmek üzere iki seviyeli bir hiyerarşik tekrarlayan sinir ağı sunulmuştur. Kapsamlı sonuçlar, arama sorgusundaki bağlamı modellemede önerilen yaklaşımın değerini göstermiştir [27].

Amaca özel bilgi için mevcut derin sinir ağı modelleri genellikle oturum bilgilerini atlamaktadır ve yalnızca bireysel sorgu-belge veri çiftleri üzerinde eğitilmektedir. Sunulan çalışmada, belge sıralaması ve QS sonuçlarının birbirinden fayda sağlayabildiği bildirilmiştir. Model, belge sıralayıcı ve QS olarak iki ana bileşenden oluşan bağlam farkında bir derin sinir ağı mimarisini içerir. Ayrıca model birden fazla görevi yerine getirebilmektedir ve oturumların alaka düzeylerini yakalayabilmektedir. Modelin QS yapan bileşeni için Diziden-diziye model yaklaşımı kullanılmıştır. Önerilen çerçeve, son teknoloji benzerleri ile yapılan karşılaştırmalar sonrasında etkinliği kanıtlamaktadır [28].

Etiket çıkarma ve çizge tabanlı yaklaşımlar sınırlı kapsam ve genelleştirme problemlerine sahiptir. Yapılan çalışma, QS görevini, iki alt probleme ayıran iki aşamalı bir model önerilmiştir. Bu iki alt problem, sorguya göre ilgili bağlam kelimelerini keşfetmek ve bu bağlama bağlı QS üretmektir. Tekrarlayan sinir ağlarını, Diziden-diziye modeli ve *İlgili Kelime Üreticisi (Relevant Words Generator)* kullanan bir model önerilmiştir. Önerilen model, çeşitli metriklerde Diziden-diziye yaklaşımı içeren diğer modellere göre üstünlük sağlamıştır [29].

Bir QS sisteminin başarısı, kullanıcı arama amacını doğru bir şekilde anlayıp modellenmesine bağlıdır. Bu kapsamda sunulan çalışma, sinir ağlarını kullanarak geri bildirimlere duyarlı bir sistem sunmuştur. Önerilen model, kullanıcıya ait daha önce yapılan önerilerdeki tıklamaları dikkate almaktadır. Ek olarak sistemin dayanıklılığını ve genelleştirme performansını artıran olumsuz örnekler eklenmiştir. Yapılan değerlendirmeler, sunulan

sistemin rekabetçi bazı yöntemlere göre daha iyi performans ortaya koyduğunu göstermektedir [30].

QS algoritmalarından bazıları, tarihsel etkileşim verilerini kullanan Diziden-diziye öğrenme yaklaşımlarına dayalıdır. Öğrenme sürecindeki çeşitli zorluklardan dolayı bu tür yaklaşımlar, gerçek zamanlı olarak kullanıcı geri bildirimine uyum sağlamada başarısız olmaktadır. Çalışmada sunulan model algoritması, anında kullanıcı geri bildirimini dikkate almak için uzman tavsiyeleri ile *Çok-kollu haydut (Multi-armed bandit)* problemini kullanmıştır. Sonuçlar, literatürde bulunan *Dönüştürücü (Transformer)* tabanlı modellere ve benzerlerine kıyasla önerilen modelin *Kümülatif pişmanlığı (Cumulative-regret)* önemli ölçüde iyileştirdiğini göstermiştir [31].

Arama önerisi için *Block-Level Optimization'ı* kullanan yaklaşım, uçtan uca derin öğrenme mimarisinin, herhangi bir zamanda kendisine sunulan diğer tüm önerilerin, eşzamanlı olarak farkında olurken bütünsel bir şekilde öneriler sunabileceğini göstermiştir. Bu öneriler, karma amaçlı optimizasyon yöntemi ve çeşitli ödül fonksiyonları kullanılarak sıralanmıştır. Bu mimarinin faydaları geleneksel yaklaşımlara göre tartışılmıştır. Sistem, tıklama oranlarının ve çeşitliliğin eş zamanlı optimizasyonu göz önünde bulundurularak eğitilmiş ve sistemin farklı senaryolarda, herhangi bir çoklu öneri sistemine ölçeklenebileceği belirtilmiştir [32].

Etkili QS için önerilerin sıralaması önemlidir. Yapılan çalışmada, bunu sağlamak adına sıralamaları öğrenen bir yaklaşım sunulmuştur. Yaklaşım; önceki araştırmalara ek olarak aday önerileri, ortak tıklamalarla ilgili sorgulardan gelen terimleri içeren yapılandırılmış sanal belgeler olarak temsil etmiştir. Bu belgeler sorgu seyrekliğinin üstesinden gelmek için kullanılmıştır. Sonuçlar, çeşitlilik açısından etkili öneriler sağlandığını göstermiştir. Ayrıca çalışma, kaliteli önerileri tanımlamayı sağlayan sorgulardan bağımsız birkaç yaklaşım ve QS etkinliğinin nicel değerlendirmesi için yeni yaklaşımlar sunmuştur [33].

QS dizilerinin alaka düzeyini daha da geliştirmek ve daha fazla kaynak ekleyebilmek için önerilen makine öğrenmesi tabanlı yaklaşım, QS sonuçlarının alaka düzeyini artırmak için yararlı olmuştur. Önerilen model sorgu günlüklerinde hiç bulunmayan sorgularının da genelleştirilmesini sağlamıştır. Bu sayede geçmiş gözlemlerden genelleme yapılabilmesine geçmişte birlikte yapılan sorguların ötesinde önerileri oluşturmaya olanak tanımıştır. Çevrimiçi ve çevrimdışı yapılan testlerde, yaklaşımın güçlü temel çizgileri geride bıraktığı belirtilmiştir [34].

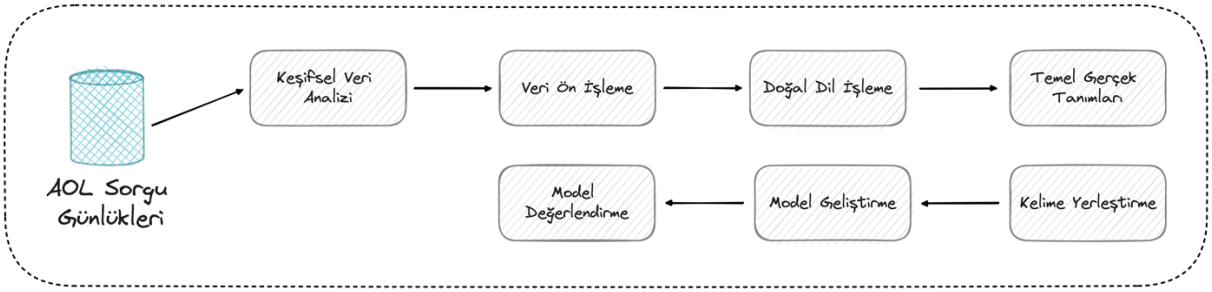
Doğal dil işleme görevlerinde son zamanlarda TA modelleri oldukça yaygın olarak kullanılmaya başlamıştır. Oldukça geniş bir alanda kullanılan TA modelleri kullanıldıkları alanlarda oldukça iyi performans sunmaktadır. QS alanında TA modelini ilk kez kullanan çalışma, önceden eğitilmiş bazı dönüştürücü ağlarını QS üretmek için kullanmıştır. Geniş bir arama veri kümesinde yapılan testler, TA modelinin QS için çok iyi performans gösterdiğini ve gürültülere karşı dayanıklılığını kanıtlamıştır [3].

TA kullanan bir diğer çalışma, QS için çeşitli önceden eğitilmiş TA modellerini inceleyerek kullanmıştır. Daha sonra bu modeller, RNN tabanlı mimariler ile kıyaslanmıştır. TA modelleri farklı *Tokenizer* ve *Kodlayıcılar (Encoder)* kullanılarak test edilmiştir. Ayrıca TA modelleri, düz ve hiyerarşik mimarilerle QS üretmek için kullanılmıştır. Sonuçlar, TA modellerinin RNN tabanlı mimarilere göre daha iyi performans sergilediğini göstermiştir. Ayrıca yapılan değerlendirmelerde hiyerarşik modellerin, QS alanında iyi performans sergilediğini, düz mimarilerin ise karmaşık ve uzun sorgularda daha kullanışlı olduğu belirtmektedir [2].

3. YÖNTEM

Çalışmada sunulan*, Dönüştürücü tabanlı Sorgu Öneri (Transformer-based Query Suggestion)* model mimarisinin detaylı olarak açıklandığı bu bölüm, çerçeveye ait yaklaşımları ve yöntemleri açıklamaktadır. Bölümde, çerçevenin geliştirme aşamalarına yardımcı olan veri toplama, veri ön işleme, veri hazırlama, derin öğrenme yöntemleri, model mimarisi, model eğitimi ve kapsayan tüm adımlar detaylı olarak sunulmaktadır. Bölümün genel amacı, kullanılan temel yöntemleri ve yaklaşımları kapsamlı olarak açıklayarak, alınan teknik kararların arkasında yatan önemli fikirleri belirtmektir. Özellikle TA kullanılan derin öğrenme yöntemleri ile doğal dil işleme yöntemlerinin birlikte kullanılması, çalışmanın temel ve en önemli yapı taşını oluşturmaktadır.

Arama motoru fonksiyonlarının geliştirilmesi bilgiye erişim açısından önemini korurken, TQS, gelişmiş derin öğrenme tabanlı TA modelini kullanarak arama sorgularını, arama sonuçlarını ve arama sorgularındaki kullanıcı deneyimini iyileştirmeyi hedeflemektedir. Önerilen TQS modelinin geliştirilmesine ait aşamalar Şekil 3.1'de gösterilmiştir.



Şekil 3.1: TQS çerçevesini oluşturan aşamaların diyagramı.

Bölüm, kullanılan veri kümesiyle ilgili bilgilerle başlamaktadır. Veriyi anlamlı hale getirmek ve model başarısını artırmak için önemli olan veri ön işleme için kullanılan temel yöntemlerden bahsedilerek devam etmektedir. Ardından kullanılan doğal dil işleme yöntemleri ve temel gerçek tanımlarından söz edilmektedir. Son olarak, derin öğrenme ve yöntemlerinden bahsedilerek aramalarda QS için TA modelinin ayrıntılı açıklaması sunulmaktadır ve önerilen modeli değerlendirmek için seçilen metrikler açıklanmaktadır. Seçilen değerlendirme metrikleri; önerilen çerçevenin, kullanıcılar için doğru ve alakalı QS üretmedeki etkinliğini kapsamlı olarak değerlendirmektedir.

Bölüm, çalışmada kullanılan yöntemlerin ayrıntılı bir açıklamasını sunarak, bu alanda veya benzer alanlardaki çalışmalar için yöntemleri anlamak ve geliştirmek isteyen kullanıcılara bir yol gösterici olmayı hedeflemektedir.

3.1. VERİ KÜMESİ

Derin öğrenme ya da TA modellerinin en önemli noktalarından biri kullanılan veri kümeleridir. QS için şimdiye kadar kullanılan çeşitli farklı veri kümeleri bulunmaktadır. Bu veri kümelerinden en sık kullanılan ve bilinenleri; AOL sorgu günlükleri kümesi, Microsoft Bing's veri kümesi, Yahoo, Sogou, TREC 2009, TREC 2010, 2011 internet izlemeleridir. Bu veri kümeleri QS alanındaki birçok çalışmada yaygın olarak kullanılmıştır. Microsoft Bing's veri kümesi 6 aylık bir periyottaki kullanıcı aramalarını içerir. Yahoo veri kümesi kullanıcı aramalarının oturum bağlamında varlıklarla ilişkilendirmek için yaygın olarak kullanılmaktadır. Sogou veri kümesi Çin merkezli arama motoru olan Sogou[5]'ya aittir. Arama ve sonuçlar ile ilişkili bir veri kümesi olduğundan arama motorlarının sonuçlardaki alaka düzeyini tahmin etmek ve sıralamak için kullanılmaktadır. Veri kümesinde arama sonuçlarının ekran görüntüleri, başlıkları, ön izlemeleri ve HTML kaynak kodları gibi kapsamlı bilgiler bulunmaktadır. TREC 2009, 2010 ve 2011 internet izlemeleri veri kümeleri, *Text Retrieval Conference*[6] tarafından sağlanan geniş kapsamlı ve amaçlı veri kümesidir. Çeşitli ölçekteki veri kümeleri (Kategori A, Kategori B gibi) belirli bir ücret karşılığında satın alınabilmektedir. Her sene TREC veri kümesi içerikleri güncellenmektedir.

3.1.1. AOL Veri Kümesi

Çalışmada AOL sorgu günlüklerinin iki farklı fomatı kullanılmıştır. Bunlardan ilki temel gerçekleri oluşturabilmek için çeşitli ön işleme adımlarını içerisinde bulunduran ham haldeki AOL sorgu günlükleridir. AOL sorgu günlükleri verilerine kolay ulaşım sağlanabilmesi, literatürdeki çalışmaların ağırlıklı olarak bu veri kümesini kullanması ve derin öğrenme modellerini geliştirmek için gereken büyük miktarda veriye sahip olması bu veri kümesisin tercih edilmesindeki temel sebeplerdendir. AOL veri kümesi, çalışma boyunca sunulacak arama motorları için QS araştırmasının temel noktası olacaktır. Çeşitli ve kapsamlı kullanıcı sorguları, kullanıcı ilişkileri ve kullanıcı oturumlarından oluşan veri kümesi, gerçek dünyadaki kullanıcı davranışlarını analiz etmek ve sorguların amaçlarını anlamak için oldukça

[5] https://www.sogou.com/
[6] https://trec.nist.gov/

yardımcı olacaktır. Bu bölümde, araştırmanın temelini oluşturan veri kümesi tanıtılmakta, veri kümesinin yapısından ve veri kümesine ait çeşitli istatistiksel değerlerden bahsedilmektedir.

AOL sorgu günlükleri, arama motorları ile ilgili çalışmalarda yaygın olarak kullanılan veri kümelerinden biridir [11, 16, 35, 36]. AOL sorgu günlükleri, 1 Mart 2006 ile 31 Mayıs 2006 tarihleri arasındaki sorguları içermektedir ve yaklaşık 36 milyon kullanıcı sorgusu içerir [36]. Sorgu günlükleri, 5 farklı sütundan meydana gelmektedir. Bu sütunlar ve karşılık gelen anlamları Tablo 3.1'de gösterilmiştir. Kullanıcı arama sorgusuna karşılık gelen sonuç listesinden herhangi bir sonuca tıkladıysa veri kümesindeki bulunan beş alanın tamamı doludur. Fakat kullanıcı bir arama yapıp ardından herhangi bir bilgiye tıklamadıysa *ItemRank* ve *ClickURL* alanları boş olarak görüntülenmektedir. Kullanıcıların tek bir sorgusu için birden fazla arama sonucuna tıklaması durumunda, kullanıcı kimliği ve sorgusunun aynı kaldığı, tıklama adresi ve tıklanan nesnenin sırasının değiştiği ikinci bir kayıt eklenmiştir.

Tablo 3.1: AOL sorgu günlüklerine ait sütunlar ve anlamları.

AnonID	Anonimleştirilmiş kullanıcı kimlik bilgisi.
Query	Kullanıcı tarafından arama için kullanılan sorgu.
QueryTime	Arama için kullanılan sorgunun kaydedildiği zaman.
ItemRank	Arama sonrası tıklama yapılan belgenin sırası.
ClickURL	Kullanıcın arama sonrası tıkladığı belgeye ait web adresi.

Veri kümesini analiz etmek için yapılan çalışmalarda bazı sayısal veriler elde edilmiştir. Bu sayısal istatistikler Tablo 3.2'de gösterilmiştir. Benzersiz kullanıcı sayısını hesaplamak için sorgu günlüklerinde maskeli olarak sağlanan *AnonID* alanı kullanılmıştır. Ayrıca aynı alan kullanılarak veri kümesindeki kişi başına ortalama arama sayısı da hesaplanmıştır. Ardından tıklama içeren ve içermeyen sorgu sayıları hesaplanmıştır. Tıklama içeren sorgular, kullanıcılar için doğrudan hedef sorgular olarak işaretlenebileceği için sunulan TA modelinin oldukça önemli bir bilgi kaynağı olarak adlandırılabilir. Kullanıcılara ait benzersiz sorgu sayısı ve kullanıcıların en sık kullandığı arama sorgu cümlecikleri de belirlenmiştir. En sık aranan sorgu cümlecikleri Şekil 3.2'de gösterilmiştir.

Tablo 3.2: AOL sorgu günlüklerine ait çeşitli bilgiler.

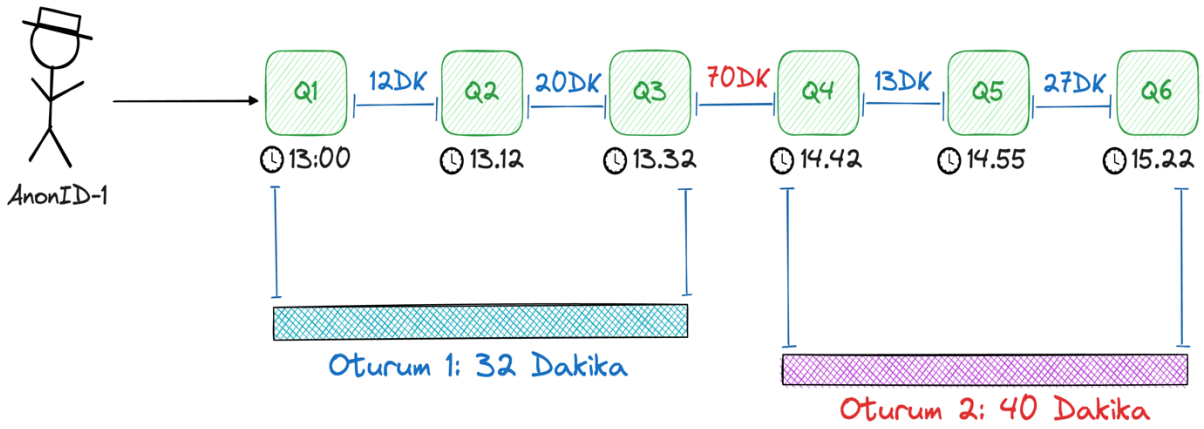
Sorgu	35.044.627
Benzersiz Kullanıcı	649.327
Benzersiz Oturum	10.382.084
Benzersiz Sorgu	8.628.001
Kullanıcı Başına Ortalama Sorgu	53.97
Oturum Başına Ortalama Sorgu	3.37
Tıklama İçeren Arama	19.068.563
Tıklama İçermeyen Arama	15.976.064



Şekil 3.2: En sık kullanılan sorgu cümlecikleri.

Veri kümesi incelenirken tüm arama motorlarında ve önceki çalışmalarda belirtildiği gibi [6] burada da bazı robot arama oturumlarının bulunduğu gözlemlenmiştir. Bu robot aramalara ek olarak tek karakterden oluşan, özel işaret içeren ya da sorgu kaydı içermeyen bazı aramalarında sorgu günlüklerinde bulunduğu tespit edilmiştir. Bu gibi bozuk ve anlamsız veriler, veri kümesindeki tutarlı ve anlamlı verilerinde bozulmasına yol açarak derin öğrenme tabanlı modellerin çalışma performansını ve başarısını etkileyebilmektedir. Bu yüzden, bu değerlerin model geliştirme aşamasına geçmeden önce veri kümesinin içerisinden çıkarılması gereklidir. Çalışma kapsamında, veri kümesinde bulunan bu tip bozuk ve aykırı değerleri tespit etmek için özel karakter filtreleri ve *Düzenli İfadeler'den (Regular Expression)* yararlanılmıştır.

Son olarak veri kümesi içerisine, kullanıcı deneyimini iyileştirmek amacıyla kullanılacak olan oturum bilgisi yapay olarak eklenmiştir. Oturum bilgisi oluşturulurken, kullanıcıların mevcut araması ve bir önceki araması arasında geçen süre kontrol edilmiştir. Geçen süre 30 dakikadan fazla olduğu durumda, bu yeni bir kullanıcı oturumu olarak değerlendirilmiştir. Oturum bilgisinin nasıl oluşturulduğu Şekil 3.3'te detaylı olarak belirtilmiştir.



Şekil 3.3: Her bir kullanıcı için oturumların belirlenme durumu.

3.1.2. İşlenmiş AOL Veri Kümesi

Hiyererşik tekrarlayan kodlayıcı-kod çözücü mimarisini sunan çalışmada [6], AOL sorgu günlüklerini kullanarak oluşturulan, veri ön işleme adımının uygulandığı ve yukarıda bahsedildiği gibi 30 dakikalık zaman aralığına bölünmüş kullanıcı oturumların temsil edildiği veri kümesidir. Bu çalışmada verilerinin bahsedilen çalışmaya uygun olarak hazırlanabilmesi için çalışma ile ilgili olarak sunulan açık kaynaklı projeden[7] yararlanılmıştır. Projenin kaynak kodları kullanılarak ham verilerin ilgili çalışmada kullanılan veri kümesine dönüştürülmesi sağlanmıştır. Böylece önerilen model ile temel çizgiler arasında yapılacak değerlendirmelerde aynı veri kümesininin kullanılması sağlanmıştır. Bu veri kümesi hazırlanırken, alfanümerik olmayan karakterlerin tamamı çıkarılmıştır. Ayrıca *yazım düzeltici (spelling corrector)* kullanılmıştır ve tüm karakterler küçük harflere dönüştürülmüştür. Ardından sorgular, sorgu zamanına göre sıralanarak ilk iki ayı içeren veriler eğitim kümesi olarak ayırılmıştır. Veri kümesinin kalan son bir aylık kısmı ise değerlendirme ve test veri kümesi olarak ayırılmıştır. Veri kümesindeki ön işleme ve oturum hazırlama adımlarının tamamlanmasının ardından son durumda, eğitim kümesi içerisinde 1.708.224, değerlendirme kümesinde 416.450 ve test kümesinde de aynı şekilde 416.450 oturum bulunmaktadır. Eğitim kümesinde en çok

[7] https://github.com/sordonia/hred-qs

karşılaşılan 90K kelime, önerilen model ve temel çizgiler için kelime dağarcığını oluşturmaktadır.

3.2. VERİ ÖN İŞLEME

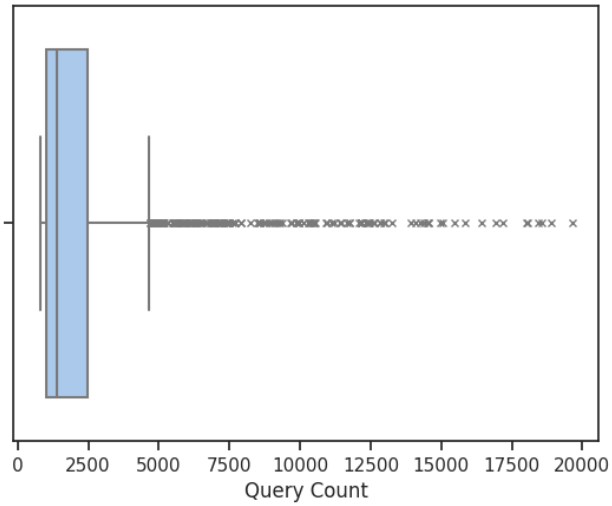
Veri ön işleme, makine öğrenmesi ve derin öğrenme başta olmak üzere birçok yapay zekâ çalışmasında uygulanan önemli bir aşamadır. Günümüz dünyasında çevrimiçi ortamlarda bulunan veriler giderek büyümektedir. Çevrimiçi dünyada ortaya çıkan bu büyük bilgi kaynakları işlenerek yeni bilgiler ve deneyimler öğrenilebilir. Fakat bu bilgi kaynaklarında bulunan veriler genelde eksiklikleri, tutarsızlıkları ve aykırılıkları içerisinde bulundurur [37]. Veri ön işleme, veri içerisindeki belirtilen problemlerin ortadan kaldırılmasını ve kullanılacak olan modeli oluşturan adımların talep ettiği duruma uygun hale getirilmesini kapsar. Ayrıca veri ön işlemedeki tüm tekniklerin birlikte kullanılması ya da hiçbirinin kullanılmaması yerine, uygun yöntemlerin seçilerek birlikte kullanımının üzerinde çalışılan alana göre model başarısını artıracağı kanıtlanmıştır [38]. Veri ön işleme ya da metin ön işleme adımının iyi kurgulanması ve uygulanması, bu adımdan sonra uygulanacak olan doğal dil işleme gibi süreçleri de doğrudan etkileyen önemli bir alandır [39].

Model performansını belirleyen ilk unsur verinin kalitesidir. Ham veriler, genellikle kullanılan modelin performansını olumsuz yönde etkileyebilecek olan aykırı, gürültülü, bozuk ve eksik değerlere sahiptir. Veriyi bu gibi değerlerden arındırmak önemlidir. Ardından eğer varsa özelliklerin belirli bir aralığa indirgenmesi gerekebilir. Metin verilerinin bulunduğu veri kümelerinde kelimelere ayırma, metin işleme ve doğal dil işleme gibi farklı yöntemler kullanılmaktadır. Bu uygulamaların ardından, model için önemli olacağı düşünülen özellikler oluşturulmalı ve seçilmelidir. TA modelleri doğaları gereği diziler ile çalışmaktadır. Bu yüzden özellik dizilerin sabit uzunluklarda olması model başarısı için belirleyici olabilir. Verilerin eğitim, doğrulama ve test gibi bölümlere ayrılması, oluşturulan makine öğrenmesi modeline ait değerlendirmelerin daha doğru ve tutarlı yapılmasını sağlar. Aşırı örnekleme, yetersiz örnekleme gibi durumlar için çeşitli dengeleme yöntemlerinin kullanılması da önem taşır. Veri ön işleme, model kararlılığını, başarısını ve performansını doğrudan etkileyen önemli bir aşamadır.

QS probleminin doğası gereği, literatürdeki veri kümelerinin tamamı metin verilerini içermektedir. Bazı çalışmalarda kullanılan veri kümeleri metin verilerine ek olarak ekran

görüntüleri ve sayısal değerlerde içermektedir. Çalışmada kullanılan veri kümesinde de veri ön işleme adımının ana odağı metin verilerini temizlemektir. Bunun için öncelikli olarak veri kümesini tanımak adına bir önceki aşamada anlatılan bazı keşif ve istatistik çalışmaları yapılmıştır. Öncelikli olarak veri kümesinde, *Query* alanında herhangi bir ifade bulunmayan, sadece 1 karakterden oluşan, 400 karakteri geçen sorgular, sadece noktalama işareti bulunan ve anlamlı olmayan harflerden oluşan sorgular uzaklaştırılmıştır. Buradaki 400 karakter veri kümesinde yapılan çalışmalardan elde edilmiştir. İncelemeler sonunda, 400 karakteri geçen sorguların neredeyse tamamının anlamsız tuş vuruşları olduğu görülmüştür. Bu yüzden belirtilen tüm bu özelliklere sahip veriler, veri kümesinden uzaklaştırılmıştır. Bu verilerin uzaklaştırılmasının, oluşturulacak temel gerçeklerin kalitesini olumlu olarak etkileyeceği öngörülmüştür. En çok arama yapılan sorgu cümlecikleri, temel gerçeklerin belirlenmesi sırasında olumsuz etkiye sebep olduğu ve kaliteyi düşürdüğü gözlemlendiği için veri kümesinden çıkarılmıştır. Benzer şekilde, veri kümesindeki sorgu sıklıkları incelendiğinde çok az sayıda aranan kelimelerin de oluşturulan temel gerçeklerin kalitesini negatif yönde etkilediği görülmüştür. Bu nedenle çok sık aranan sorgulara benzer şekilde çok az aranan sorgularda veri kümesinden çıkarılmıştır. Bu aşamadan sonra veri kümesinde çeyreklikler hesaplanmış ve sorgu sayısı aralıklarının daha tutarlı bir şekilde dağıldığı görülmüştür. Bu çeyrekler ile veri kümesinin sorgu sayısı dağılımı Şekil 3.4'te gösterilmiştir. Çeyrekler arası aralığa göre aykırı değerleri bulmak için kullanılan yöntem Denklem 3.1'de gösterilmiştir. Bu aralık kullanılarak, aykırı değerler tespit edilmiş ve veri kümesinin içerisinden çıkarılmıştır. Kalan veriler incelendiğinde, kullanıcı sorgularının büyük bir çoğunluğunun doğrudan internet sayfalarına ait adresler olduğu görülmüştür. Aynı şekilde veri kümesindeki kullanıcı tıklamaları da doğal olarak internet adreslerinden oluşmaktadır. Tıklama verileri de hedefleri belirlemek için önemli olduğundan bu ön işleme adımının tıklama verileri üzerinde de uygulanması önemlidir. Bunun için ilk olarak internet adresi şemalarını ve bazı internet uzantılarını içeren etkisiz kelimeler listesi oluşturulmuştur. *"com, www, http, https"* etkisiz kelimelere örnek olarak gösterilebilirler. Bu etkisiz kelimeler listesi kullanılarak veri kümesindeki sorgu cümlecikleri ve tıklama verileri etkisiz kelimelerden temizlenmiştir. Ardından metin verileri içerisindeki noktalama işaretleri, özel karakterler ve birden fazla sayıda bulunan boşluk karakteri temizlenmiştir. Arama sorgularındaki büyük ve küçük karakterlerin bir kullanıcı için karakteristik olarak aynı anlamı taşıdığını kabul eden çalışmaya [6] benzer şekilde tüm karakterler küçük harflere dönüştürülmüştür. Böylece, *Canada* ve *canada* aramalarının aynı forma ait olduğu ve tam olarak aynı anlamı taşıdığı kabul edilmiştir. Çoğu doğal dil işleme

görevi için büyük ve küçük harf hassasiyetinin ortadan kaldırılmasının, uygulandığı görevin başarısını iyileştirdiği düşünülmektedir. Ön işleme adımı, genel manada sonraki aşamalara ait yöntemlerin başarısını artırırken, aynı yöntemler için bazı küçük öngörülebilir kayıplara neden olabilir. Çünkü veri kümesinden çıkarılan bazı bilgiler bir sonraki aşamada bu veriler kullanılarak elde edilebilecek değerli bilgilerin önüne geçebilir.



Şekil 3.4: Veri kümesine ait çeyreklikler ve aykırı değer aralıkları.

$$\mathrm{IQR} = \mathrm{Q}_3 - \mathrm{Q}_1$$

$$\mathrm{LowerFence} = Q_1 - 1.5 \cdot I\mathrm{QR} \tag{3.1}$$

$$\mathrm{UpperFence} = Q_3 + 1.5 \cdot I\mathrm{QR}$$

Çalışmanın ana odaklarından biri, arama sorgusu önerilerindeki kullanıcı deneyimini iyileştirmek olduğu için kullanıcı oturumlarının belirlendiğine bir önceki bölümde değinilmiştir. Bu adımda, kullanıcı oturumları incelenerek gerçek bir kullanıcıya ait olamayacağı düşünülen çeşitli oturumlar veri kümesinin içerisinden çıkarılmıştır. Sonuç olarak veri kümesi, kapsamlı bir incelemeden geçirilerek, temel gerçeklerin hazırlanmasının öncesinde kalitenin arrtırılması amacıyla daha işlevsel ve tutarlı hale getirilmiştir.

3.3. DOĞAL DİL İŞLEME

Doğal dil işleme, insanlar ve makineler arasında doğal dil aracılığıyla etkili bir iletişim kurmayı sağlamak için geliştirilen teori ve yöntemlerin bir bütünüdür [40]. Doğal dil işleme konusundaki çalışmalar, 1948 yılı sonrasında başlamıştır. Bu yıllarda Warren Weaver gibi bilim insanları makine çevirisi alanında çalışmalar yürütmüştür. Ardından gelen süreçte, doğal dil

işleme ilgili birçok gelişme meydana gelmiştir. Yakın dönemde ise artan veriler, hesaplama gücünün yükselişi ve makine ile derin öğrenme tekniklerinin yükselişiyle beraber doğal dil işleme alanında ciddi bir ilerleme kaydedilmiştir. Günümüzde ortaya çıkan RNN, CNN ve TA modelleri de makine ve insan arasındaki etkili iletişim görevlerini önemli ölçüde iyileştirmektedir.

Doğal dil işleme yöntemleri; metin verilerini işlemeyi, analiz etmeyi ve bunlardan anlam çıkarmayı kapsayan yöntemlerin tamamını kapsamaktadır. Metinleri daha küçük alt varlıklara bölme, kelime yerleştirme, adlandırılmış varlık tanıma, kelime ayırma ve makine çevirisi doğal dil işleme yöntemlerinden bazılarıdır. Bu yöntemlerin bazıları ayrı ayrı ya da birlikte kullanılarak yapay zekâ modellerinin başarısı artırılabilmektedir. Bu nedenle doğal dil işleme teknikleri, yapay zekâ gelişmelerinin de etkisinde çeşitli alanlarda kullanılmaya ve gelişmeye devam etmektedir.

Doğal dil işleme teknikleri, QS alanındaki çalışmalarda da diğer alanlarda olduğu gibi yaygın bir kullanıma sahiptir. Makine ve derin öğrenme tabanlı çalışmaların dışında anlamsal yaklaşımlar ve çizge tabanlı yaklaşımlarda da yaygın olarak kullanılmaktadır. Son zamanlarda literatürde bulunan çalışmaların bazıları kelime yerleştirmeleri kullanmaktadır. Sıra, içerik ve sorgu yerleştirmelerini girdi olarak yapay sinir ağı modeline veren çalışma, kelime yerleştirmelerin model başarısını artırdığını belirtmektedir [5]. Bir diğer çalışma yine sorgu seviyesinde yerleştirmelerden yararlanarak kelimeler arasındaki benzerliği ölçmüştür [41]. 2017 yılının ortalarında ortaya çıkan TA [7] doğal dil işleme yöntemlerinde devrim yaratmıştır.

TA modellerinin ortaya çıkışı ve gelişimi ile doğal dil işleme süreçlerinin kullanıldığı görevlerde; *BERT* [42], *GPT* [43], hatta GPT'nin daha gelişmiş formlarından biri olan *GPT-3* ve BERT mimarisinin daha iyi ölçeklenen ve daha az parametre ile kullanılabilen hali olan *ALBERT* [44] kullanılmaya başlamıştır. Yapılan çalışmalardan bir diğeri, ALBERT'i sorgu yerleştirme ve TA modeline ait kodlayıcı olarak kullanmıştır [32]. QS alanında sunulan TA modelleriyle yapılan çalışmalar, çoğu yapay sinir ağı modeline göre çok daha iyi performans göstererek, dönüştürücülerin metin verilerindeki başarısında olduğu gibi QS görevinde de başarısını ve performansını doğrulamıştır.

Çalışmada kullanılan veri kümesinde ön işlemin tamamlanmasının ardından önerilen çerçeveyi oluşturmak için bir sonraki adım olan doğal dil işleme süreçleri ile devam edilmiştir. Veri kümesinde yapılan incelemeler sırasında, sorgu günlüklerinin çok çeşitli formda sorgu

cümleciğini barındırdığı gözlemlenmiştir. Bir önceki aşamada yapılan ön işleme ardından internet adresi şemalarından arındırılan sorgular ve tıklama adresleri, doğal olarak birleşik kelimeler içermektedir. Bu birleşik cümlecikler hem kullanıcı sorgularını hem de hedef sorgu verilerini oluşturmak için doğru formda bulunmamaktadır. Birleşik formdaki metin verilerini daha kabul edilebilir bir forma dönüştürmek ya da doğru şekilde ayırmak için, *n-gram* modelleri [36, 45] veya kelime yerleştirmeleri [46] yaygın olarak kullanılmaktadır. Bu çalışmada, kolayca uygulanabilir olması ve çeşitli n-gram türlerini uygulayan hazır kütüphanelerin bulunması nedeniyle n-gram modelleri tercih edilmiştir. N-gram modeli temel olarak belirli bir dizi içerisindeki *n* tane ögenin ardışık dizilimlerinin bütününü içerir. Bir cümle için uygulanmış basit bir *unigram*, *bigram* ve *trigram* örneği Şekil 3.5'te gösterilmiştir. Çeşitli n-gram modelleri kullanılarak sorgu günlüklerine ait birleşik sorgu cümleciklerinin olabilecek en doğru şekilde ayrıştırılması sağlanmıştır. Bu işlemi gerçekleştirmek için açık kaynak kodlu *python-wordsegment*[8] kütüphanesinden yararlanılmıştır. Bu kütüphanenin kaynak kodları ise Peter Norvig'e ait kitabın ilgili bölümünden alınmıştır [47]. Kelimeleri birleşik hallerinden ayırmak için kullanılan yöntem ve kütüphane, test edilen diğer kütüphanelere göre kelimeleri birleşik hallerinden ayırmak için daha iyi performans göstermiştir. Ham sorgu, ön işleme uygulanmış sorgu ve ön işlem sonrasında elde edilen sorgunun birleşik kelimelerinin ayrılmasını kapsayan örnek bir sorgu Şekil 3.6'da gösterilmiştir.



Şekil 3.5: Bir cümleye ait unigram, bigram ve trigram örnekleri.

[8] https://github.com/grantjenks/python-wordsegment



Şekil 3.6: Örnek bir sorgu cümleciğinin veri ön işleme ve doğal dil işleme süreci.

3.4. TEMEL GERÇEKLER

Yapay zekâ uygulamalarında *Temel Gerçekler (Ground Truths)* ifadesi, veri kümelerindeki mutlak gerçekleri ifade etmek için kullanılır. Yani temel gerçek, eldeki veriler içerisindeki en güvenilir ve en doğru bilgiyi temsil eder. Makine öğrenmesi, derin öğrenme gibi yapay zekanın alt alanlarında geliştirilen modellere ait; ölçümlerin, değerlendirmelerin ve kıyaslamaların yapılabilmesinde referans görevini üstlenir. Amacı görüntülerdeki nesnelerin ne olduğunu tahmin etmek olan bir görüntü işleme uygulaması için temel gerçek, görüntülerdeki nesnelerin ne olduğunu gösteren bilgidir. Bu bilgi insan gücü ile elde edilebileceği gibi bazı makine süreçleri ile de gerçekleştirilebilir. Farklı alanlara ait temel gerçekler de farklılık gösterebilir. Bir coğrafi bilgi sistemi için ilgili koordinatlarda hangi nesnenin bulunduğu temel gerçek olabilirken metin verileri için bir kelimeden sonra gelecek diğer kelimenin hangisinin olacağı temel gerçek olabilir. Belirli bir görev için hazırlanan öğrenme modeli için, doğru etiket verisinin belirlenmesi bu görev için temel gerçeği tanımlayabilir.

Model ve algoritmaların performanslarını değerlendirmek için temel gerçek büyük önem taşımaktadır. Çünkü temel gerçekler model performansını değerlendirirken referans görevini üstlenir. Kısaca temel gerçek tanımlamaları, modellerin doğruluğunu, performansını, etkinliğini ve kalitesini değerlendirmekte doğrudan etkili bir adımdır.

Metin verilerinde temel gerçeklerin belirlenmesi için bazı adımları takip etmek gereklidir. Metinlerin kullanılacağı göreve göre temel gerçek tanımları farklılıkları içerebilir. Bu sebeple ilk olarak metin verilerinin hangi görev için kullanılacağı belirlenmelidir. Daha sonra bu verilerin nasıl etiketleneceklerini belirten ayrıntılı açıklamalar ve kriterler geliştirilmelidir. Eğer mevcutsa, bu açıklama ve kriterlere ek destekleyici açıklamalar ve kriterler hazırlanmalıdır. Son olarak belirlenen açıklama ve kriterlere göre etiketlenmiş veriler incelenerek, temel gerçeklerin kalitesi kontrol edilmelidir. Eğer inceleme sonrasında, etiket

verilerinin uygun olduğunu karar verilirse bu veriler ile temel gerçek kümesi oluşturulur. Bu verilerin nasıl oluşturulduğuna dair detaylı açıklama verilmelidir ve tekrar kullanılabilirliği sağlanmalıdır. Ardından göreve özel geliştirilen model ya da uygulama temel gerçek veri kümesinden yararlanılarak değerlendirilebilir. Metindeki adlandırılmış varlıkların çıkarılması ve metin sınıflandırma için önceden belirlenmiş sınıflara göre metinleri kategorize etme gibi görevler metin verilerinde temel gerçek belirlemeye örnek olarak gösterilebilir.

QS görevinde temel gerçeklerin belirlenmesi, kullanıcının belirli bir sorgusuna karşılık gelen uygun ve doğru ve ilgili bulacağı bir dizi sorguların oluşturulmasını kapsar. Kullanılan veri kümesi büyüklüğüne göre temel gerçekler, veri kümesindeki bir veya birden fazla sorgunun bir araya gelmesiyle oluşturulabilir. Oluşturulan bu sorgular ile ilgili bir inceleme ve onay mekanizması kullanılmalıdır. Ayrıca temel gerçek kümesinin yinelemeli olarak iyileştirilmesi (yeni gelen sorgular için yeni bir dizi temel gerçek belirlenmesi) model performansını yinelemeli olarak iyileştirmeye katkıda bulunur [48]. QS için temel gerçekleri belirlemenin ana amacı; kullanıcılar için ana sorguya en yakın, en alakalı, en faydalı ve en yüksek kalitede alternatif sorguların oluşturularak, geliştirilecek modeller için bu sorguların etiketli veriler olarak kullanılmasını sağlamaktır.

Temel gerçekler belirlenirken genellikle nesnel doğulardan yararlanılır. Örneğin bir görüntüde kedi görülüyorsa bu herkes için kedi görüntüsüdür ve bu nedenle bu görüntüye ait temel gerçeğin kedi olduğu söylenebilir. Literatürde çeşitli alanlarda temel gerçekleri oluşturabilmek için kullanılabilecek pek çok yaklaşım bulunmaktadır. Tarihi belgelerdeki el yazılarını tanımlamak için kullanılan yarı otomatik bir temel gerçek belirleme yöntemi [49], hasta kayıtları ile ilgili dokümanlar için temel gerçek yöntemi [48], nefret söylemini belirlemek için nefret tanımlarını içeren bir temel gerçek hazırlama yöntemi [50] gibi farklı temel gerçek oluşturma çalışması mevcuttur. Fakat bazı temel gerçek yöntemleri bazı görevler için kullanılabilir olsa da bazı görevlerde öznel temel gerçekler belirlenmesi gerekebilir. QS alanında kullanıcıya ait etiket verileri oluştururken kedi örneğindekine benzer şekilde net bir temel gerçek yargısında bulunmak mümkün değildir. Yapılan çalışma, buna benzer bireysel temel gerçek belirleme yöntemlerinin, en sık kullanılan nesnel temel gerçek belirleme yaklaşımlarından daha iyi sonuç verebileceği gösterilmektedir [50].

Çalışmada bahsedilen iki veri kümesinden biri olan ham AOL sorgu günlükleri kullanıcılara ait bir hedef sorgusunu içermemektedir. Bir diğer veri kümesi olan işlenmiş AOL sorgu günlükleri de kullanıcı oturumlarındaki bir sonraki sorguyu hedef sorgu olarak kabul

etmektedir. Bu çalışmada gelecek çalışmalara da yardımcı olabileceği düşünülerek sorgu önerisi görevine özgü olarak belirli öznel yöntemler kullanılarak temel gerçekler oluşturulmuştur. Çalışmada sunulan TQS modelinin temel çizgilerle karşılaştırılmalı olarak test edilebilmesi adına burada oluşturulan temel gerçekler yerine işlenmiş sorgu kümesi ile bir sonraki kullanıcı sorgusu hedef olarak kullanılmıştır. Bu bölümde sunulan temel gerçekler, sorgu önerisi görevine özel olarak oluşturulacak TA modellerinde kullanılabilir. Fakat bu türdeki problemler için mükemmel bir temel gerçek oluşturmak oldukça zordur [2]. Sorguların nadirliği, karmaşıklığı ve arama işleminin doğası gereği temel gerçek belirlenmesi zorlu bir süreçtir. Bu zorlukların üstesinden gelebilmek adına birden çok bakış açısını içeren bir temel gerçek belirleme stratejisi üzerinde durulmuştur. Bunun için öncelikle sorgu günlükleri detaylı olarak incelenmiştir. Temel gerçekleri oluşturmak için veri kümesindeki hangi alanlardan ve hangi türden bilgilerden yararlanılabileceği düşünülmüştür. Burada öznel bir yaklaşım ile hazırlanan temel gerçek belirleme yöntemi kullanılmıştır. Temel gerçekleri belirlemede kullanılan öznel yaklaşımlar şu şekildedir:

I. Veri kümesindeki tüm sorgu cümlecikleri vektör olarak ifade edilmiştir. Bu vektörler, *Kosinüs Benzerliği (Cosinus Similarity)* ile değerlendirilerek bir sorgu için en yakın benzerliğe sahip diğer sorgu cümlecikleri belirlenmiştir. En benzer olan cümleciklerden belirli sayıdaki ifade seçilerek temel gerçek veri kümesine eklenmiştir.
II. Bir sorgunun daha önce arandığı tüm oturumlar belirlenmiştir. Belirlenen bu oturumlarda ilgili sorgu dışında farklı hangi sorgu cümleciklerinin arandığı tespit edilmiştir. İlgili sorgu dışındaki en sık aranan sorgulardan belirli sayıdaki sorgu temel gerçek kümesine eklenmiştir.
III. Aranan sorguyu içeren ve tıklama eylemine sahip tüm veriler incelenmiştir. Bu veriler arasında en fazla tıklama almış olan internet adresleri tespit edilmiştir. Bu internet adreslerinden en fazla tıklama sayısına sahip internet adreslerine ait alan adları temel gerçek kümesine eklenmiştir.

Temel gerçekleri belirlemek için kullanılan öznel yaklaşım, tavsiye sistemlerinde yaygın olarak kullanılan ve başarısını kanıtlamış olan; kullanıcıların kullanıcılara benzerliklerini ve arama cümleciklerinin arama cümleciklerine benzerliklerini içermektedir. Temel gerçekleri belirlemek için kullanılan 3 yöntem, sadece kullanıcı oturumlarına bakmak ya da sadece sorgu cümleciklerinin benzerliklerine bakmak yerine birden fazla bakış açısını

beraber kullanarak, belirlenen temel gerçeklerin kalitesini de doğrudan etkilemektedir. Bu tez çalışmasında sunulan, birden fazla bakış açısını içeren bu melez temel gerçek belirleme yöntemi, bilindiği kadarıyla QS alanında temel gerçekleri belirlemek için kullanılan ilk yöntemdir.

Bu tanımların uygulanması ile oluşturulan temel gerçek kümesi ve bunlara karşılık gelen sorgulardan bazıları Tablo 3.3'te gösterilmiştir. Temel gerçek kümesi, veri kümesindeki baskın sorgu cümleciklerinden fazla etkilendiği için veri kümesi, ön işleme adımında bahsedildiği gibi baskın ve aykırı değerlerden arındırılmıştır. Fakat hala bazı temel gerçeklerin çeşitli baskın sayılabilecek değerden etkilendiği ve bu değerleri tekrarladığı görülebilir. Bunun için aykırı değer aralıkları değiştirilerek, bazı baskın değerlerin bazı sorgulara ait temel gerçeklere sürekli olarak eklenmesi engellenebilir. Fakat bu çalışmada belirlenen aykırı değer aralıkları doğru kabul edilerek çalışmanın kalanı bu şekilde yürütülmüştür. Aykırı değer aralıkları değiştirilerek daha kaliteli bir temel gerçek kümesinin oluşturulması mümkündür.

Tablo 3.3: Bazı sorgular için 3 farklı yöntem ile oluşturulan temel gerçekler.

Sorgu	I.	II.	III.
poetry	poems, love poems, mother day poems, music lyrics, music song lyrics, music	poems, anna benson, migente, song lyrics	poets, poems, emule, eserver, giggle poetry, poetry eserver, loc, poetry 4kids, poetry magazine, allpoetry
toyota	nissan, lexus, mazda, honda, bmw, chevy	honda, nissan, ford, mazda	toyota jp, toyota au, buy a toyota, is this toyota, toyota f1, toyota my, toyota ca, toyota nation, scion, toyota financial
free online games	online games, free bingo, free, free downloads, free music, free screensavers	anna benson, migente, mini recent, song lyrics	miniclip, free arcade, riddler, shockwave, freeworldgroup, play free online games, free games online au, games msn, fetch fido, games
jessica simpson	ashlee simpson, jessica alba, pamela anderson, ann taylor, anna nicole smith, jennifer aniston	jessica alba, pam anderson, eva longoria, carmen electra	superiorpics, askmen, imdb, sweet kisses, mtv, e for u, starpulse, phun, good plastic surgery, celebritywonder
passport	travel, immigration, airline tickets, citizen bank, credit cards, discover card	anna belson, migente, song lyrics, mini recent	travel state, american passport, i afdb travel state, travis a, passport express, passports and visas, link referral, ppt gcc a, microsoft, s account services passport

Temel gerçek kümesini oluşturmak için hazırlanan 3 yöntem farklı sayıda QS üretmek için kullanılmıştır. Daha sonra bu üç farklı yöntem ile üretilen QS sonuçları birleştirilerek temel gerçek kümesi ortaya çıkarılmıştır. Kelimelerin benzerliklerini kullanan ilk yöntem kullanılarak 6 adet QS oluşturulmuştur. İkinci yöntemin performansı diğer iki yönteme göre daha kötü görüldüğü için bu yöntem ile sadece 4 adet QS üretilmiştir. Son yöntem oldukça kaliteli QS üretmeye elverişli olduğu için 10 adet QS bu yöntem kullanılarak oluşturulmuştur. Bir sorgu cümleciği için toplamda 20 adet QS hazırlanmıştır.

Temel gerçekleri belirlemek için kullanılan yöntemlerden en iyi performans gösteren yöntem yapılan incelemelere göre kullanıcı tıklama verileri kullanılarak oluşturulan III. Yöntem olmuştur. Bu durum tablodaki verilerden de gözlemlenebilmektedir. Kullanılan I. Yöntem de III. Yöntem'e benzer şekilde oldukça iyi bir performans göstermiştir. Fakat kullanıcıların oturum verilerinden gelen sonuçları kullanan II. Yöntem, çoğu sorgu cümleciği için QS üretmekte oldukça kötü performans göstermiştir. II. Yöntem'in başarısız olmasındaki temel neden tam olarak anlaşılamasa bile veri kümesindeki verilerin tutarsızlıklarından kaynaklandığı düşünülmektedir.

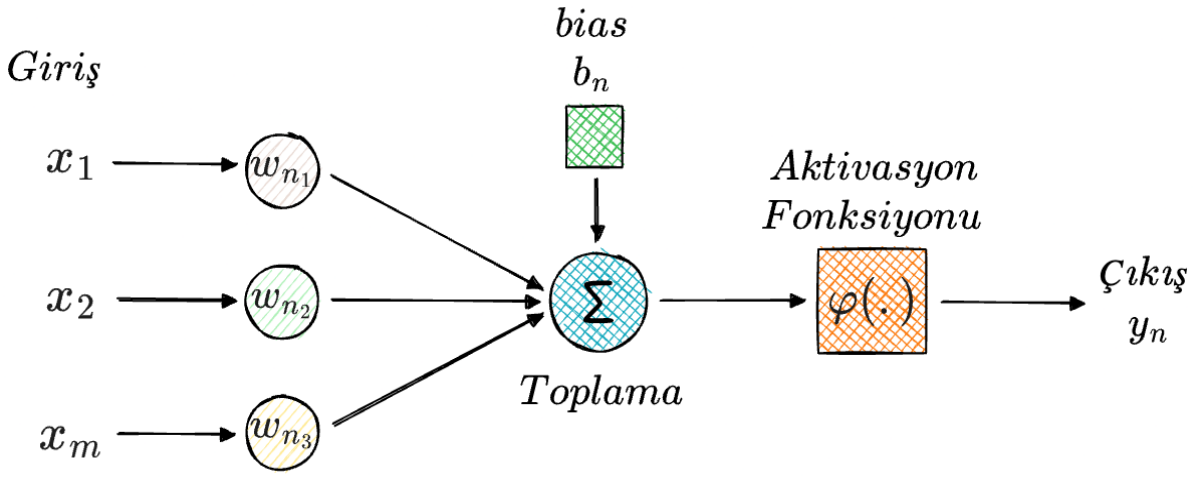
Temel gerçekler, QS alanındaki benzer çalışmalara yol göstermesi için çalışma sonrasında herkesin erişimine açık olarak paylaşılacaktır. Bu çalışmanın literatüre eklemiş olduğu önemli kazanımlardan bir tanesidir. Oluşturulan temel gerçek tanımları, doğrudan ya da geliştirilerek QS alanında yeniden kullanılabilir.

3.5. DERİN ÖĞRENME

Derin öğrenme, yapay sinir ağlarını temel alan makine öğrenmesi yöntemlerinin bir alt kümesi olarak tanımlanabilir. Temel olarak birden fazla işlem katmanından oluşan hesaplama modellerinin, veri temsillerini arkasında yatan derin anlamları öğrenmesine imkân sağlar [51]. Derin öğrenme, görüntü, video ve metin gibi çeşitli yapılandırılmamış verilerle çalışmak için uyarlanmış yapay sinir ağlarına dayanır [52]. Yapay sinir ağları, bir problemi çözmek adına bir araya gelen insan beynindeki nöronlara benzeyen birçok yapay nörondan meydana gelir. Bu nöronların yapısı basit olarak Şekil 3.7'de gösterilmiştir. Bu nöronlar bir araya gelerek katmanları, katmanlar ise bir araya gelerek yapay sinir ağlarını ve derin öğrenme modellerini oluşturur. Yapay sinir ağlarında bulunan katmanlardaki nöronlar, bir sonraki katmana ait nöronlar ile bağlantı halindedir. Sinir ağının içerisinde ağırlıklarla oluşturulmuş sinyalleri sonraki katmana iletirler ve gelen girdilere bir aktivasyon fonksiyonu uygularlar. Derin

öğrenme, milyonlarca yapay nöronun belirli sayıda katman içerisinde bir araya gelmesiyle oluşur.

Derin öğrenme modelleri, verileri anlamak, öğrenmek ve tahmin etmek için yapay sinir ağlarına göre çok daha fazla sayıda nöron içerir. Derin öğrenme, daha fazla sayıda katman içerirken bu katmanların karmaşıklıkları da oldukça yüksektir. Derin öğrenme ismi de bu katmanların sayısının büyüklüğünden ve katmanların ilişkilerinin karmaşıklığından gelmektedir. Bu nedenle yapay sinir ağlarına göre daha karmaşık problemleri çözmek için kullanılır ve daha fazla hesaplama gücüne ihtiyaç duyar.

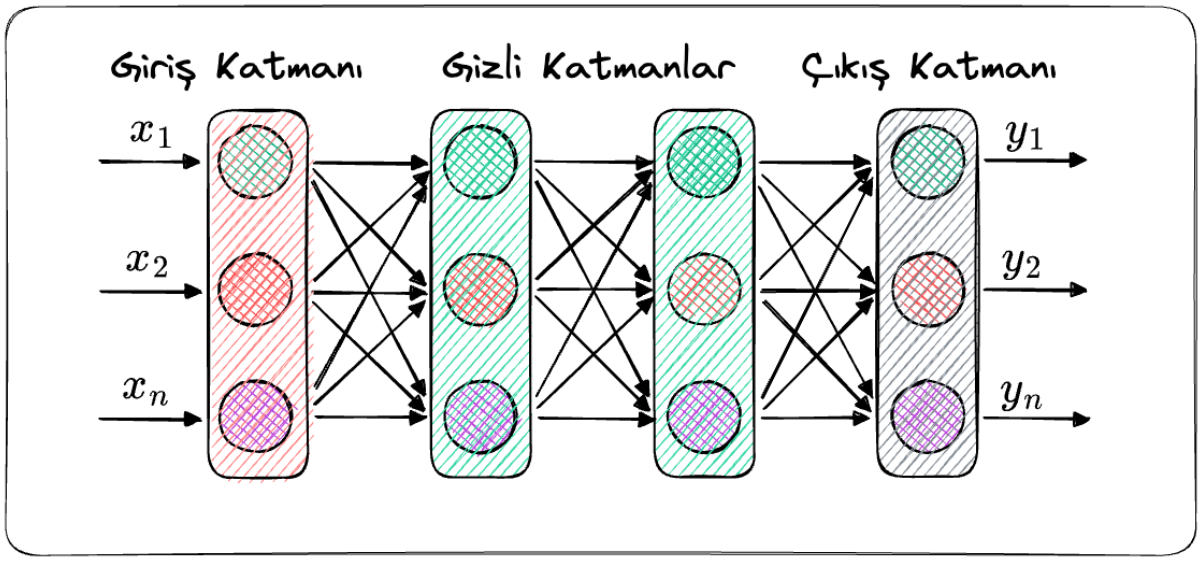


Şekil 3.7: Yapay sinir ağları içerisinde bulunan bir nöron yapısı.

Yapay sinir ağlarının ortaya çıkışı 1950'li yılların sonları olarak gösterilebilir. Bu yıllarda, Frank Rosenblatt tarafından önerilen *Algılayıcı (Perceptron)* [53], basit şablonları öğrenebilen ilk algılayıcı makine modeliydi. Bu gelişmeden sonraki dönemlerde, yapay sinir ağlarının öğrenme zorluğu ve dönemin makinelerindeki hesaplama zorluklarıyla birlikte yapay sinir ağı ile ilgili gelişim de sekteye uğramıştır. 1980'li yıllardan sonra sinir ağlarını verimli şekilde eğitmek için çeşitli yöntemlerin ortaya çıkmasıyla birlikte sinir ağları gelişimi hız kazanmıştır. Yapay sinir ağlarının gelişimiyle beraber, bu sinir ağlarından meydana gelen çok katmanlı mimariler de keşfedilmeye başlanmıştır. Hesaplama gücünün gelişmesi, etiketli büyük veri kümelerinin ortaya çıkması ve sinir ağlarında yeni geliştirilen tekniklerin daha fazla katmana sahip ve daha karmaşık ağlar oluşturmaya olanak tanımasıyla birlikte derin öğrenme kavramı da önem kazanmaya başlamıştır. Derin öğrenme alanı, günümüze gelene dek çeşitli gelişmeler kaydetmiş olsa da gerçek potansiyellerinin ortaya çıkması ve çeşitli problemler için kullanılmaya başlaması son 15 yıl içerisinde gerçekleşmiştir [52]. Özellikle *ImageNet* görüntü veri tabanı tarafından 2012 yılında düzenlenen büyük ölçekli görsel tanıma yarışmasında, *Derin*

CNN (Deep Convolutional Neural Networks) [54], derin öğrenme alanında çığır açan başarısıyla bir dönüm noktası olmuştur.

Derin öğrenme, milyonlarca nörondan ve birden çok *Gizli Katman'dan (Hidden Layer)* oluşan yapısıyla büyük veri kümelerindeki bağlantıları, katmanlar içerisinde ve katmanlar arası adımlarda öğrenerek anlamlı sonuçlara ulaşmak konusunda oldukça başarılıdır. Derin öğrenme mimarisi en basit haliyle Şekil 3.8'de gösterilmiştir. Derin öğrenme mimarileri, içerisinde birden fazla yapay sinir ağı katmanını barındırdığı için bunlara oldukça benzer özellikler taşımaktadır. Yapay sinir ağları genellikle bir tane giriş, bir tane çıkış ve bir tane gizli katmandan meydana gelirken; derin öğrenme tabanlı mimariler bir tane giriş, bir tane çıkış ve n tane gizli katmandan meydana gelir. Bu gizli katmanlar derin öğrenme modellerinin, verideki önemli özellikleri ve bağlantıları keşfederek modelin performansını iyileştirmesine yardımcı olmaktadır. Derin öğrenme, günümüzde bilgisayarlı görme [55], doğal dil işleme [56], otonom sürüş [57] ve konuşma tanıma [55] gibi çeşitli alanlarda kullanılmaktadır.



Şekil 3.8: Birden fazla gizli katman içeren bir derin öğrenme mimarisi.

Bir derin öğrenme algoritmasının öğrenme süreci, içerdiği sinir ağlarının kullanılacak hedef verilerle beslenmesini ve tahmin edilen ile gerçek hedefler arasındaki farkı en aza indirmek için çeşitli optimizasyon algoritmalarının [58] kullanılarak nöronların öğrenme için ağırlıklarını yinelemeli olarak değiştirmesini içerir. Veriler, nöronlar arasında iletilerek öğrenme adımında meydana gelen hata hesaplanır ve tahmin doğruluğunu artırmak için ağırlıklar çıkış katmanından giriş katmanına doğru giderek hesaplanır. Mimari olarak birçok farlı derin öğrenme türü bulunur. CNN, RNN ve TA bu türlere örnek verilebilir. Derin öğrenme,

sunduğu özellikler ve veri içerisindeki derin anlam ve karmaşık ilişkileri bulmak konusundaki başarısına karşın, büyük miktarda etiketli veri ihtiyacı, yüksek hesaplama gücü ihtiyacı ve verilerin aşırı derecede özelleşmesine bağlı olarak görünmeyen veriler üzerinde düşük performans gösterebilmesi gibi bazı zorluklara da sahiptir. Ancak derin öğrenme, sürekli olarak ilerlemesi ve farklı alanlarda daha da karmaşık hale gelen sorunları çözme potansiyeli nedeniyle gelecekte de önemini korumaya devam edecektir.

Çeşitli doğal dil işleme görevlerinde ve araştırmalarında derin öğrenme yöntemleri kullanılmaktadır. Derin öğrenmenin yapısı itibariyle metin verilerinin altında yatan anlamları ve bağlantıları yakalama konusunda oldukça iyi bir yöntem olduğu bilinmektedir. Günümüzde derin öğrenmenin kullanıldığı pek çok farklı doğal dil işleme modeli geliştirilmiştir. BERT [42] ve GPT [43] gibi TA modelleri, duygu analizi, soru cevaplama, metin sınıflandırma ve makine çevirisi gibi doğal dil işleme görevlerini yerine getirmede oldukça iyi performans gösterdiğini kanıtlamıştır. Dillerin yapısal benzerliklerini kullanan Word2Vec [59], GloVe [60] gibi kelime yerleştirmeleri ve anlamsal benzerlikleri yakalayan ELMo [61], BERT [42] gibi bağlamsal yerleştirmeler doğal dil işleme görevlerinde önemli bir gelişme olarak görülebilir. Özellikle TA modelleri, kullandıkları dikkat mekanizmasıyla [7] metinlerin bağlamsal bilgilerinin yakalanmasını geliştirmiştir ve dillere ait daha ince ayrıntıları öğrenebilmektedir. Ayrıca büyük veri kümeleri ile daha önceden eğitilmiş hazır modeller, bazı alt görevleri yerine getirmek için ince ayarlamalar yapılarak kullanılabilir. Metin ve görüntüleri birleştiren çok modelli derin öğrenme alanındaki gelişmeler, metin verileri ile görüntü verilerinin birleştirilerek daha derin anlamların çıkarılmasını sağlayan zengin modellerin ortaya çıkmasını sağlamıştır [62].

QS alanında, doğal dil işleme yöntemlerine benzer şekilde kullanıcı sorgularının karmaşık anlamlarının ve kullanıcı amaçlarının anlaşılması için derin öğrenme yöntemleri kullanılmaktadır. Derin öğrenme yöntemleri, kullanıcı amaçlarını ortaya daha kolay çıkarabilmektedir. Büyük miktardaki sorgu günlükleri içerisinden, kullanıcı etkileşimlerini ve karmaşık sorgu kalıplarını öğrenerek daha alakalı sorguları etkili şekilde önerebilmektedir. QS alanında, tekrarlayan sinir ağlarını kullanan birçok çalışma yapılmıştır [6, 21, 63]. Literatürdeki çalışmaların bazıları tekrarlayan sinir ağlarının yanı sıra *Uzun-kısa süreli bellek (LSTM)* modellerini de kullanmıştır [5, 41]. Büyük veri kümeleri ile önceden eğitilmiş olan ve doğal dil işleme görevlerindeki başarısı bilinen, BERT [42] ve BART [64] dönüştürücülerini kullanarak bu dönüştürücüleri QS alt kümesinde uygulayan çalışma, dönüştürücü ve hiyerarşik dönüştürücü [65] gibi farklı yöntemlerle önceden eğitilmiş TA modellerinin başarısını

kıyaslamaktadır [2]. Çoklu model yaklaşımları, kullanıcı bağlamı, zaman ve tıklama gibi verilerle metin verilerini birleştiren derin öğrenme mimarilerini kullanmaktadır. Bu yaklaşımlar, daha bağlam duyarlı ve kişiselleştirilmiş sorgular sunmayı amaçlamaktadır [5]. QS alanında da derin öğrenmenin doğal dil işleme alanındaki kullanımına benzer şekilde dikkat mekanizmasına sahip sinir ağları daha iyi performans göstermektedir.

TA modelleri, sorguların anlamlarını, ince detaylarını ve birbirleri ile aralarındaki ilişkileri anlayabilmektedir. Derin öğrenme modelleri, sorgu anlamlarını ve kullanıcı amaçlarını anlamada önemli gelişmeler sağlar. Ayrıca kullanıcının geçmiş davranış ve tercihlerine göre arama önerilerini kişileştirilebilmektedir. Derin öğrenme modelleri, anlamı belirsiz ve uzun kuyruklu sorguların altında yatan anlamları da yakalayarak başarılı önerilerde bulunabilmektedir. Derin öğrenme, büyük sorgu günlüklerini kullandığı için çok farklı ve çok fazla sayıda kullanıcı etkileşimini yakalayarak öğrenebilmektedir ve performansını yinelemeli olarak iyileştirebilmektedir. Ayrıca model kendisini, sürekli değişen verilerle birlikte yeni eğilimlere karşı uyarlayabilmektedir. Sonuç olarak, derin öğrenme mimarileri, kullanıldığı birçok farklı alanda olduğu gibi QS alanında da ciddi iyileştirmelere olanak tanımaktadır.

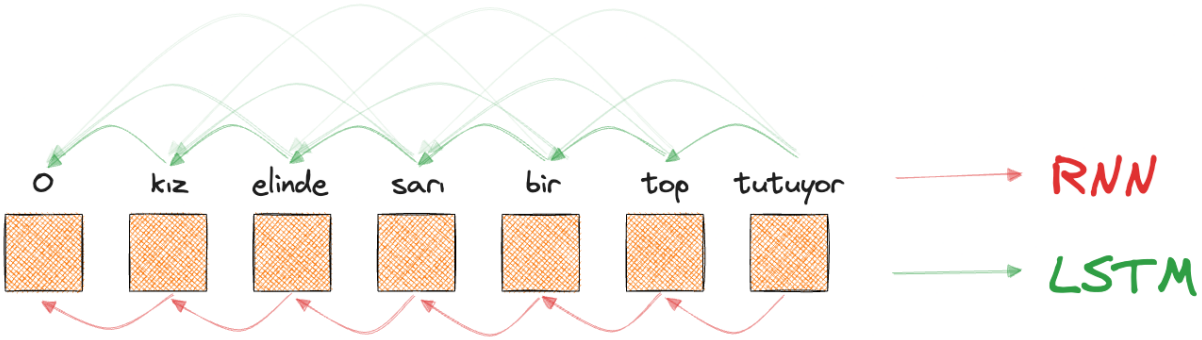
3.6. ÖNERİLEN DERİN ÖĞRENME MİMARİSİ

Derin öğrenme mimarileri tarihsel olarak sürekli bir gelişim içerisinde olmuşlardır. Çok katmanlı algılayıcılar, ardından tekrarlayan sinir ağları ve hemen sonrasında ise uzun kısa süreli bellekler geliştirilmiştir. Yakın tarihte ise derin öğrenme ile ilgili devrim yaratan ilk TA ortaya çıkmıştır [7]. Derin öğrenme teknolojilerin varlığı, doğal dil işleme ile ilgili görevlerde büyük bir ilerlemeye katkıda bulunmuştur. Metin işleme ve doğal dil işleme süreçlerinde, tekrarlayan sinir ağları ve uzun kısa süreli bellekleri kullanan bazı uygulamalar sunulmuştur [66, 67]. Fakat bu mimarilerin sadece doğal dil işleme süreçlerine özgü olmayan bazı temel problemleri bulunmaktadır. Bu problemler şu şekilde özetlenebilir:

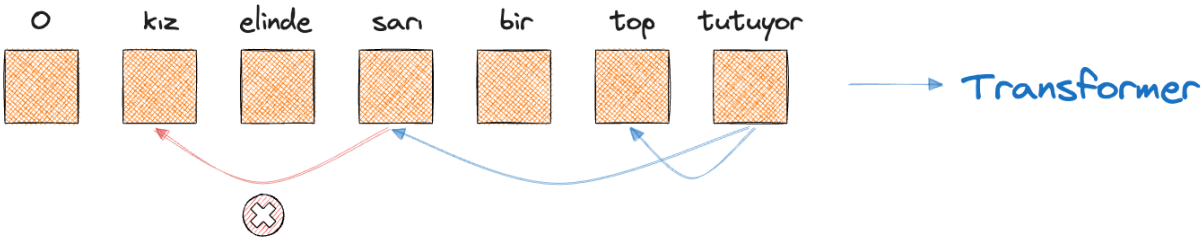
- Sıralı hesaplamalar kullanıldığı için paralel çalışması mümkün değildir bu durum performans ve zaman açısından önemli bir probleme neden olmaktadır.
- Sıralı hesaplamalar esnasında, ardışık dizi elamanlarının arasındaki mesafe arttıkça önceki elemanlara ait bilgilere erişim zorlaşmaktadır.

Doğal dil işleme veya metin işleme görevleri genellikle bir dilden dile dönüşümü veya sonraki cümle tahmini gibi süreçleri içermektedir. Bu yüzden sıralı dizilerdeki bir önceki ya da

birkaç adım önceki kelimeler model için o an hesaplanan kelimenin anlamını tam olarak anlayabilmek adına önemli olabilmektedir. Derin öğrenme yöntemlerinden RNN ve LSTM mimarileri için anlatılan bu duruma ait örnek Şekil 3.9'da gösterilmiştir. RNN ve LSTM mimarilerinde yaşanan bu zorlukların üstesinden gelmek için dikkat mekanizmalarını kullanan TA modelleri ortaya çıkmıştır. TA kullanan derin öğrenme yöntemleri, RNN ve LSTM mimarilerinin yaşadığı belirtilen problemleri ortadan kaldırmaktadır ve verileri paralel olarak işleyebildiği için bu mimarilere göre hız açısından daha iyi performans sergilemektedir. Ayrıca barındırdıkları dikkat mekanizması, işlenen kelimeyle dizi içerisinde ilişkili olabilecek tüm kelimeleri kullanabilmelerini sağlamaktadır. Bir dizinin tüm verilerine ait bilgileri erişim sağlayabilmektedir. Fakat bu erişim istenildiği takdirde kısıtlanabilmektedir. Belirtilen dikkat mekanizması sayesinde sıralı dizilerin içerisindeki önceki kelimelere ait gizli anlamlar, daha sonraki adımlarda kullanılabilmektedir. Bunun arkasında yatan fikir, bir kelime dizisinin herhangi bir noktasındaki kelime ile bu kelimenin öncesindeki ya da sonrasındaki herhangi bir kelimenin ilişkisinin olabilmesi durumudur. TA modelinin, dikkat mekanizmasını kullanarak bir dizi üzerinde sağladığı bilgi erişimi Şekil 3.10'da gösterilmiştir.



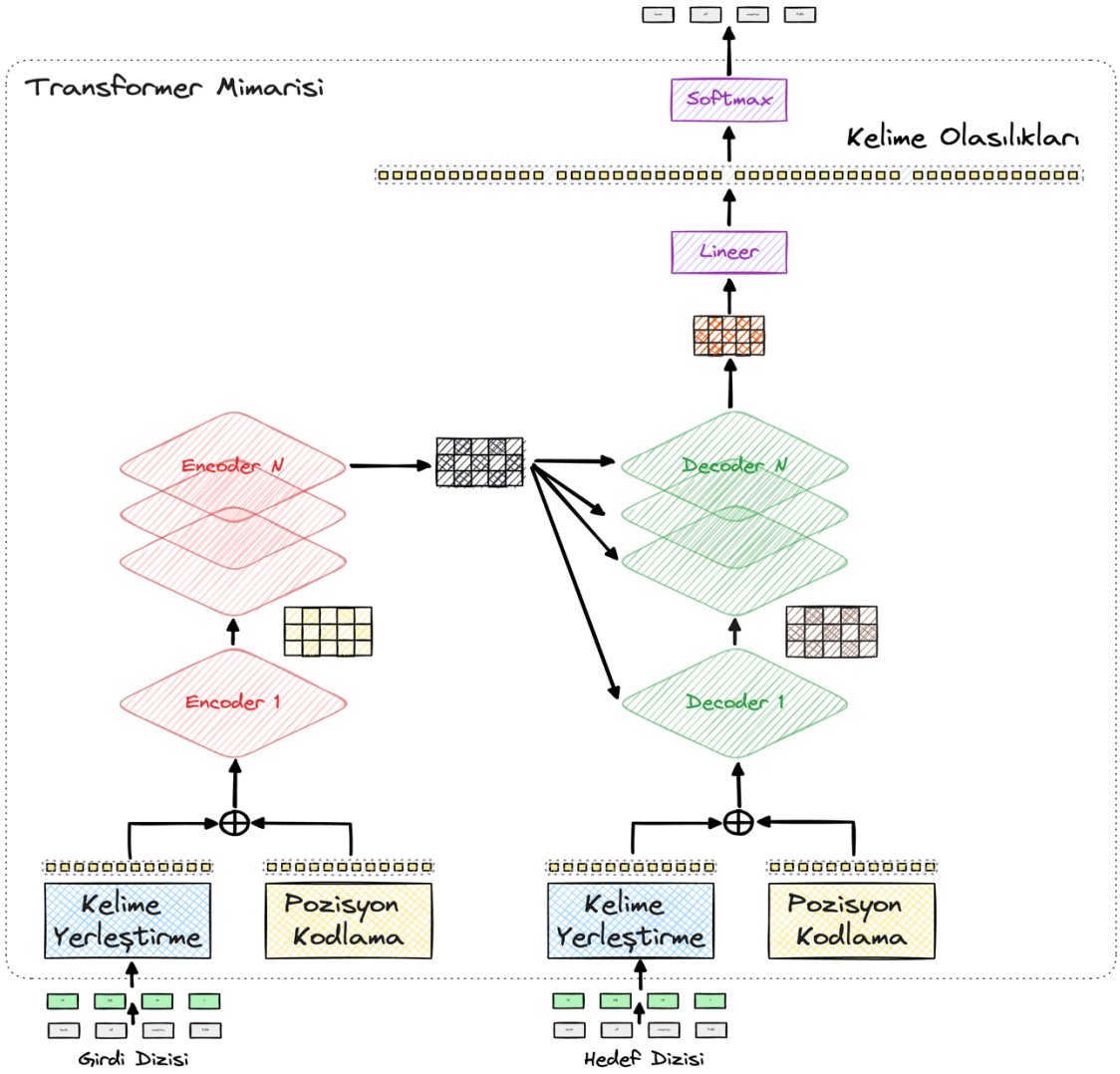
Şekil 3.9: RNN ve LSTM mimarilerinde kelimeler arası ilişkilerin yakalanması.



Şekil 3.10: TA modellerinde kelimeler arası ilişkilerin yakalanması.

Bu çalışmada, problem tanımı da göz önünde bulundurularak aramaların altında yatan anlamların yakalanabilmesi, kullanıcı oturumlarına göre hemen ardından gelebilecek sorguların öğrenilmesi sağlayacak bir TA modeli oluşturmak amaçlanmıştır. Bu öğrenim aşamasından sonra kullanıcılar için daha kaliteli QS oluşturulması hedeflenmiştir. Kullanıcılar için QS

hazırlanırken, QS görevine özgü bir TA olmasına ve hesaplama maliyetleri açısından verimli olmasına da önem verilmiştir. Oluşturulan TA modelinin genel mimarisi Şekil 3.11'de gösterilmiştir. Önerilen TQS mimarisine ait bileşenler ilerleyen kısımlarda daha detaylı olarak ele alınacaktır.



Şekil 3.11: Derin öğrenmeyi kullanan TA modeli.

TA'yı kullanan önerilen derin öğrenme mimarisi, ilk aşamada modelin öğrenmesi için bir giriş kelime dizisi alır. Burada önerilen modelin girdisi veri kümesinde bahsedilen işlenmiş AOL sorgu kümesine ait kullanıcı oturumlarındaki sorgulardır. Kullanıcıların sorgu cümlecikleri, veri kümesindeki verileri kullanarak hazırlanan *Kelime Dağarcığı (Vocabulary)* yardımıyla sayısal bir gösterim ile ifade edilir. TA giriş kısmında, bir yerleştirme ve *Pozisyon Kodlama (Positional Encoding)* katmanı bulunur. Yerleştirme katmanında, sayısal bir gösterim ile ifade edilen sorgular için yerleştirme vektörleri oluşturulur. Yerleştirme katmanı,

kelimelerin geniş bir sunumunu elde ederek kelime anlamlarını yakalamaya yardımcı olur. TA için, RNN mimarilerindeki ardışık ilerlemeden farklı olarak paralel ilerleme söz konusu olduğu için dizideki her bir kelimenin konum bilgisi kaybolmaktadır. Bu bilginin, her bir kelimeye tekrar eklenmesi için pozisyon kodlama katmanı kullanılır. Bahsedilen iki katman TA modelinin hem kodlayıcı hem de kod çözücü kısımlarında yer almaktadır. Bu iki katmandaki işlemlerin ardından vektör olarak ifade edilen giriş sorgu cümlecikleri, kodlayıcı katmanına beslenir. Bu katmanda farklı sayıda kodlayıcı bulunabilmektedir. İlk kodlayıcı, giriş değerlerini bir önceki katmanda bulunan kelime yerleştirme ve pozisyon kodlayıcıdan almaktadır. Diğer kodlayıcılar ise, bir önceki kodlayıcının çıkışından beslenirler. Kodlayıcılar, içerisindeki sinir ağlarına ait nöronlar ve dikkat mekanizmaları yardımıyla girdi sorgu cümleciklerinin derin anlam ve ilişkileri öğrenmeye başlar. Son kodlayıcı, çıktısını kod çözme katmanına iletir. Bu adımdan sonrasında TA modelinin ikinci kısmı olan kod çözücü devreye girer.

Kod çözücü katmanı, kodlayıcı katmanına oldukça benzer bir yapıya sahiptir. Kod çözücü, kodlayıcıya giriş olarak verilen kelime dizisine karşılık, tahmin edilmesi gereken hedef dizisini alarak kodlayıcıda olduğu gibi yerleştirme ve pozisyon kodlamayı uygular. İlk kod çözücü kodlayıcıya benzer şekilde girdisini, yerleştirme ve pozisyon kodlama katmanından alır. Kalan kod çözücüler, girdilerini bir önceki kod çözücünün çıkışından alırlar. Kod çözücülerin içerisinde yapısal olarak kodlayıcılardan farklı çalışmasını sağlayan bazı farklı dikkat mekanizmaları mevcuttur. Bu dikkat mekanizmaları TA modelinin hem kodlayıcıdan gelen girdileri hem de kod çözücülerden gelen girdileri kullanarak öğrenmesini sağlar. Son kod çözücünün ardından çıktı vektörü, *Doğrusal Katman (Linear Layer)* yardımıyla tüm hedef sözlüğünün içindeki kelimeler için bir puan hesaplaması yapar. *Softmax* katmanı, bu puanları olasılık değerlerine dönüştürür. Daha sonra en yüksek olasılığa sahip kelimeler hedef önerileri olarak model çıkışında sunulur. Eğitim sırasında üretilen çıktılarla hedef dizi kelimelerini karşılaştırmak için *Kayıp Fonksiyonları (Loss Function)* kullanılır. Bu kayıp fonksiyonu, TA modelini *Geriye yayılım (Backpropagation)* yardımıyla eğitmek için kullanılır.

3.7. TQS

Derin öğrenme yöntemleri, pek çok metin problemini çözmek için oldukça yaygın bir kullanım ağına sahiptir. Çok çeşitli alanlardaki dil problemlerini çözmek için derin öğrenme yöntemleri son zamanlarda popülerlik kazanmaya başlamıştır. TA kullanılan derin öğrenme yöntemleri, son yıllarda ortaya çıkan ve birçok yapay zekâ görevi için kullanılabilirliğini

kanıtlamış oldukça önemli bir yöntemdir. Dönüştürücüleri ortaya koyan ilk çalışma [7], bu çalışmaya benzer şekilde doğrudan metin verilerini kullanmıştır. Metin verileri için; RNN, CNN ve LSTM gibi yöntemlere göre önemli derecede iyileştirmeyi ve hız artışını beraber sunmaktadır. Çalışmada sunulan yöntem, diğer QS çalışmalarına benzer şekilde kullanıcıya ait bir oturumdaki bir sonraki sorgu cümleciğini tahmin etmeye odaklanmıştır.

Şimdiye kadar TA kullanarak yapılan QS sistemleri, ya BERT, BART, T5 [68] gibi önceden eğitilmiş TA modellerini, orijinal çalışmadaki TA modelinin bir benzerini ya da çok katmanlı bir hiyerarşik TA modelini kullanmıştır. Önceden eğitilmiş modeller, genellikle çok yönlü kullanım için oluşturulmuş modellerdir. Spesifik bir görev için üretilmedikleri için performansları bazı görev özel durumlar için sınırlı kalabilmektedir. Bunun bir sebebi, modelin eğitilmiş olduğu veri kümesidir. Veri kümesi hangi konu ile ilgili ise o konuya özgü anlamlara sahip olacaktır. Ayrıca bu tip önceden eğitilmiş modellerin, göreve özgü olarak uyarlanması için çeşitli ayarları yapmak gerekir. Bazı görevler, önceden eğitilmiş TA modelleri yerine göreve özgü veri kümelerini kullanarak eğitilen dönüştürücülerle daha iyi performans gösterebilmektedir. Bu çalışmada, QS görevine özgü olarak orijinal çalışmada önerilen TA [7] ele alınarak göreve özgü olarak iyileştirilmeye ve geliştirilmeye çalışılmıştır. Ayrıca önerilen TQS modeli, QS problemine özel olarak işlenmiş AOL sorgu günlükleri ile eğitilmiştir. Bunun için literatürde farklı alanlarda, TA ve bileşenleri için yapılan çalışmalara ve bu çalışmalarda kullanılan yenilikçi yöntemlere odaklanılmıştır. Sonuç olarak çalışma, dönüştürücü bileşenleri için farklı yöntemleri ele alarak değerlendirmiş ve bu değerlendirmeler sonucunda QS görevi için hazırlanan TQS'nin performansını artıracak yöntemleri uygulamaktadır.

Temel çizgilerde kullanılan işlenmiş AOL sorgu kümesi; eğitim, doğrulama ve test veri kümeleri şeklinde bölünmüştür. Bu veri kümesindeki veriler, bir sonraki kullanıcı sorgusunun tahmin edilmeye çalışılmasından ötürü, oturumlara ayrılmıştır ve kullanıcı oturumlarındaki aramanın hemen ardından gelecek sorgular belirlenmiştir. AOL sorgu günlükleri Mart-Mayıs 2006 tarihleri arasındaki verileri içermektedir. Mart ve nisan ayına ait veriler eğitim kümesini oluştururken, mayıs ayına ait veriler doğrulama ve test veri kümesi olarak kullanılmıştır. Modeli oluşturmak için kullanılan eğitim, doğrulama ve başarıyı ölçmek için kullanılan test veri kümelerindeki oturum sayılarından veri kümesi bölümünde bahsedilmiştir. İşlenmiş veri kümesinde bulunan özel karakter ifadeleri önceki çalışmalardan farklı etiketler kullanılarak belirtilmiştir. Bunun için ilgili çalışmanın kaynak kodlarında küçük bir değişikliğe gidilerek eğitim, doğrulama ve test verilerinin oluşturulması sağlanmıştır.

TA modelleri, doğaları gereği sayısal verileri kullanarak öğrenme işlemini gerçekleştirmektedir. TQS modelinin girişinde bulunacak sorgu cümleciklerinin de sayısal gösterimlere çevrilerek ifade edilmesi gerekir. Model için eğitim kümesinde bulunan ve en çok tekrar eden 90K sorgu kelimesi, kelime dağarcığını oluşturmak için kullanılmıştır. Bu sözlüğe bazı özel ifadeler de eklenmiştir. <*sos*> ifadesi bir kullanıcı oturumunun başlangıcını belirtmektedir. <*eos*> ifadesi bir kullanıcı oturumunun sonunu ifade eder. <*blank*> ifadesi bir oturumundaki her bir sorgu aralığını ifade eder. <*unk*> ise bilinmeyen bir kelime ile karşılaşması durumunda modelin sayısal gösterim için kullanacağı ifadedir. Oluşturulan kelime dağarcığı, veri kümesinde en sık kullanılan sorgu kelimelerini ve onlara karşılık gelen sayısal değerlerini içerir. Aynı şekilde tam tersi olan her sayısal değere karşılık gelen benzersiz bir sorgu kelimesini de ifade eder. Bu kelime dağarcığı, modelin sayısal verilerin yardımıyla öğrenmesini ve daha sonra sayısal değerlerden hedef QS dizilerinin üretilmesini sağlayacaktır. Bazı sorgu kelimeleri ve bu kelimelere ait sayısal gösterimler Şekil 3.12'de gösterilmiştir. Önerilen TQS modeline giriş olarak verilen bir kullanıcı oturumu aşağıda gösterilmiştir.

S1 = <*sos*> build website <*blank*> easy website design <*eos*>

Sunulan TA modelinin ilk bileşeni olan yerleştirme katmanı, doğal dil işleme görevlerinde olduğu gibi modelin girdisi olan kelimelerin (sorgu cümleciklerinin) altında yatan anlamları ifade etmek için kullanılır. Yerleştirme katmanı, kelimelerin belirli uzunluktaki vektörler ile ifade edilmesini sağlar. Birçok farklı şekilde kelime yerleştirmelerini uygulamak mümkündür. Model için buradaki uzunluk çeşitli denemelerin ardından 1024 olarak seçilmiştir. Yani her bir girdi için oluşturulan vektör uzunluğu 1024 olacaktır. Bu vektörler daha sonra modelin önerilecek sorguları öğrenebilmesi sırasında kullanılacaktır.

Modelin bir diğer bileşeni olan konumsal kodlama, RNN türü mimarilerden farklı olarak TA modelinin paralel öğrenme sürecinde ihtiyaç duyulan bir bileşendir. Bu bileşen, RNN gibi mimarilere göre ciddi bir avantaj sağlamaktadır. TA, herhangi bir RNN veya CNN yapısı içermediği için paralel öğrenme sürecinde konum bilgisinin kaybolmasını engellemek adına konumsal kodlamayı kullanarak dizinin ögelerine ait konum bilgisini ekler. Konumsal kodlama için *Göreceli (Relative)* konum ya da *Mutlak (Absolute)* konum bilgisi kullanılabilmektedir [69]. Mutlak konum bilgisi, her bir sırayı farklı bir yerleştirme vektörü ile ifade ederken, göreceli konum bilgisi dizideki farklı mesafede bulunan ögeler arasındaki ilişkileri yakalayarak göreceli bir konum bilgisini ekler. Dönüştürücüleri tanıtan ilk çalışma [7], konumsal kodlama

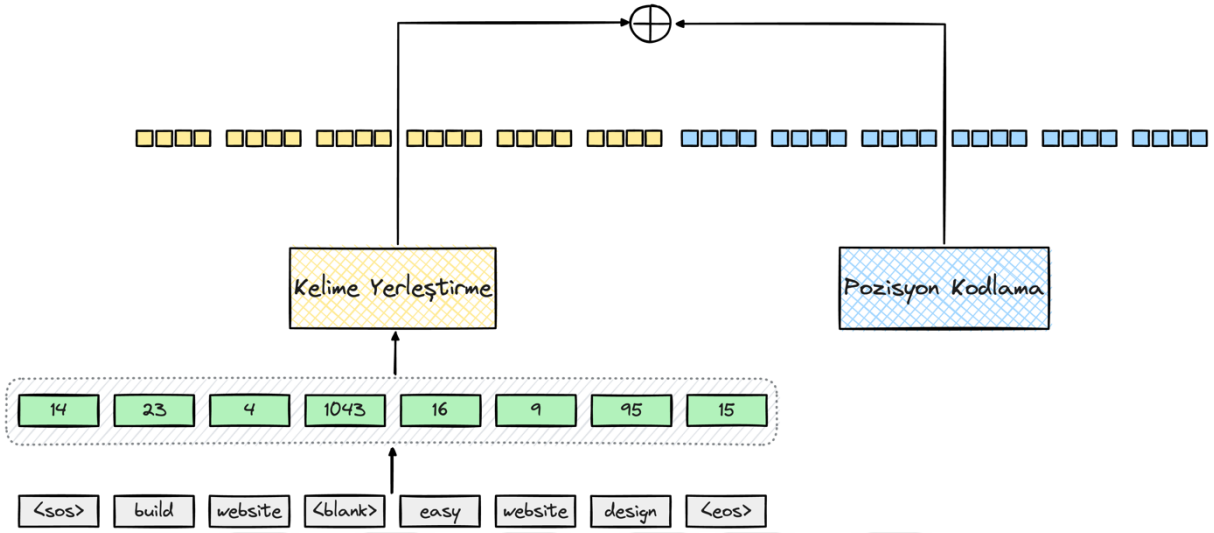
için mutlak konum bilgisini kullanmıştır. Göreceli konumsal kodlama için sunulan yöntem [70], bu çalışmada sunulan model kullanılarak test edilmiştir fakat mutlak konum bilgisi kullanıldığında model performansının daha başarılı olduğu gözlemlenmiştir. Mutlak konum bilgisi için *Sinüzoidal konumsal kodlama (Sinusoidal positional encoding)* [7] kullanılmıştır. Konumsal kodlama için kullanılan yöntem, Denklem 3.2'de gösterilmiştir. Denklemde kullanılan d_{model} parametresi, kelime yerleştirme için kullanılan vektör uzunluğunu ifade etmektedir. Bu denklem, konum kodlama için gereken önemli şartları sağlamak için hızlı ve kolay uygulanabilir bir yöntemdir. Bunun nedenin, her bir sorgu oturumunun veri ön işleme adımında belirli bir boyuta kadar indirgenmesi olduğu düşünülmektedir. Ön işleme adımından sonra ortaya çıkan oturumlardaki sorgu sayılarının çok büyük miktarlarda olmaması, mutlak konumları kullanan yöntemin daha başarılı olmasını sağlamıştır. Çünkü göreceli konumların daha uzun boyuttaki dizileri genellemek için daha uygun oldukları bilinmektedir. Sorgu ifadelerinin sayısal ifadelere dönüştürülmesinin ardından, önerilen modelin ilk bileşenleri olan kelime yerleştirme ve konumsal kodlama adımları Şekil 3.13'te gösterilmiştir. Gösterilen üç aşama hem kodlayıcı hem de kod çözücüde ortak olarak kullanılır. Sadece girdi değerlerinde değişiklik olmaktadır.

Kelime	Sayısal Gösterim
zymogenetics	242248
amputate	204550
parfume	71662
sapiens	84936
sonymusicstore	107544
renault	28459
biographies	14981

Şekil 3.12: Kelime dağarcığında bulunan bazı sorgular ve sayısal gösterimleri.

$$\mathrm{x} = \begin{cases} sin\left(\frac{pos}{10000^{i/d_{model}}}\right) & if\ i\ mod\ 2\ =\ 0 \\ cos\left(\frac{pos}{10000^{(2i+1)/d_{model}}}\right) & otherwise \end{cases} \tag{3.2}$$

Pozisyon kodlama, TA modelleri için çok önemli bir bileşendir. RNN ve CNN gibi mimarilerin dizi ögelerinin sırasını öğrenmekte yaşadığı sorunları ortadan kaldırarak TA için benzersiz bir yol sunmaktadır. Bu bileşen, TA için çok sayıda sorgudan oluşan kullanıcı oturumlarındaki sorgu anlamlarının ve bağlamlarının yakalanması için oldukça verimlidir. Ayrıca değişken uzunluktaki dizi işleme yeteneklerine sahiptir.



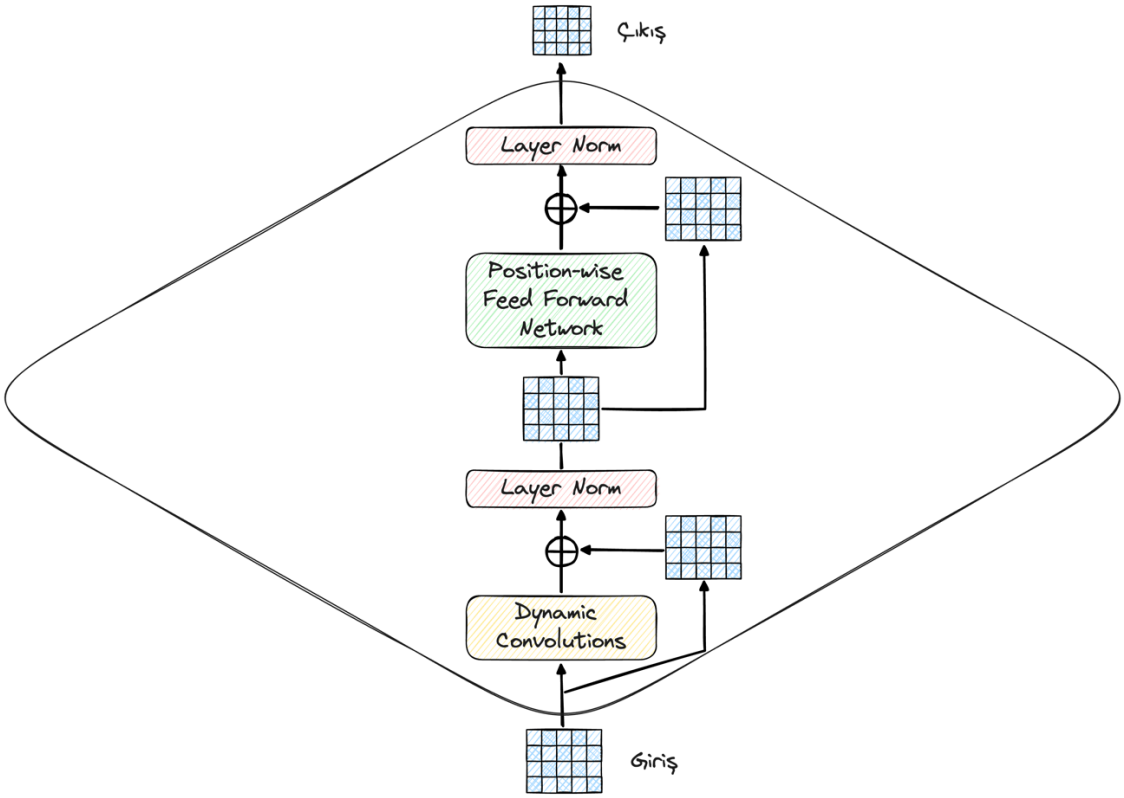
Şekil 3.13: Yerleştirme ve konumsal kodlama adımları.

TQS'nin şimdiye kadar bahsedilen bileşenlerinden üretilen çıktıların iletileceği kısım, kodlayıcılardır. Kodlayıcılar, bir TA için kod çözücülerle birlikte en önemli kısım olarak kabul edilebilirler. TA modelleri, istenilen sayıda kodlayıcı ve kod çözücüye sahip olabilmektedir. Kodlayıcı katman sayısının artırılması modelin performansını ve yeteneklerini doğrudan etkiler. Daha fazla sayıda kodlayıcı katmanının bulunması, modelin veriler üzerindeki hakimiyetini artırmasına yardımcı olur. Bu sayede kelime dizilerinin arasındaki ilişkileri ve kelimelerin altında yatan ince anlamları yakalamada daha iyi performans gösterebilirler. Ayrıca daha derin temsillerin oluşturulmasına yardımcı olarak daha fazla özelliğin keşfedilmesini, daha ince ayrıntıların öğrenilebilmesini ve daha uzun diziler üzerindeki etkinliğinin artırılmasını sağlayabilir. Fakat bu iyileştirmelere ek olarak aşırı uyuma, hesaplama karmaşıklığına ve hesaplama maliyetinin yükselmesine neden olabilir. Bu yüzden model geliştirme sırasında öncelikli olarak orijinal çalışmada [7] sunulan 6 kodlayıcı katmanı ile değerlendirmeler yapılmıştır. Daha sonra katman sayısı 10'a kadar kademeli olarak artırılmıştır. Fakat bu seviyeden sonra hesaplama maliyetlerindeki öngörülemez artış nedeniyle katman sayısı artırılarak daha fazla deneme yapılmamıştır. Kodlayıcı için katman sayısı ise zaman ve hesaplama maliyetleri düşünülerek 8 olarak belirlenmiştir.

Kodlayıcı katmanının içerisinde, bir önceki katmandan gelen çıktılar kullanılarak çeşitli işlemleri gerçekleştirir. İşlemlerin ardından oluşturulan çıktılar, bir sonraki kodlayıcının girdisi olarak kullanılır ve aynı süreç tekrarlanır. Son kodlayıcı katmanında oluşturulan çıktılar, diğer kodlayıcılardan farklı olarak kod çözücüleri beslemek için kullanılır. RNN gibi yöntemler, doğası gereği ardışık ilerledikleri için uzun dizilerde ögeler arası mesafe büyüdükçe bu ögelerin önceki ögeler ile olan ilişkilerini ve anlamlarını kaybetmektedir. Yani bir kelime dizisinin başındaki kelime, dizinin sonundaki kelime için önem taşıyorsa model son adıma geldiğinde ilk kelimeyi unutmuş olabilmektedir. TA modellerinin sunduğu en büyük yenilik bu gibi yöntemlerin kısıtlarından olan ögeler arasındaki ilişkileri kaybetmemesidir. Hem ilişkileri kaybetmemeyi hem de birbirili ile ilişkili kelimelelerin unutulmamasını sağlayan bileşen, dikkat mekanizmalarıdır. Dikkat mekanizmaları, diğer TA modellerinde olduğu gibi TQS'nin de en kritik bileşenlerinden biri olarak tasarlanmıştır.

TQS'e ait standart bir kodlayıcı katmanı Şekil 3.14'te gösterilmiştir. Kodlayıcının içerisinde orijinal çalışmadakinden farklı olarak *Öz Dikkat (Self Attention)* için *Çok-kafalı Dikkat (Multi-head Attention)* yerine *Dinamik Evrişim (Dynamic Convolution)* kullanılmıştır. Dinamik evrişim, çalışma zamanında çok-kafalı dikkate göre %20 kadar daha hızlıdır. Ayrıca bazı makine çevirisi görevlerinde öz dikkate göre daha iyi performans gösterdiği bildirilmiştir [71]. Bu yüzden TQS modeli için çok-kafalı dikkat ve dinamik evrişim mekanizmaları ayrı ayrı denenmiştir. Yapılan değerlendirmeler, dinamik evrişimleri kullanarak eğitilen kodlayıcı modelinin performans açısından ufak bir artış sağladığını göstermektedir. Bunun yanı sıra, modelin çalışma hızını çok-kafalı dikkatin kullanıldığı modele göre belirli ölçüde iyileştirdiği için dinamik evrişimler tercih edilmiştir. Dinamik evrişimler ile öz dikkat mekanizmasının farklılıkları Şekil 3.15 (a-b)'de gösterilmiştir. Önerilen TA için kolay ama önemli bir bileşen olan *Katman Normalleştirme (Layer Normalization)* [72] hem kodlayıcılar hem de kod çözücülerde kullanılmıştır. Burada normalleştirme bileşeni için katman normalleştirilmesinin kullanılmasının temel nedeni doğal dil işleme görevleri için daha önce test edilerek uygunluklarının kanıtlanmış olmasıdır. Alternatif olarak *Toplu Normalleştirme (Batch Normalization)* kullanılabilir fakat bunun doğal dil işleme görevleri için uygun olmadığı düşünülmektedir. Bu katman, kodlayıcı ve kod çözücü için her katmanda gerçekleştirilen işlemlerden sonra katman çıktılarını normalleştirmek için kullanılır. Bunun için katmanın girişi ve katman sonrasında oluşan çıktılar *Bırakma (Dropout)* fonksiyonu kullanılarak toplanır. Model için bırakma, belirli özelliklere aşırı bağımlılığı engellemek ve modelin genelleştirilmiş

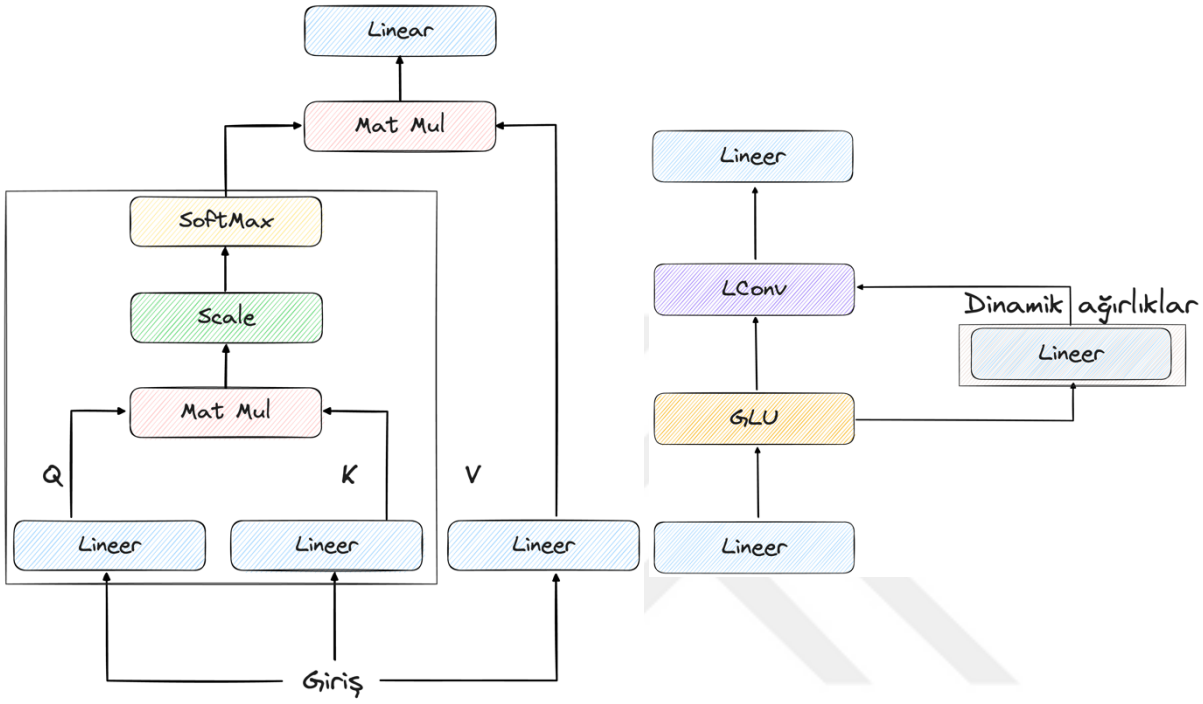
temsilleri öğrenmesini sağlamak için kullanılmıştır. Model üzerinde, bırakma için çeşitli değerler kullanılarak yapılan denemeler sonucunda en iyi sonuçların 0.2 değeri kullanıldığında sağlandığı görülmüştür. Bu nedenle katman normalleştirme sırasında kullanılması için bırakma, 0.2 olarak ayarlanmıştır. Katman normalleştirme ve bunun için kullanılan bırakma hem kodlayıcılar hem kod çözücüler için kullanıldığı tüm noktalarda aynı ayarlar ile yapılandırılmıştır.



Şekil 3.14: TQS'ye ait kodlayıcı katmanı.

TQS'nin hem kodlayıcıların hem de kod çözücülerin bir bileşeni olan *Konumsal İleri Besleme Ağı (Position-wise Feed Forward Network)*, bir kelime dizisindeki her sıra ögesini bağımsız ve aynı şekilde işlediği için bu şekilde isimlendirilmektedir. Bileşen, bir dizi içerisindeki her bir pozisyon için özel olarak çalışır. Bu her bir sıra ögesinin eşit ve bağımsız olarak değerlendirilerek sadece kendi konum bilgilerine göre güncellendiğini gösterir. Ağın, giriş boyutu ile çıktı boyutu aynıdır. Bu durum modelin, dizi uzunluğunu ve konum bilgisini korumasına olanak tanır. Dönüştürücüleri tanıtan çalışmada [7] sunulan bu ağ, doğrusal olmayan bir aktivasyon fonksiyonu olan *ReLU* [73] ile birbirine tam bağlı iki doğrusal katmandan oluşur. Bu bileşenin çıktısını az önce bahsedilen katman normalleştirmesi izler. Katman normalleştirmesi, bir önceki adımdaki çıktıyı ve adımın girdisini beraber kullanarak

bir sonraki adımda bulunan bileşene aktarır. Bu işlemler, tüm normalleştirme katmanları boyunca birbirini takip ederek uygulanır. TQS için konumsal ileri besleme ağındaki gizli katman sayısı kelime yerleştirme boyutunun 4 katı olacak şekilde 4096 olarak ayarlanmıştır. İleri besleme ağına ait ifade, Denklem 3.3'te gösterilmiştir.



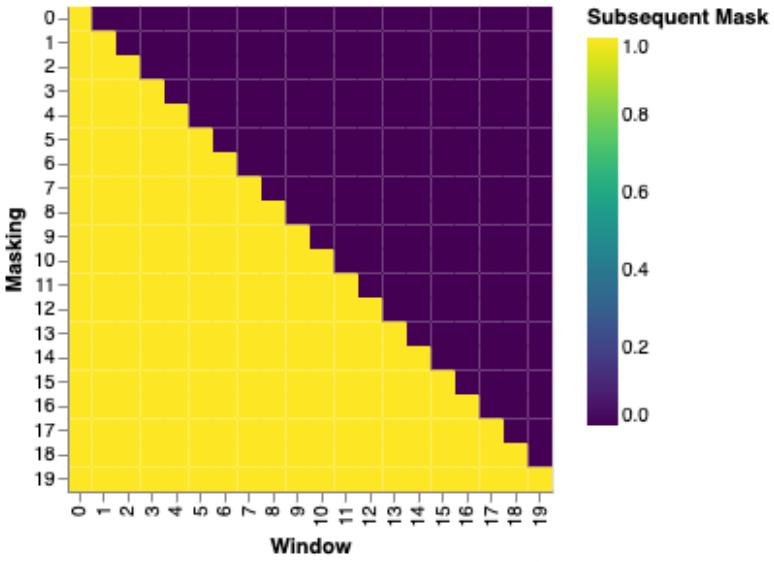
(a) Öz dikkat (b) Dinamik evrişim

Şekil 3.15: Öz dikkat mekanizması ve dinamik evrişim mekanizmalarının gösterimi [71].

$$FFN(x, W_1, W_2, b_1, b_2) = max(0, xW_1 + b_1)W_2 + b_2 \quad (3.3)$$

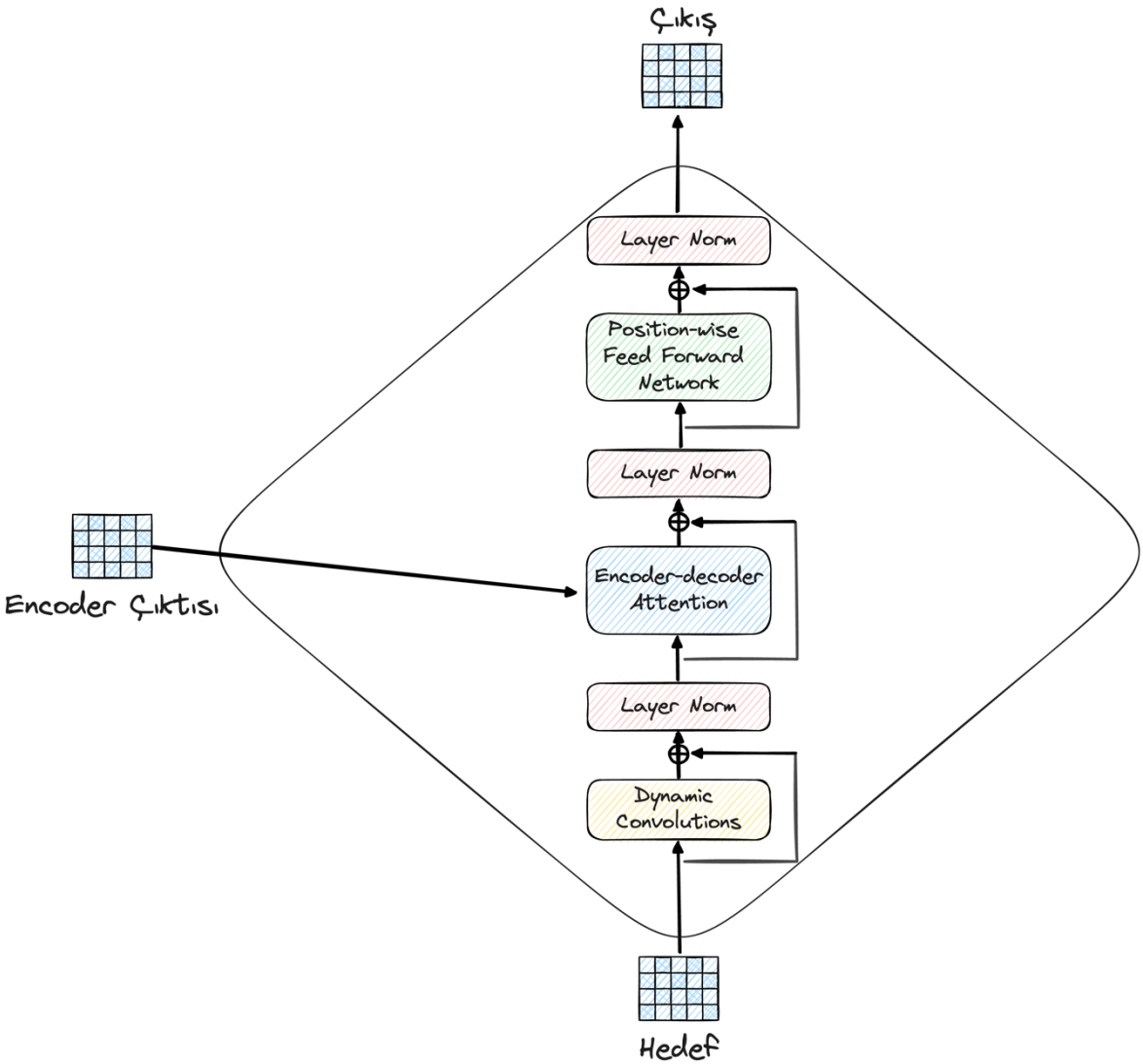
TQS'nin kod çözücülerinin mimarisinin büyük bir kısmı kodlayıcılarla aynıdır. İlk kod çözücünün girdisini üreten bir kelime yerleştirme ve konumsal kodlama katmanı bulunur. Bu katman, hedef QS dizilerini girdi olarak alır ve aynı kodlayıcı da olduğu gibi sorguların anlamları ile konum bilgilerini ifade etmek için işler. Ardından işlenmiş bu bilgiler kodlayıcı akışındakine çok benzer şekilde kod çözücülerin içerisinde ilerler. Fakat bu sürecin ilerleyişinde iki temel farklılık bulunur. Bu farklılıklardan ilki kod çözücünün içerisinde bulunan öz dikkat mekanizmasında meydana gelir. Kod çözücülerdeki dikkat bileşeni için kodlayıcı da olduğu gibi performansı daha önce test edilen dinamik evrişim mekanizması kullanılmıştır. Fakat kullanılan bu dikkat mekanizmasına ayrıca bir maskeleme stratejisi eklenir. Bu strateji, ilgili zaman adımında, sadece zaman adımına kadar görülmüş olan dizi

ögelerinin kullanılmasını sağlar. TQS için maskeleme stratejisinin ana nedeni, hedef QS dizisinin oluşturulması sırasında modelin oturumdaki gelecek sorguları görmesini engellemektir. Bu strateji, gelecek sorgunun tahmin edilmeye çalışıldığı bu görev için oldukça önemlidir. Sadece ilgili konumdaki sorgudan önceki konumlara ait sorgulara erişilebileceğini gösteren maskeleme Şekil 3.16'da gösterilmiştir. Kod çözücüler, oturumdaki kelimeleri işler ve hedef QS dizilerini tahmin etmek için bunları kullanır. Eğitim sırasında kod çözücü, tüm hedef sorgular kullanılarak beslendiği için belirli bir konumdaki sorgu tahmin edilirken tüm hedef sorguları kullanabilir bu da modelin hile yapmasına olanak tanır. Kod çözücülerde maskeleme stratejisi, bu gibi durumları önlemek için kod çözücünün sadece zaman adımına kadar olan hedef QS dizisini görmesine izin verir. Kod çözücülerin, kodlayıcılardan ikinci ve son farklılığı ise ikinci bir dikkat katmanına sahip olmasıdır. Bu katman, kodlayıcıdan gelen bilgilerle kod çözücü tarafından gelen iki bilginin birleştirilerek işlenmesini sağlar. Bu dikkat mekanizması için orijinal metinde açıklanan *Ölçeklendirilmiş Nokta-çarpım Dikkati (Scaled Dot-product Attention)* dikkati kullanılmıştır. TA modelleri, makine çevirisi görevi için *Katkılı Dikkat (Addentive Attention)* [74] olarak sunulan yöntemi de kullanılabilmektedir. Fakat TQS'de kolay uygulanabilirliği ve bilindiği kadarıyla oldukça iyi performans göstermesi nedeniyle ölçeklendirilmiş nokta-çarpım dikkati kullanılmıştır. Kod çözücü bileşenleri, bahsedilen bu iki temel bileşen dışında tamamen aynıdır. Her bir bileşenin sonucuna bir katman normalleştirmesi uygulanır. Son dikkat mekanizması olan ölçeklendirilmiş nokta-çarpım dikkatinin çıktıları konumsal ileri besleme ağını beslemek için kullanılır. Son kod çözücü, çıktıları çıktı bileşenlerine iletir. TQS kod çözücülerinin mimarisi Şekil 3.17'de gösterilmiştir.

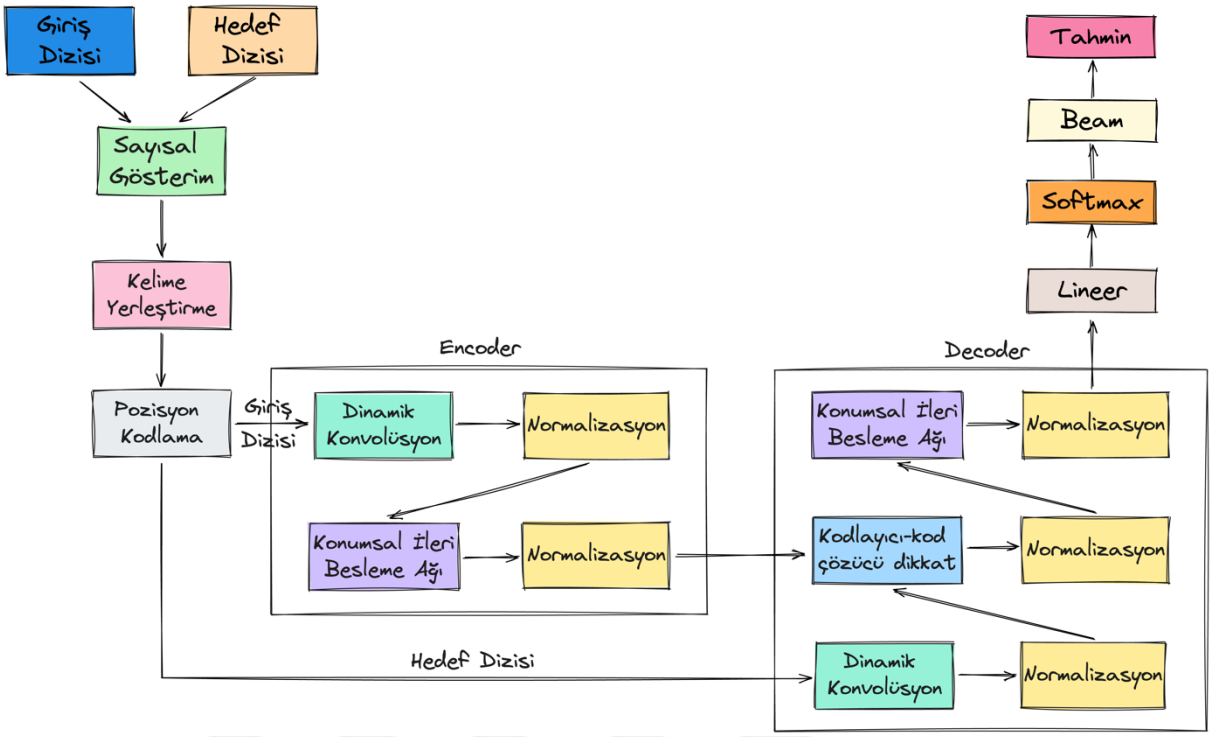


Şekil 3.16: Kod çözücüye ait maskeleme.

Çıktı bileşenleri, bir doğrusal katman ve Softmax fonksiyonundan oluşur. Son kodlayıcıdan gelen çıktılar öncelikle doğrusal katman kullanılarak sorgu dağarcığı boyutuna indirgenir. Ardından Softmax fonksiyonu kullanılarak model tarafından hesaplanan puanların olasılık dağılımlarına göre normalleştirilmesini sağlar. Her pozisyondaki en yüksek puana sahip sorguların sayısal gösterimleri, kelime dağarcığı kullanılarak eşleştirilir ve TQS'nin çıktısı olarak sunulur. TQS modelinin, başlangıç anından modelin tahminde bulunduğu ana kadar ki tüm süreçlerini gösteren aşamaları gösteren akış Şekil 3.18'de gösterilmiştir.



Şekil 3.17: TQS'ye ait kod çözücü katmanı.



Şekil 3.18: TQS modelinin giriş anından tahmin anına kadar olan tüm adımları.

4. BULGULAR

Bu tez çalışmasının ana amacı, QS alanına özgü eğitilmiş bir TA oluşturmaktır. Ayrıca TA modellerinin kanıtlanan başarısı kullanılarak, QS görevine özgü modelin iyileştirilmesi sağlanmaya çalışılmaktadır. Günümüzde QS alanında çeşitli çalışmalar, önceden eğitilmiş dönüştürücüleri veya dönüştürücüleri sunan orijinal çalışmadaki [7] TA modelini neredeyse tamamen aynı şekilde uygulamıştır. Tez çalışmasında önerilen TQS, şimdiye kadar yapılan çalışmalardan farklı olarak, sorgu günlüklerine özel olarak geliştirilmiştir ve bileşenlerinde ana çalışmasından farklı bazı bileşenleri içerisinde barındırır. QS gibi özelleştirilmiş bir görev için QS görevine özgü verilerin kullanılmasının, çok genel veri kümeleri kullanılarak önceden eğitilmiş TA modellerine göre çok daha iyi bir TA modeli oluşturabileceği düşünülmektedir.

AOL sorgu günlükleri kullanarak hazırlanan çalışma, önceki çalışmalarda yer alan çeşitli veri analiz ve ön işleme adımlarından geçirilmiştir. Çalışmada sunulan TQS'nin hedefi, sunulan diğer çalışmalarda olduğu gibi [2, 6] bir kullanıcı oturumundaki sonraki sorguyu tahmin etmektir. Bunun için sorgu günlüklerine kullanıcıların oturum bilgileri eklenmiştir. Daha önce yapılan çalışmadakine benzer şekilde [6] oturum bilgisi, sorgu günlüklerine maksimum 30 dakikalık bekleme süresi kullanarak eklenmiştir.

QS alanının doğasındaki zorluklardan kaynaklı olarak hedef sorguyu belirlemek oldukça güçtür. Bu zorluğun üstesinden gelerek, TA modellerinde kullanılmak üzere QS alanına özel temel gerçekler üretilmiştir. Bunun için 3 farklı yöntem kullanılarak her sorgu için hedef sorguları içeren temel gerçek kümesi oluşturulmuştur. Bu temel gerçek kümesine benzer şekilde aynı yöntemler kullanılarak her bir oturum için de temel gerçekler kümesi oluşturulmuştur. Bu temel gerçek kümesi oluşturulurken ham AOL sorgu günlükleri verisindeki toplam 10.382.084 oturum kullanılmıştır ve her bir oturum sorgusu için 20 adet aday sorgu içeren temel gerçekler belirlenmiştir.

TQS modelini değerlendirmek üzere, üç temel çizgi belirlenmiştir. Bu temel çizgilerden ilki olan *HRED_WP* [6], 2015 yılında ortaya çıkan ve RNN mimarisini kullanarak veri seyrekliğini önleyen ve bağlam duyarlılığına sahip modeldir. HRED nadir ve uzun kuyruklu sorgular için önerilerde bulunabilmektedir. Değerlendirme için kullanılan ikinci temel çizgi

olan *TS_WP* [2], orijinal çalışmada sunulan ve makine çevirisi için kullanılan TA modelini [7], QS görevine özel olarak eğitilmesini sağlamış ve bu eğitilen model ile ortaya çıkan sonuçları incelemiştir. Sonuncu temel çizgi *H_TS_WP* ise, yine aynı çalışmada hazırlanan hiyerarşik TA modelini kullanmaktadır. Hiyerarşik TA [65] kullanarak oluşturulan model, 2 seviyeli bir kodlayıcı içerir. Bunlardan ilki kelime seviyesinde kodlayıcılar diğeri ise sorgu seviyesinde kodlayıcılardır. TQS modeli, bu üç temel çizgi kullanılarak değerlendirilmektedir. Modelleri, Perplexity metriği açısından değerlendirebilmek için kelime dağarcığındaki kelime miktarlarının aynı olması gerekir. Çünkü modelin ürettiği QS için ürettiği bu kelime grubuna ne kadar şaşırdığını (veya kafasının karıştığını) gösterir. Bu sebeple tüm modeller için aynı tokenizer kullanılmıştır. Tokenizer olarak ise bir önceki çalışmada da yer alan *Word Piece Tokenizer* seçilmiştir [2].

TQS modelinin çıktıları, bir olasılık dağılımına göre üretilen puanları içermektedir. Modelin ürettiği bu puanlar kullanılarak kullanıcı için en kaliteli sorgular seçilerek önerilmektedir. Burada bilinen iki önemli yaklaşım mevcuttur: *Açgözlü Kod Çözme (Greedy Decoding)* ve *Işın Araması (Beam Search)* [75]. Açgözlü kod çözme performans için oldukça hızlı bir yöntemdir ve hesaplama maliyetleri düşüktür. Temel olarak hedef QS dizisindeki her bir öge için en yüksek puana sahip ögenin seçilerek önerilmesini sağlar. Işın araması, her QS adımında en olası K adet sorguyu alarak birden fazla olası QS dizisini araştırır. Araştırılan bu diziler paralel olarak işlenerek sabit bir sayıda öneri seçilir. Bu yöntem hesaplama maliyetleri ve hız açısından daha verimsizdir fakat daha kaliteli QS sunmaktadır. Bu yüzden çalışmada değerlendirilen metrikler, TQS modelinin çıktılarının ışın araması kullanılarak oluşturulan sonuçlarını içerir.

QS alanında bir modelin performansını ve başarısını ölçmek için genel ve kaliteli bir değerlendirme metriği yoktur. QS sonuçlarının kalitesini değerlendirmek sorunludur çünkü bir kullanıcı oturumu için çok sayıda QS üretilebilir. TQS'yi değerlendirmek için önceki çalışmalarda kullanılan değerlendirme metriklerinden yola çıkarak sistemin kalitesi ölçülmeye çalışılmıştır. Değerlendirme için *Kafa Karışıklığı (Perplexity)* ve *BLEU* [76] metrikleri kullanılmıştır. Tüm temel çizgiler ve TQS oturumlardaki sorgular üzerinde olasılık dağılımları üretirler. Kafa karışıklığı bir modelin, gerçek değeri ne kadar iyi tahmin ettiğinin bir ölçüsü gibidir. Bu çalışmadaki QS için, kelime dağarcığında birçok sorgu kelimesi mevcuttur. Ayrıca modelin tahminde bulunduğu birçok sorgu kelimesi de mevcuttur. Eğitim aşamasının hemen ardından test verileri kullanılarak modelin bu veriler için bir tahminde bulunması sağlanır.

Sonrasında her tahmin gerçek sorgu önerisi ile karşılaştırılır. Model kendine çok güvendiğinde, doğru kelimelere yüksek olasılıklar verecektir. Bu nedenle kafa karışıklığı düşük olacaktır. Ancak model doğru kelimeyle ilgili net bir tahminde bulunamadığında ve belirsizlikler mevcut olduğunda kafa karışıklığı yüksek olacaktır. Bu nedenle daha düşük kafa karışıklığı değerleri her zaman daha iyi sonuçların olduğunu aksinin ise daha kötü sonuçları belirttiğini gösterir. Değerlendirme için kullanılan bir diğer metrik olan BLEU, hedef QS kelimelerinin model tarafından tahmin edilen QS kelimeleri ile ne kadar eşleştiğini n-gramlar ile kontrol eder. BLEU-1, BLEU-2 ve BLEU-3 sırasıyla oturumlardaki sorguların 1-gram, 2-gram ve 3-gramları şeklindeki formlarını ifade etmektedir.

Temel çizgiler ve önerilen TQS modeline ait BLEU puanları Tablo 4.1'de gösterilmiştir. Bu modellere ait BLEU puanlarını gösteren grafik ise Şekil 4.1'de verilmiştir. Sonuçlar detaylara inilmeden yüzeysel olarak incelendiğinde QS için TA modellerinin oldukça iyi performans sunduğunu kanıtlamaktadır. Temel çizgilerde değerlendirilen TA modelleri ve önerilen TQS modeli, RNN tabanlı HRED_WP'yi geride bırakmıştır. Bunun ana nedeninin, TA modelinin kullandığı ve çalışma boyunca bahsedilen dikkat mekanizmaları olduğu düşünülmektedir. Dikkat mekanizmaları, RNN ve LSTM gibi yöntemlerin, dizi uzunlukları arttıkça hatırlamakta güçlük çektiği önceki dizi ögelerini hatırlamakta oldukça başarılıdır. Ayrıca tüm modeller için hedef dizilerindeki incelenen öge sayısı arttıkça (n-gramlar için n sayısının artması), BLEU puanlarının düşme eğiliminde olduğu görülmektedir. Bu da genel bakış açısına göre beklenen bir davranıştır. Çünkü modeller, daha fazla tahminde bulundukça QS, bir oturumdaki doğru değerlerden giderek uzaklaşma eğiliminde olacaktır.

Tablo 4.1: Temel çizgilere ve TQS'ye ait BLEU puanları.

	HRED_WP	**TS_WP**	**H_TS_WP**	**TQS**
BLEU-1	0.422	0.454	**0.493**	0.473
BLEU-2	0.135	0.178	**0.238**	0.212
BLEU-3	0.059	0.089	**0.146**	0.116

BLEU puanları daha detaylı olarak incelendiğinde, önerilen TQS modelinin TS_WP temel çizgisini geride bıraktığı görülmektedir. Bu beklenen bir durumdur çünkü TQS, orijinal çalışmada sunulan TA modeline ait bazı bileşenleri QS için daha iyi sonuç verebileceği düşünülen bileşenler ile değiştirmiştir. Ayrıca orijinal çalışmayı kullanan TS_WP modeli 6 kodlayıcı ve kod çözücü katmanından yararlanır. TQS, kodlayıcılar ve kod çözücüler için 8

katman kullanmaktadır. Bu katman sayısı artışı ile performans artışının sağlanması zaten beklenen bir durumdur. Ek olarak çok kafalı dikkat yerine kullanılan dinamik evrişimlerin de buradaki performansı artırdığı düşünülmektedir. H_TS_WP mimarisi, değerlendirilen üç TA modeli içerisinde en iyi performansı göstermiştir. H_TS_WP'nin kullandığı kodlayıcının en iyi performansı göstermesindeki an etken iki seviyeli bir kodlayıcı mimarisine sahip olmasıdır. Bu iki katmanlı mimari sorguların içerisindeki derin anlamları ve bağlamları öğrenirken, aynı zamanda sorgulardaki kelimelerin anlam ve bağlamlarını öğrenebilmektedir. Fakat önerilen TQS modelinin, bazı ince ayarlar ve çeşitli yeni bileşenler kullanılarak hiyerarşik model ile rekabet edebilecek düzeye gelebileceği öngörülmektedir.

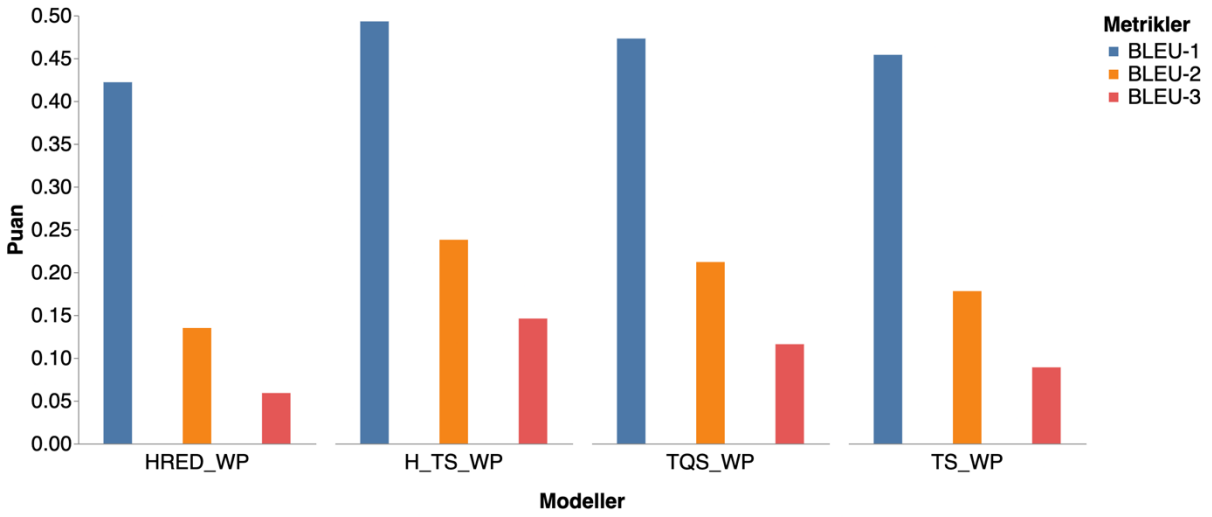
Kafa karışıklığı, yukarıda da bahsedildiği gibi bir dil modelin karşılaştığı ya da karşılaşmadığı sorgulara karşı oluşturduğu tepkiyi ölçer. Basit bir ifade ile modelin yeni karşılaştığı sorguları ne kadar iyi tahmin edebildiğini anlamaya yardımcı olur. TQS için kafa karışıklığını analiz ederken, yapılan son çalışmada belirtildiği gibi [2] temel çizgilerin kullandığı kelime dağarcığı ile aynı sayıda kelime dağarcığı kullanılmıştır. Çünkü kelime dağarcığı değiştiğinde modelin karşılaştığı bir kelimeye karşı kafasının karışma olasılığı da tüm modeller için farklılık gösterecektir. Kafa karışıklığı sonuçları Tablo 4.2'de verilmiştir. Ek olarak bu değerlere ait grafik gösterimi de Şekil 4.2'de verilmiştir. TA modelleri, RNN tabanlı temel çizgiyi geride bırakmıştır. Bu durum, öz dikkat mekanizmalarının bir kelime dizisindeki ögeleri RNN mimarilerine göre daha etkili yakaladıklarını doğrulamaktadır. TA modelleri içerisinde ise önerilen TQS modeli, TS_WP'ye göre kafa karışıklığını iyileştirmiştir. Buna, katman sayısının artırılmasının ve kullanılan farklı bileşenlerin neden olduğu düşünülmektedir. Katman sayısının artışı, TA modelinin bir dizideki daha derin anlamları ve ilişkileri yakalamasına fayda sağlamaktadır. Kafa karışıklığı için iyileştirme sağlamak adına kelime dağarcığı boyutunun artırılması iyi bir çözüm olabilir fakat bu çalışmada önerilen modelin temel çizgilerle doğru şekilde kıyaslanması için kelime dağarcığı boyutu sabit tutulmuştur.

Tablo 4.2: Değerlendirilen temel çizgilere ve TQS'ye ait kafa karışıklığı değerleri.

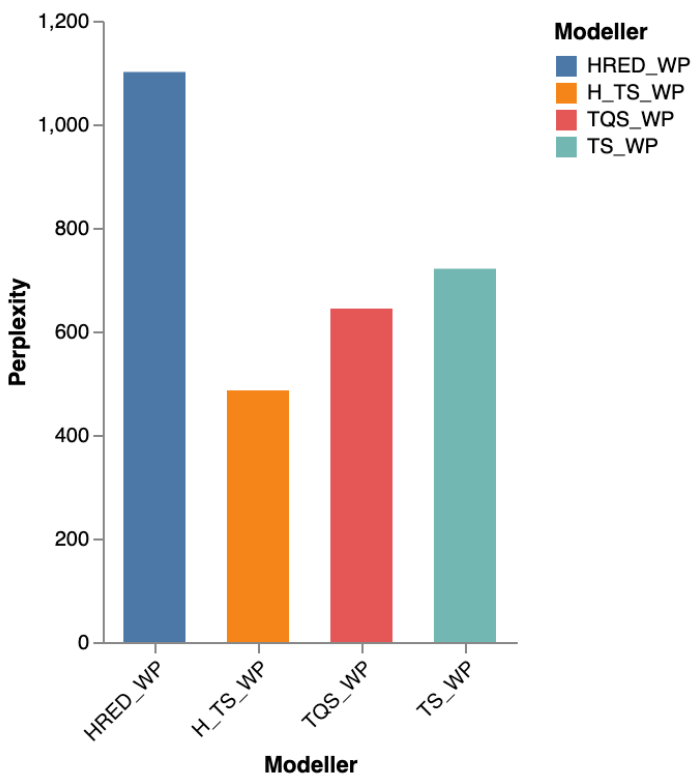
	Perplexity
HRED_WP	1101
TS_WP	721
H_TS_WP	486
TQS	644

TQS'nin kullandığı dinamik evrişim, çok kafalı dikkat mekanizmasına göre daha sadedir ve daha verimli çalışmaktadır. Bu durum TQS'nin orijinal çalışmada sunulan çok kafalı dikkat mekanizmasını kullanarak hazırlanan dönüştürücülere göre daha verimli çalışabileceğini göstermektedir. Bunu kanıtlamak için TQS modeli hem dinamik evrişimler hem de çok kafalı dikkat mekanizmaları kullanılarak eğitilmiştir. Bunun için iki farklı öz dikkat mekanizmasını ve farklı veri kümesi boyutlarını kullanan TQS modelleri eğitilmiştir. Bu eğitimler souncunda dinamik evrişimleri kullanan modelin eğitim süresi açısından daha hızlı olduğu gözlemlenmiştir. Dinamik evrişimleri kullanarak eğitilen modele ait hesaplanan süreler Tablo 4.3'te gösterilmiştir. Sonuçlar, dinamik evrişimlerin çoklu kafaya göre verimi artırdığını doğrulamaktadır. Çalışmada son olarak çeşitli kullanıcı oturumları için, temel çizgiler ve önerilen TQS modeli ile ışın aramaları kullanılarak oluşturulan QS örnekleri Tablo 4.4'te gösterilmiştir.

Ortaya çıkan sonuçlar, QS alanındaki önceden eğitilmiş TA'ları kullanmadan oluşturulan TQS modeli için gelecek adına umut verici sonuçlar sunmaktadır. Ayrıca eğitim için daha az zaman ve karmaşıklık sunması performans kaybına yol açmadan zamandan tasarruf sağlamaktadır. Gelecekte TQS modelinin performansını iyileştirerek daha alakalı ve bağlam duyarlı QS yapılabilmesi adına çalışmalar yürütülecektir. Bunun için farklı görevler için ortaya çıkan güncel TA bileşenleri incelenerek verimlilikleri ve QS görevine uygun olup olmadıkları değerlendirilecektir. Daha geniş ve daha kapsamlı araştırmaların ve TA bileşenlerinin sürekli gelişen dünyasının TQS modelini iyileştirmeye yardımcı olacağı düşünülmektedir. Ayrıca daha büyük hesaplama gücünün, modeli eğitmek ve performansını iyileştirmek için gelişim alanı sunacağı düşünülmektedir.



Şekil 4.1: Değerlendirilen temel çizgilere ve TQS'ye ait BLEU puanları.



Şekil 4.2: Değerlendirilen temel çizgilere ve TQS'ye ait kafa karışıklığı değerleri.

Tablo 4.3: TQS modelini çeşitli oturum büyüklüklerinde eğitmek için geçen süreler.

	Compute Engine	Model Params	Training Time
TQS-50K	**GPU:** Nvidia A100 **GPU RAM:** 40GB **CPU:** Intel(R) Xeon(R) CPU @ 2.20GHz	**Batch size:** 32 **Num of epochs:** 8	53m 4s
TQS-1.5M			3d 12h 48m 44s

Tablo 4.4: Temel çizgiler ve TQS'ye ait bazı QS örnekleri.

	Q1: divorce chat rooms **Q2: divorce support groups**	**Q1: maryland ocean city** **Q2: marylandocean vity hotel**	**Q1: carobean cruises** **Q2: spa resort** **Q3: disney world**
HRED_WP	divorce support rooms divorce chat groups divorce divorce groups divorce groups groups divorce support	maryland hotel hotel maryland hotel ocean maryland hotel hotel ocean maryland hotel maryland hotel ocean hotel	disney world resorts disney resort resorts disney world disney vacation resorts disney resorts
TS_WP	chat room listings ebay aol chat chat rooms divorce chat room	ocean city maryland ocean city md mapquest ocean country maryland expedia	disney world travelocity disney world hotels disney world cruise disney cruise
H_TS_WP	divorce support groups free divorce support groups divorce divorce chat rooms divorce support	maryland ocean city ocean city maryland hotels in maryland hotel ocean mapquest	disney world sea world disneyworld carnival cruise spa resort
TQS	divorce groups divorce chat room divorce support divorce support about divorce about	ocean city hotel ocean city hotel guides maryland ocean city tripadvisor ocean city maryland	cruise hotels travel with kids disney world disney cruise disney world cruise

5. TARTIŞMA

Yakın dönemde QS alanında pek çok çalışma yapılmıştır. Yapay zekanın hızlı ilerleyişi QS alanında kendisini göstermiştir. Derin öğrenme tabanlı yöntemlerin yükselişi ile QS alanında da derin öğrenmeyi kullanan çeşitli çalışmalar ortaya çıkmıştır. RNN ve CNN'i kullanan çalışmalar QS alanında önemli bir ilerlemeye katkıda bulunmuştur [5, 6, 41]. TA modellerinin ortaya çıkmasıyla birlikte doğal dil işleme alanında ciddi bir gelişim kaydedilmiştir. QS alanında da TA modelleri, son birkaç yıl içerisinde kullanılmaya başlamıştır [2, 3]. Yapılan çalışmalar, TA modellerinin QS alanındaki başarısını da kanıtlamaktadır. TA modelleri, barındırdıkları öz dikkat mekanizmaları ve paralel işlem yetenekleri ile daha kaliteli ve bağlam duyarlı QS sunabilmektedir.

TA, RNN, CNN ve LSTM gibi modellere göre daha iyi ve kaliteli sonuçlar üretiyor olmasına karşı hala ciddi bir gelişim alanı bulunmaktadır. Önceden eğitilmiş TA modellerini kullanan QS uygulamaları [2, 3], özel bir alana yönelik verileri kullanarak eğitilmedikleri için kullanıcıların arama görevine ait özel bağlamları yakalamaktan ve arama anlamlarına uygun öneriler sunmaktan uzaktır. Çok daha genel veri kümeleri kullanılarak eğitilen TA modelleri [42, 43, 64], kullanım kolaylığı ve eğitim maliyetleri için fırsatlar sunuyor olsa da hala sorgu görevine özel başarılı bir TA tabanlı QS modeli oluşturmak mümkündür. Yapılan çalışmada sunulan TQS modeli, bu alanda önceden eğitilmiş verileri kullanmak yerine bu göreve özgü olarak eğitilmiş dönüştürücülerin ilk adımı olarak görülmektedir. Ayrıca sunulan temel gerçekler QS alanında, sıfırdan bir TA modelini eğitmek için büyük miktarda etiketli veri sağlar. TQS modelinin, gelişime açık olduğu ve farklı dönüştürücü bileşenleri ve yöntemleri ile birleştirilerek daha kaliteli öneriler sunmak adına iyileştirilebileceği düşünülmektedir.

TQS modelini ortaya çıkaran bileşenler dikkatli ve titiz bir inceleme ve aramanın ardından seçilmiştir. Bu yapılırken hesaplama maliyetleri ve gücü de düşünülmüştür. Çünkü bir dil modeli geliştirmek oldukça maliyetlidir. Dönüştürücüleri sunan ilk çalışmadan [7] farklı olarak öz dikkat mekanizması için dinamik evrişimler [71] kullanılmıştır. Bu TQS'ye performans açısından önemli bir katkı sunmuştur. TQS'de orijinal çalışmada sunulan çok kafalı dikkat [7] de ayrıca denenmiştir. Fakat TQS modelinde, kalitede bir kayıp olmadan performansta artış sağladığı gerekçesiyle dinamik evrişimler tercih edilmiştir.

TQS modeli, çeşitli metrikleri ve temel çizgileri kullanarak değerlendirilmiştir. Derin sinir ağlarını kullanan, QS görevi için orijinal çalışmadaki dönüştürücüyü kullanan ve daha önce farklı bir görev için kullanılan hiyerarşik dönüştürücüyü [65] QS görevine uyarlayarak kullanan üç temel çizgi kullanılmıştır. TQS'in karşılaştırılan temel çizgilere göre etkili yönleri ve eksikleri değerlendirilmiştir. Sunulan TQS modeli, tekrarlayan sinir ağlarını ve orijinal çalışmada sunulan dönüştürücüyü QS görevi için özel olarak eğiten temel çizgileri geride bırakmıştır. Önerilen model, hiyerarşik dönüştürücüyü kullanan temel çizgiye göre daha başarısız olmuştur. Fakat bu durumun, TQS'nin geliştirilmesi ile aşılacağı düşünülmektedir. Ayrıca sunulan TQS modeli, eğitim için çok kafalı dikkat mekanizmasını kullanan dönüştürücülere göre hesaplama gücü ve maliyetleri açısından daha verimlidir.

6. SONUÇ VE ÖNERİLER

Çevrimiçi dünyanın gelişimi ile doğru orantılı olarak kullanıcı ve kaynak sayısında ciddi bir artış meydana gelmiştir. Kullanıcılar, çevrimiçi dünyada meydana gelen kaynak sayısının artışı ile aradıklarına ulaşabilmek için çok çeşitli filtrelerden geçirilmiş sonuçlara ihtiyaç duyarlar. Arama motorları tam olarak bu filtreleme işlemini gerçekleştirmeye yarayan araçlardır. Arama motorları kullanıcılara aradıkları nesneleri en uygun ve en hızlı şekilde göstermeye çalışır. Bu hedefi yerine getirebilmek için en önemli görevlerden biri QS dizileridir. QS kullanıcıların aramaların altında yatan anlamları kavrayarak, bu kavramlardan en doğru seçeneği kullanıcıya sunmaya çalışır. Böylelikle kullanıcılar için bilgiye erişim hızı artırılırken deneyim de iyileştirilir. Günümüzde QS ile ilgili pek çok farklı yöntem kullanılmıştır. Bunlar çizge tabanlı yöntemler, anlam tabanlı yöntemler ve makine öğrenmesi tabanlı yöntemlerdir. Fakat kullanılan çoğu yöntem, kullanıcıların isteklerini ve taleplerini anlamaktan uzaktır. Kullanıcı sorgularının altında yatan anlamları anlamak ve kaliteli QS dizileri sunmak günümüzde de güncelliğini korumaktadır. 2017 yılında ortaya çıkmaya başlayan TA modelleri yapay zekâ ve doğal dil işleme alanında bir devrim yaratmıştır. Bu mimariler, bir dizi öge arasındaki derin anlamları ve ilişkileri yakalamakta oldukça etkilidir.

Çalışmada sunulan TQS, TA modelinin gücünden yararlanırken bu mimarilerde kullanılan en yeni ve verimli bileşenleri de kullanmaya çalışmaktadır. Böylece kullanıcıların aradıkları bir dizi sorgu içerisindeki anlam ve bağlantılar yakalanarak, kullanıcılara en doğru önerilerin sunulması sağlanmaktadır. Aynı zamanda kalite ön planda tutulurken, hesaplama gücü ve maliyetleri de azaltılmıştır. Sunulan TQS modeli, değerlendirilen temel çizgilerinden bazılarını geride bırakmıştır. Bu TQS mimarisinin geleceği için umut vericidir. Sonuç olarak önerilen TQS modeli, önceden eğitilmiş TA kullanılmadan QS görevine özgü dönüştürücünün oluşturulabilmesi için bir temel niteliğindedir.

Önerilen TQS modeli, umut verici sonuçlar sunarken, gelecekteki geliştirmeler (parametre optimizasyonu, model mimarisindeki bileşenlerdeki değişiklikler, ek özelliklerin dahil edilmesi gibi) sistemi iyileştirmeye odaklanacaktır. Ayrıca kullanıcı oturumlarının daha iyi analiz edilerek veri kümesinin daha iyi hale getirilmesi ile TQS modelinin performansının iyileştirilebileceği düşünülmektedir.

KAYNAKLAR

[1] Reddy RR, Vaishnavi C, Mamatha C, et al. Location-aware keyword query suggestion techniques with artificial intelligence perspective. In: *Computational Analysis and Deep Learning for Medical Care: Principles, Methods, and Applications*. 2021. Epub ahead of print 2021. DOI: 10.1002/9781119785750.ch2.

[2] Mustar A, Lamprier S, Piwowarski B. On the Study of Transformers for Query Suggestion. *ACM Trans Inf Syst*; 40. Epub ahead of print 2022. DOI: 10.1145/3470562.

[3] Mustar A, Lamprier S, Piwowarski B. Using BERT and BART for query suggestion. In: *CEUR Workshop Proceedings*. 2020.

[4] Chen J, Mao J, Liu Y, et al. A Hybrid Framework for Session Context Modeling. *ACM Trans Inf Syst*; 39. Epub ahead of print 2021. DOI: 10.1145/3448127.

[5] Wu B, Xiong C, Sun M, et al. Query suggestion with feedback memory network. In: *The Web Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018*. 2018. Epub ahead of print 2018. DOI: 10.1145/3178876.3186068.

[6] Sordoni A, Bengio Y, Vahabi H, et al. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In: *International Conference on Information and Knowledge Management, Proceedings*. 2015. Epub ahead of print 2015. DOI: 10.1145/2806416.2806493.

[7] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: *Advances in Neural Information Processing Systems*. 2017.

[8] Miyanishi T, Sakai T. Time-aware structured query suggestion. In: *SIGIR 2013 - Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2013. Epub ahead of print 2013. DOI: 10.1145/2484028.2484143.

[9] Jiang D, Leung KWT, Yang L, et al. Query suggestion with diversification and personalization. *Knowl Based Syst*; 89. Epub ahead of print 2015. DOI: 10.1016/j.knosys.2015.09.003.

[10] Ma C, Zhang B. A New Query Recommendation Method Supporting Exploratory Search Based on Search Goal Shift Graphs. *IEEE Trans Knowl Data Eng*; 30. Epub ahead of print 2018. DOI: 10.1109/TKDE.2018.2815544.

[11] Huang Z, Mamoulis N. Location-aware query recommendation for search engines at scale. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2017. Epub ahead of print 2017. DOI: 10.1007/978-3-319-64367-0_11.

[12] Huang Z, Cautis B, Cheng R, et al. Entity-based query recommendation for long-tail queries. *ACM Trans Knowl Discov Data*; 12. Epub ahead of print 2018. DOI: 10.1145/3233186.

[13] Sadoughi S, Zarifzadeh S. An efficient hybrid query recommendation using shingling and hashing techniques. *Inf Syst*; 104. Epub ahead of print 2022. DOI: 10.1016/j.is.2021.101928.

[14] Jiang D, Leung KWT, Vosecky J, et al. Personalized query suggestion with diversity awareness. In: *Proceedings - International Conference on Data Engineering*. 2014. Epub ahead of print 2014. DOI: 10.1109/ICDE.2014.6816668.

[15] Anagnostopoulos I, Razis G, Mylonas P, et al. Semantic query suggestion using Twitter Entities. *Neurocomputing*; 163. Epub ahead of print 2015. DOI: 10.1016/j.neucom.2014.12.090.

[16] Momtazi S, Lindenberg F. Generating query suggestions by exploiting latent semantics in query logs. *J Inf Sci*; 42. Epub ahead of print 2016. DOI: 10.1177/0165551515594723.

[17] Barman D, Sarkar R, Tudu A, et al. Personalized query recommendation system : A genetic algorithm approach. *Journal of Interdisciplinary Mathematics*; 23. Epub ahead of print 2020. DOI: 10.1080/09720502.2020.1731964.

[18] Venugopal KR, Santosh Nimbhorkar S. Related search recommendation with user feedback session. In: *Web Recommendations Systems*. 2020. Epub ahead of print 2020. DOI: 10.1007/978-981-15-2513-1_6.

[19] Huang Z, Cautis B, Cheng R, et al. KB-enabled query recommendation for long-tail queries. In: *International Conference on Information and Knowledge Management, Proceedings*. 2016. Epub ahead of print 2016. DOI: 10.1145/2983323.2983650.

[20] Blanco R, Cambazoglu BB, Mika P, et al. Entity recommendations in web search. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2013. Epub ahead of print 2013. DOI: 10.1007/978-3-642-41338-4_3.

[21] Jiang JY, Wang W. RiN: Reformulation inference network for context-aware query suggestion. In: *International Conference on Information and Knowledge Management, Proceedings*. 2018. Epub ahead of print 2018. DOI: 10.1145/3269206.3271808.

[22] Jiang JY, Wang W. RiN: Reformulation inference network for context-aware query suggestion. In: *International Conference on Information and Knowledge Management, Proceedings*. 2018. Epub ahead of print 2018. DOI: 10.1145/3269206.3271808.

[23] Wu B, Xiong C, Sun M, et al. Query suggestion with feedback memory network. In: *The Web Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018*. 2018. Epub ahead of print 2018. DOI: 10.1145/3178876.3186068.

[24] Dehghani M, Rothe S, Alfonseca E, et al. Learning to attend, copy, and generate for session-based query suggestion. In: *International Conference on Information and Knowledge Management, Proceedings*. 2017. Epub ahead of print 2017. DOI: 10.1145/3132847.3133010.

[25] Sordoni A, Bengio Y, Vahabi H, et al. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In: *International Conference on Information and Knowledge Management, Proceedings*. 2015. Epub ahead of print 2015. DOI: 10.1145/2806416.2806493.

[26] Chen W, Cai F, Chen H, et al. Attention-based hierarchical neural query suggestion. In: *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*. 2018. Epub ahead of print 2018. DOI: 10.1145/3209978.3210079.

[27] Ahmad WU, Chang KW, Wang H. Context attentive document ranking and query suggestion. In: *SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019. Epub ahead of print 2019. DOI: 10.1145/3331184.3331246.

[28] Ahmad WU, Chang KW, Wang H. Multi-task learning for document ranking and query suggestion. In: *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. 2018.

[29] Han FX, Niu D, Chen H, et al. A deep generative approach to search extrapolation and recommendation. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019. Epub ahead of print 2019. DOI: 10.1145/3292500.3330786.

[30] Li R, Li L, Wu X, et al. Click feedback-aware query recommendation using adversarial examples. In: *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*. 2019. Epub ahead of print 2019. DOI: 10.1145/3308558.3313412.

[31] Puthiya Parambath SA, Anagnostopoulos C, Murray-Smith R. Improving Sequential Query Recommendation with Immediate User Feedback.

[32] Kohli H. Training Mixed-Objective Pointing Decoders for Block-Level Optimization in Search Recommendation. In: *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in*

Information Retrieval. 2020. Epub ahead of print 2020. DOI: 10.1145/3397271.3401236.

[33] Santos RLT, Macdonald C, Ounis I. Learning to rank query suggestions for adhoc and diversity search. *Inf Retr Boston*; 16. Epub ahead of print 2013. DOI: 10.1007/s10791-012-9211-2.

[34] Ozertem U, Chapelle O, Donmez P, et al. Learning to suggest: A machine learning framework for ranking query suggestions. In: *SIGIR'12 - Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2012. Epub ahead of print 2012. DOI: 10.1145/2348283.2348290.

[35] Ahmad WU, Chang KW, Wang H. Context attentive document ranking and query suggestion. In: *SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019. Epub ahead of print 2019. DOI: 10.1145/3331184.3331246.

[36] Doval Y, Gómez-Rodríguez C. Comparing neural- and N-gram-based language models for word segmentation. *J Assoc Inf Sci Technol*; 70. Epub ahead of print 2019. DOI: 10.1002/asi.24082.

[37] Kathuria A, Gupta A, Singla RK. A review of tools and techniques for preprocessing of textual data. In: *Advances in Intelligent Systems and Computing*. 2021. Epub ahead of print 2021. DOI: 10.1007/978-981-15-6876-3_31.

[38] Uysal AK, Gunal S. The impact of preprocessing on text classification. *Inf Process Manag*; 50. Epub ahead of print 2014. DOI: 10.1016/j.ipm.2013.08.006.

[39] Chai CP. Comparison of text preprocessing methods. *Nat Lang Eng*; 29. Epub ahead of print 2023. DOI: 10.1017/S1351324922000213.

[40] Zhang J, Li J. Intelligent language knowledge for cognitive engine. In: *Spatial Cognitive Engine Technology*. 2023. Epub ahead of print 2023. DOI: 10.1016/b978-0-323-95107-4.00012-3.

[41] Dehghani M, Rothe S, Alfonseca E, et al. Learning to attend, copy, and generate for session-based query suggestion. In: *International Conference on Information and Knowledge Management, Proceedings*. 2017. Epub ahead of print 2017. DOI: 10.1145/3132847.3133010.

[42] Devlin J, Chang MW, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In: *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. 2019.

[43] Radford A, Narasimhan K, Salimans T, et al. Improving Language Understanding by Generative Pre-Training. *OpenAI.com*.

[44] Lan Z, Chen M, Goodman S, et al. ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS. In: *8th International Conference on Learning Representations, ICLR 2020*. 2020.

[45] Nowakowski K, Ptaszynski M, Masui F. MiNgMatch-A fast n-gram model for word segmentation of the Ainu language. *Information (Switzerland)*; 10. Epub ahead of print 2019. DOI: 10.3390/info10100317.

[46] Alemi AA, Ginsparg PH. Text Segmentation based on Semantic Word Embeddings. *ArXiv*; abs/1503.05543, https://api.semanticscholar.org/CorpusID:16328631 (2015).

[47] Norvig P. Natural Language Corpus Data. 2009, pp. 219–242.

[48] Liang JJ, Tsou C-H, Devarakonda M V. Ground Truth Creation for Complex Clinical NLP Tasks - an Iterative Vetting Approach and Lessons Learned. *AMIA Jt Summits Transl Sci Proc*; 2017.

[49] Fairfield CJ, Cambridge WA, Cullen L, et al. ToKSA - Tokenized key sentence annotation - A novel method for rapid approximation of ground truth for natural language processing. *medRxiv*. Epub ahead of print 2021. DOI: 10.1101/2021.10.06.21264629.

[50] Kanclerz K, Gruza M, Karanowski K, et al. What if Ground Truth is Subjective? Personalized Deep Neural Hate Speech Detection. In: *1st Workshop on Perspectivist Approaches to Disagreement in NLP, NLPerspectives 2022 as part of Language Resources and Evaluation Conference, LREC 2022 Workshop*. 2022.

[51] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015; 521: 436–444.

[52] Kotu V, Deshpande B. *Chapter 10 Deep Learning*. 2019.

[53] Rosenblatt F. *The perceptron - A perceiving and recognizing automaton*. Ithaca, New York, January 1957.

[54] Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. *Commun ACM* 2017; 60: 84–90.

[55] Voulodimos A, Doulamis N, Doulamis A, et al. Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*; 2018. Epub ahead of print 2018. DOI: 10.1155/2018/7068349.

[56] Kamath U, Liu J, Whitaker J. *Deep learning for NLP and speech recognition*. 2019. Epub ahead of print 2019. DOI: 10.1007/978-3-030-14596-5.

[57] Grigorescu S, Trasnea B, Cocias T, et al. A survey of deep learning techniques for autonomous driving. *J Field Robot*; 37. Epub ahead of print 2020. DOI: 10.1002/rob.21918.

[58] Soydaner D. A Comparison of Optimization Algorithms for Deep Learning. *Intern J Pattern Recognit Artif Intell*; 34. Epub ahead of print 15 December 2020. DOI: 10.1142/S0218001420520138.

[59] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. In: *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*. 2013.

[60] Pennington J, Socher R, Manning CD. GloVe: Global vectors for word representation. In: *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. 2014. Epub ahead of print 2014. DOI: 10.3115/v1/d14-1162.

[61] Peters ME, Neumann M, Iyyer M, et al. Deep contextualized word representations. In: *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. 2018. Epub ahead of print 2018. DOI: 10.18653/v1/n18-1202.

[62] Shin HC, Lu L, Summers RM. Natural Language Processing for Large-Scale Medical Image Analysis Using Deep Learning. In: *Deep Learning for Medical Image Analysis*. 2017. Epub ahead of print 2017. DOI: 10.1016/B978-0-12-810408-8.00023-7.

[63] Chen W, Cai F, Chen H, et al. Attention-based hierarchical neural query suggestion. In: *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*. 2018. Epub ahead of print 2018. DOI: 10.1145/3209978.3210079.

[64] Lewis M, Liu Y, Goyal N, et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2020. Epub ahead of print 2020. DOI: 10.18653/v1/2020.acl-main.703.

[65] Garg V, Dhillon I, Yu H-F. Multiresolution Transformer Networks: Recurrence is Not Essential for Modeling Hierarchical Structure.

[66] Jagannatha AN, Yu H. Structured prediction models for RNN based sequence labeling in clinical text. In: *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*. 2016. Epub ahead of print 2016. DOI: 10.18653/v1/d16-1082.

[67] Yao L, Guan Y. An Improved LSTM Structure for Natural Language Processing. In: *Proceedings of 2018 IEEE International Conference of Safety Produce Informatization, IICSPI 2018*. 2019. Epub ahead of print 2019. DOI: 10.1109/IICSPI.2018.8690387.

[68] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*; 21.

[69] Dufter P, Schmitt M, Schütze H. Position Information in Transformers: An Overview. *Computational Linguistics*; 48. Epub ahead of print 2022. DOI: 10.1162/coli_a_00445.

[70] Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations. In: *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*. 2018. Epub ahead of print 2018. DOI: 10.18653/v1/n18-2074.

[71] Wu F, Fan A, Baevski A, et al. Pay less attention with lightweight and dynamic convolutions. In: *7th International Conference on Learning Representations, ICLR 2019*. 2019.

[72] Lei Ba J, Kiros JR, Hinton GE. Layer Normalization. *arXiv e-prints* 2016; arXiv:1607.06450.

[73] Hahnioser RHR, Sarpeshkar R, Mahowald MA, et al. Digital selection and analogue amplification coexist in a cortex- inspired silicon circuit. *Nature*; 405. Epub ahead of print 2000. DOI: 10.1038/35016072.

[74] Bahdanau D, Cho KH, Bengio Y. Neural machine translation by jointly learning to align and translate. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. 2015.

[75] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks. In: *Advances in Neural Information Processing Systems*. 2014.

[76] Papineni K, Roukos S, Ward T, et al. BLEU: A method for automatic evaluation of machine translation. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2002.

Fatih ÇELİK

ORİJİNALLİK RAPORU

%4	%4	%1	%2
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	%2
2	hal.sorbonne-universite.fr İnternet Kaynağı	%1
3	acikbilim.yok.gov.tr İnternet Kaynağı	<%1
4	Submitted to TechKnowledge Turkey Öğrenci Ödevi	<%1
5	mucet.github.io İnternet Kaynağı	<%1
6	www.selcukmedj.org İnternet Kaynağı	<%1
7	Agnès Mustar, Sylvain Lamprier, Benjamin Piwowarski. "On the Study of Transformers for Query Suggestion", ACM Transactions on Information Systems, 2022 Yayın	<%1
8	Submitted to Arab Open University Öğrenci Ödevi	<%1

ETİK KURUL İZİN YAZISI

Uyarı: Canlı denekler üzerinde yapılan tüm araştırmalar için Etik Kurul Belgesi alınması zorunludur.

☐ **Etik Kurul izni gerekmektedir.**

☒ **Etik Kurul izni gerekmemektedir.**

Fatih ÇELİK

KURUM İZNİ YAZILARI

Uyarı: Canlı ve cansız deneklerle yapılan tüm çalışmalar için kurum izin belgelerinin eklenmesi zorunludur. Gizlilik ve mahremiyet içeren durumlarda kurum adı kapatılmalıdır.

☐ **Kurum izni gerekmektedir.**

☒ **Kurum izni gerekmemektedir.**

Fatih ÇELİK