GÜRBÜZ BİLGİ ERİŞİMİ İÇİN SEÇKİLİ GÖVDELEME

Doktora Tezi

Gökhan GÖKSEL

Eskişehir 2022

GÜRBÜZ BİLGİ ERİŞİMİ İÇİN SEÇKİLİ GÖVDELEME

Gökhan GÖKSEL

Doktora Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Yazılımı Bilim Dalı

Danışman: Doç. Dr. Ahmet ARSLAN

(İkinci Danışman: Prof. Dr. Bekir Taner DİNÇER)

Eskişehir

Eskişehir Teknik Üniversitesi

Lisansüstü Eğitim Enstitüsü

Aralık 2022

Bu tez çalışması TÜBİTAK bilim ve teknoloji araştırma projeleri tarafından kabul edilen 114E558 no.lu proje kapsamında desteklenmiştir.

JÜRİ VE ENSTİTÜ ONAYI

Gökhan GÖKSEL'in GÜRBÜZ BİLGİ ERİŞİMİ İÇİN SEÇKİLİ GÖVDELEME başlıklı çalışması 05/12/2022 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Eskişehir Teknik Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği"nin ilgili maddeleri uyarınca, Bilgisayar Mühendisliği Anabilim dalında Doktora Tezi olarak kabul edilmiştir.

Jüri Üyeleri	**Unvan Adı Soyadı**	**İmza**
Üye	: Prof. Dr. Nihal ERGİNEL	
Üye	: Doç. Dr. Alper Kürşat UYSAL	
Üye	: Doç. Dr. Alper BİLGE	
Üye	: Dr. Öğr. Üyesi Cahit PERKGÖZ	
Üye	: Dr. Öğr. Üyesi Ali YÜREKLİ	

Prof. Dr. Murat TANIŞLI

Lisansüstü Eğitim Enstitüsü Müdürü

05/12/2022

DANIŞMAN ONAYI

Danışmanlığını yürüttüğüm Doktora öğrencisi Gökhan GÖKSEL, GÜRBÜZ BİLGİ ERİŞİMİ İÇİN SEÇKİLİ GÖVDELEME başlıklı tez çalışmasını tamamlamıştır. Hazırlamış olduğu tez tarafımca incelenmiş ve öğrencinin tez savunma sınavına alınması bilimsel ve etik açıdan uygun görülmüştür.

Tez Danışmanı

Doç. Dr. Ahmet ARSLAN

ÖZET

GÜRBÜZ BİLGİ ERİŞİMİ İÇİN SEÇKİLİ GÖVDELEME

Gökhan GÖKSEL

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Yazılımı Bilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Aralık 2022

Danışman: Doç. Dr. Ahmet ARSLAN

(İkinci Danışman: Prof. Dr. Bekir Taner DİNÇER)

Gövdelemenin bir bilgi erişim sisteminin performansını artırması beklenir, ancak pratikte, daha önceden elde edilmiş deneysel sonuçlar bunun her zaman geçerli olmadığını göstermektedir. Tezde, gövdelemenin sorgu bazında uygulanıp uygulanmayacağına karar veren seçkili bir gövdeleme yaklaşımı önerilmektedir. Metodumuz, anlamsal olarak ilişkili dokümanların geri getirilmesinde gövdelemeden kaynaklanan başarısızlık riskini en aza indirmeyi amaçlamaktadır. Tez, gövdelemenin bir bilgi erişim sisteminde nasıl seçkili bir şekilde uygulanabileceğini tanıtmasının yanı sıra; kelimelerin gövdeleme uygulanmadan önceki ve sonraki terim frekanslarından türetilen bir dizi yeni öznitelik önererek bilgi erişim literatürüne katkıda bulunur. Önerilen yöntem, hem bazı mevcut sorgu performansı tahmin edicilerinden hem de yeni türetilmiş özniteliklerden ve denetimli sınıflandırma yapan bir makine öğrenme tekniğinden yararlanır. Yöntem, üç kural tabanlı gövdeleme algoritması ile TREC ve NTCIR'da yer alan sekiz sorgu seti kullanılarak değerlendirilir. WSJ doküman koleksiyonu hariç kullanılan doküman koleksiyonları, sayısı 25 milyon ile 733 milyon arasında değişen Web dokümanlarından oluşmaktadır. Deneylerin sonuçları, yöntemin, sistemin gürbüzlüğünü artıran ve sorgular arasında başarısızlık riskini (sorgu başına performans kayıpları) en aza indirerek doğru seçimler yapabildiğini göstermektedir. Sonuçlar ayrıca, yöntemin çoğu sorgu seti için tekil sistemlerden sistematik olarak daha yüksek bir ortalama geri getirim performansına ulaştığını göstermektedir.

Anahtar Sözcükler: Seçkili gövdeleme, Seçkili bilgi erişimi, Gürbüzlük, Sınıflandırma.

ABSTRACT

SELECTIVE STEMMING FOR ROBUST INFORMATION RETRIEVAL

Gökhan GÖKSEL

Department of Department of Computer Engineering

Programme in Software Engineering

Eskişehir Technical University, Institute of Graduate Programs, December 2022

Supervisor: Assoc. Prof. Dr. Ahmet ARSLAN

(Co-Supervisor: Prof. Dr. Bekir Taner DİNÇER)

Stemming is supposed to improve the average performance of an information retrieval system, but in practice, past experimental results show that this is not always the case. In this thesis, a selective approach to stemming that decides whether stemming should be applied or not on a query basis is proposed. Our method aims at minimizing the risk of failure caused by stemming in retrieving semantically related documents. The thesis contributes to the information retrieval literature by proposing an application of selective stemming and introducing a set of new features derived from the term frequencies of the words and their stems produced by different stemming algorithms. The proposed method leverages both some of the existing query performance predictors and the newly derived features, and a supervised classification (machine learning) technique. It is evaluated using three rule-based stemmers and eight query sets of the standard TREC and NTCIR tracks. The document collections used, except WSJ document collection, consist of Web documents ranging in number from 25 million to 733 million. The results of the experiments show that the method is capable of making accurate selections that increase the robustness of the system and minimize the risk of failure (i.e., per-query performance losses) across queries. The results also show that the method attains systematically higher average retrieval performance than the single systems for most query sets.

Keywords: Selective stemming, Selective information retrieval, Robustness, Classification.

TEŞEKKÜR

Tezimin tamamlanmasında katkıları bulunan ve desteklerini esirgemeyen danışmanlarıma teşekkür ederim.

Gökhan GÖKSEL

05/12/2022

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Eskişehir Teknik Üniversitesi tarafından kullanılan "bilimsel intihal tespit programı"yla tarandığını ve hiçbir şekilde "intihal içermediğini" beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Gökhan GÖKSEL

İÇİNDEKİLER

Sayfa

TABLOLAR DİZİNİ

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

α	:	Alpha
θ	:	Theta
λ	:	Lambda
AP	:	Average Precision
BM	:	Best Match
CLEF	:	Cross Language Evaluation Forum
CW09B	:	ClueWeb09 Category B
CW12B	:	ClueWeb12 Category B
DF	:	Document Frequency
DFI	:	Divergence From Independence
DFR	:	Divergence From Randomness
ERR	:	Expected Reciprocal Rank
HTML	:	HyperText Markup Language
IDF	:	Inverse Document Frequency
IP	:	Internet Protocol
IR	:	Information Retrieval
k-NN	:	*k*-Nearest Neighbors
LTR	:	Learning-to-Rank
MAP	:	Mean Average Precision
nDCG	:	normalized Discounted Cumulative Gain
NRRC	:	Northeast Regional Research Center
NTCIR	:	NII Testbeds and Community for Information access Research
PMI	:	Pointwise Mutual Information
RIA	:	Reliable Information Access
SCQ	:	Collection Query Similarity
SCS	:	Simplified Clarity Score
PRP	:	Probability Ranking Principle
TREC	:	Text REtrieval Conference
URL	:	Uniform Resource Locator
WSJ	:	Wall Street Journal
XML	:	Extensible Markup Language

1. GİRİŞ

Bilginin yazıya dönüştürülmesi ve saklanmasının tarihi yaklaşık M.Ö. 3000'li yıllara kadar uzanmaktadır. Sümerler, çivi yazıları ile hayatın her alanını kapsayan bilgileri tabletlere aktarmıştır (Casson, 2001). İnsanlık yazı ile tanışmasıyla bilgi üretmiş, ürettiği bilgiyi sonraki nesillere aktarmak için çeşitli nesnelere aktarmış ve kütüphanelerde koruma altına almıştır. Kütüphane yönetim ve çalışanlarının, istenilen bilgiye nasıl ulaşılabileceğini ve arşivlerinde ne olduğunu bilmesi gerekmektedir. Bu problemlerin çözümü için yazılı materyaller üzerinde farklı kataloglama ve sınıflandırma yöntemleri geliştirilmiştir. Francis Bacon, The Advancement Of Learning eserinde, sınıflandırmayı Bellek, Hayal Gücü ve Akıl olmak üzere üç gruba bölmüş ve bunlarında alt gruplarını oluşturmuştur (Bacon, 1605). Thomas Jefferson ise Bacon'un kategorilerini Tarih, Felsefe ve Güzel sanatlar olarak çevirmiş ve kendi sınıflandırmasını oluşturmuştur (Jefferson, 2010).

1945 yılından sonra bilimsel çıktıların oldukça hızlı artması, bu çıktılara erişimde kullanılan mevcut yöntemlerin yetersizliğini ortaya çıkarmıştır. Böylece indeksleme yöntemleri de gelişme sürecine girmiştir. Calvin Mooers 1950 yılında "information retrieval" (bilgi erişimi) terimini literatüre kazandıran ilk kişi olarak daha çok bilinir (Mooers, 1950). 1990'lı yıllara geldiğimizde bilgisayar üretimi ucuzlamış birçok evlere ve ofislere girmiştir. World Wide Web, 1992 yılında kullanıma açılmıştır ve bu sanal ortamda oluşturulan bilgi her geçen yıl üstel bir hızla artmaya başlamıştır. Örneğin, Ocak 1996 yılında 100,000 olan web sitesi sayısı haziranda ikiye katlanmıştır. (Bilgi erişimi tarihi ile ilgili daha geniş bilgi için: Information Retrieval: The Early Years (Harman, 2019).)

Bilgi çağı içinde yaşadığımız bu dönemde insanlar teknoloji ve bilgi denizinde yüzmektedirler ve bu denizin büyüklüğü her geçen gün katlanmaktadır. Bu hızlı artış ile insanlar buna uyum sağlayabilmek için hızlı ve etkili bir şekilde bilgiye ulaşabilme ihtiyacı duymaktadır. Bu durum bilgi erişim yöntemlerinin, günümüzde yaşamımızın vazgeçilmez bir parçası haline gelmesini sağlamıştır. Bilgisayar veya mobil akıllı telefonlar ile yaptığımız hemen hemen her uğraşın bir yerlerinde farkında olalım ya da olmayalım bir şekilde bilgi erişim sistemlerinden yararlanırız. Bir akademisyenin bir konu hakkında literatürde makale araması, bir kişinin alışveriş sitesinde milyonlarca ürün arasından istediği ürünü bulabilmesi veya bulunulan konuma yakın restoran, eczane, kafe gibi konumların listelenmesi gibi işlemler bu tür sistemlerin temelini oluşturmaktadır.

1.1. Motivasyon

Gövdeleme, bilgi erişim sistemlerinin performansını önemli ölçüde ve sistematik olarak artırıp artırmadığı bilgi erişimi alanında tartışmalı bir konudur. Harman yaptığı çalışmada (1991) bilgi erişim sistemlerinin ortalama performansını etkilemediğini göstermiştir. Ancak, Krovetz (1993) ve Hull (1996) Harman'ın sonuçlarının aksi yönde, gövdelemenin ortalama performansı artırdığı sonucuna ulaşmıştır. Sonraki yıllarda yapılan çalışmalar ile gövdelemenin bilgi erişim sistemlerinin sonuç listesinde duyarlılığı (recall) artırarak az da olsa sistemin genel başarımı artırdığı gözlemlenmiştir ve bu durum gövdeleme algoritmaları için bilgi erişimi alanında genel kabul görmektedir. Gövdeleme duyarlılığı (recall) artırarak alakasız dokümanların da sonuç listesine girmesine neden olabilmektedir. Dolasıyla alakasız dokümanların sonuç listelerine dahil olabilmesi, gövdelemenin bazı sorgular için performansı artırdığı bazı sorgular için ise performansı olumsuz yönde etkilediğinin göstergesidir.

Bir bilgi erişim sisteminin verilen her bir sorgu için kabul edilebilir derecede tatmin edici sonuçlar üretmesi beklenir. Bilgi erişim öncesi ve sonrası uygulanan teknikler göz önünde bulundurulduğunda bilgi erişim sistemlerin bazı sorgular için başarısız olması, böyle bir sistemin inşa edilmesini oldukça güç kılar. Günümüz gövdeleme algoritmaları her ne kadar bilgi erişim sistemlerinin başarımını ortalamada artırsa da, bazı sorgularda sistemin gövdeleme uygulanmamış yalın hali daha yüksek performans sergileyebilmektedir. Gövdeleme aslında bu tür münferit sorgular için sistemin başarımını düşürerek uygulandığı sistemin optimal potansiyel performansına ulaşmasına engel olmaktadır. Bir başka deyişle bazı sorgular için risk teşkil etmektedir. Seçkili yaklaşımlar yeni bir yöntem sunmak yerine her sorgu için var olan yöntemler arasından seçim yaparak bu tür hataları en aza indirmeyi hedeflemektedir. Böylece gövdeleme uygulanmamış sistemin daha yüksek başarım ile cevaplayabileceği sorgular seçkili yöntem ile önceden tespit edilebilmektedir.

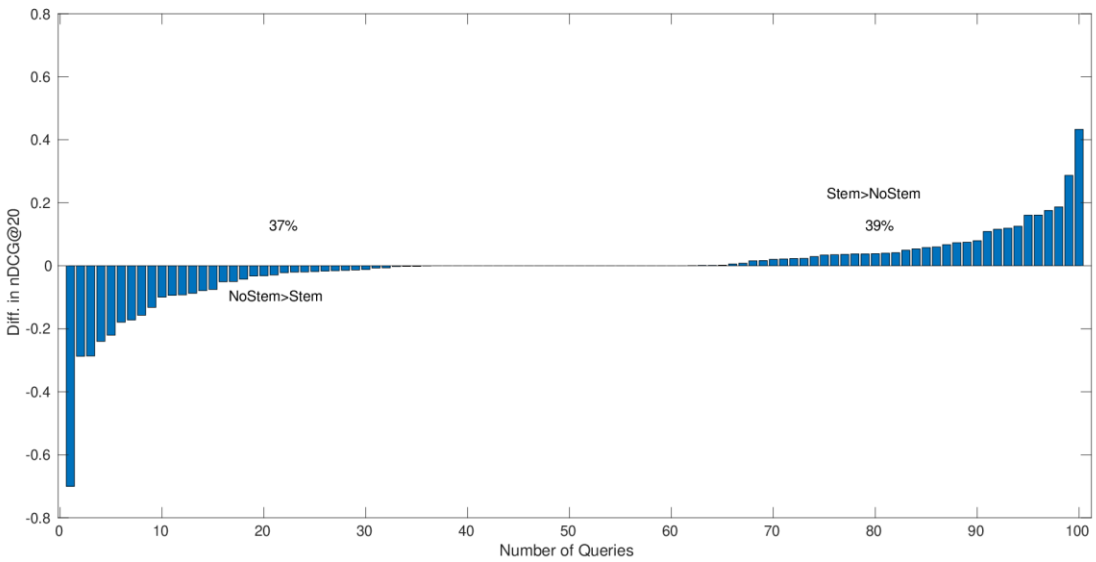
Güvenilir bilgi erişimi çalıştayında (Reliable Information Access (RIA)) (Harman & Buckley, 2004) bilgi erişim etkinliği için sorguların değişkenlik faktörleri kapsamlı bir şekilde ele alınmıştır. Çalıştayda, katılımcılar bilgi erişim sistemlerinin bazı münferit sorgularda sistemin ortalama başarımına kıyasla neden başarısız olduğunu araştırmıştır. Buckley (2004; 2009) hataları on temel kategoride toplamıştır ve bunlardan birinde genel teknik hata olarak gövdelemeye yer vermiştir. RIA'da üretilen sonuçlardan biri de (Harman & Buckley, 2009) yeni bir teknik tanıtmak yerine hangi sorgularda hangi IR

tekniklerinin kullanılabileceğini bulmaktır. Bu bağlamda, kullanıcıların her bir bilgi ihtiyacının karşılanmasını hedefleyen ve mevcut sistemler arasından en iyi erişim sistemini seçen seçkili bilgi erişim çalışmaları önem kazanmıştır. Seçkili yaklaşımlar, optimum performans elde etmek için birden fazla sistemin güçlü yönlerinden yararlanmaktadır. Bu sayede, seçkili yaklaşım optimum durumda seçime katılan tekil sistemlerden daha iyi performans üretecektir.

Gövdeleme, bilgi erişim sistemlerinde uygulanan bir tekniktir. Harman (1991) yaptığı çalışmada sorguların gövdelemeden pozitif veya negatif yönde etkilendiğini şu sözleri ile aktarmıştır:

> "*Although individual queries were affected by stemming, the number of queries with improved performance tended to equal the number with poorer performance, thereby resulting in little overall change for the entire test collection.*" (Harman, 1991)

Harman'ın sözleri "Sorgular gövdelemeden etkilenmesine rağmen performansı artıran sorguların sayısı performansı düşüren sorguların sayısına eşit olmaya eğilimlidir, dolayısıyla tüm test koleksiyonu için ortalamada az bir değişiklikle sonuçlanır." şeklinde çevrilebilir.



Şekil 1.1. *NTCIR-13 WWW-1'deki 100 sorgu için risk grafiği*

Şekil 1.1. NTCIR-13 WWW-1'deki 100 sorgu için risk grafiğini göstermektedir. BM25 terim ağırlıklandırma modeli ile oluşturulan grafikte yatay eksende sorgu sayıları

(Number of Queries), dikey eksende ise her sorgu için gövdeleme (Stem) ve gövdelemesiz (NoStem) bilgi erişim sistemlerinden elde edilen nDCG@20 (normalized Discounted Cumulative Gain) performans değerlerinin farkı verilmektedir. KStem gövdeleme algoritması (Krovetz, 1993) uygulanan sistemde (Stem) 39 sorgu, gövdelemesiz sisteme (NoStem) göre daha başarılı iken, 37 sorgu için gövdelemesiz sistem daha başarılı olmuştur. Gövdeleme uygulanmış sistemin ortalama başarımı 0.3709; gövdelemesiz sistemin ise 0.3755'dir. Her bir sorgu için en iyi sistemler seçildiği durumlarda (oracle model) ise 100 sorgu üzerinden ortalama başarım 0.4041'dir. Ayrıca oracle model, her iki tekil sisteme göre istatistiksel olarak anlamlı derecede ($p - value_{Stem} = 0.0004, p - value_{NoStem} = 0.00002$) daha yüksek başarım elde etmektedir. Grafik göz önünde bulundurulduğunda NTCIR-13 WWW-1 (Luo, ve diğerleri, 2017) kapsamında sunulan 100 sorgu için Harman'ın gözleminin hala geçerli olduğu gözlenmektedir.

1.2. Bilimsel Araştırma Soruları ve Hipotezler

Bu tezdeki amaç, her bir sorgu için verilen bir gövdeleme algoritması uygulanmış sistem ile o sistemin gövdeleme uygulanmamış hali arasından sorgu bazlı en etkili sistemi yüksek doğruluk oranı ile önceden tahmin edebilmektir. Bu bağlamda çalışmamız aşağıdaki araştırma sorularını içermektedir:

- R1 Seçkili yaklaşım bir sistemin performansını istatistiksel olarak anlamlı düzeyde artırma potansiyeline sahip midir?
- R2 Seçkili yaklaşım, gövdeleme algoritması kullanan bilgi erişim sistemlerinin gürbüzlüğüne katkıda bulunuyor mu?
- R3 Seçkili yaklaşım, gövdelemenin uygulanması gereken sorguları tahmin etmede ne derecede başarılıdır?

Yukarıda belirtilen araştırma sorularından *R1*'in amacı seçkili gövdeleme yaklaşımının üretebileceği en yüksek performansı belirlemeye yöneliktir. Hatasız tahminde bulunan seçkili gövdeleme yönteminin oluşturulduğunun varsayıldığı bu durumda (Oracle model); üretilebilecek en yüksek performans puanı seçkili gövdelemenin potansiyelini belirler. *R2* ise, seçkili bir gövdeleme yaklaşımının gürbüzlüğünü sınamaya yöneliktir. Son olarak *R3*, sınıflandırma problemi olarak ele alınan seçkili yöntemin ne kadar doğrulukla başarılı olduğunu ölçmeyi amaçlamaktadır. Araştırma sorularımıza cevap ararken oluşturulan hipotezler:

H1 Seçkili gövdeleme yaklaşımının üretebileceği en yüksek performans mevcut tekil sistemin ürettiği performansa göre istatistiksel olarak anlamlı düzeyde farklıdır.

H2 Verilen bir sorgu için; gövdeleme uygulanmış bir bilgi erişim sisteminin, anlamsal olarak ilgili dokümanları geri getirmedeki başarısızlık riskini en aza indiren ve gürbüzlüğü artıran bir seçkili gövdeleme modeli oluşturulabilir.

H3 Sınıflandırma problemi olarak ele alınan seçkili yöntemin doğruluk oranı, sorgulara gövdeleme uygulanıp uygulanmayacağını tahmin etmede başarılıdır.

1.3. Bilimsel Katkılar

Tezin ana katkılarından biri literatürde sunulan erişim öncesi terim öznitelikleri ile gövdeleme uygulandığı ve uygulanmadığı durumlarda terimlerin frekans dağılımlarındaki farklılık göz önünde bulundurularak üretilen öznitelikler kullanılarak, gözetimli sınıflandırma temelli bir seçkili gövdeleme modeli oluşturulmasıdır. Gövdeleme uygulandığı durumda terimlerin doküman ve koleksiyon terim frekansları değişmektedir. Terimlerin frekanslarındaki bu değişimlerden yararlanılarak terim boyutundan sorgu boyutuna (sorgular birden fazla terimler içerebilmektedir) taşınarak çeşitli öznitelikler oluşturulmuştur.

Oracle model, birden fazla sistem içinden her bir sorgu için en iyi sistem seçilerek ulaşılabilecek en yüksek başarımdır. Seçkili gövdeleme ile bir gövdeleme algoritmasının kullanılması ile elde edilebilecek performansı oldukça yüksek seviyelere taşıyabilme potansiyeline sahip olduğu görülmüştür. Bu bağlamda seçkili gövdeleme yaklaşımının ulaşabileceği en yüksek başarımlar ortaya konmuştur.

Son olarak, bilgi erişim sistemlerinde gövdeleme algoritmasını saf bir şekilde uygulamak yerine, seçkili gövdeleme yaklaşımın benimsenmesi sistemin performansı artırabilmektedir.

Özetle tezde sunulan katkılar şu şekilde sıralanabilir:

- Sorgu bazlı bir seçkili gövdeleme yaklaşımı, bir bilgi erişim sistemine yalın bir şekilde kural tabanlı bir gövdeleme algoritmasının uygulanması veya uygulanmamasıyla elde edilebilecek başarımdan daha yüksek bir başarım elde etme potansiyeline sahiptir.

- Uygun öznitelikler ve makine öğrenmesi teknikleri ile oluşturulan seçkili gövdeleme modeli bilgi erişim sisteminin gürbüzlüğünü artırmaktadır.

1.4. Tez Organizasyonu

Tezin organizasyonu için akış şu şekildedir:

Bölüm 2- Bilgi Erişim Sistemleri: Bu bölümde bilgi erişim sistemleri ile ilgili temel bilgiler verilmiştir. Sistemin temel bölümlerine değinilmiş ve sistemin performans ölçümü ile ilgili detaylar sunulmuştur.

Bölüm 3- Gövdeleme Algoritmaları: Bu bölümde gövdeleme algoritmaları ile bu algoritmaların sınıflandırılması ile ilgili bilgilere yer verilmiştir.

Bölüm 4- İlgili Çalışmalar: Bu bölümde tez konusu ile ilgili literatürdeki çalışmalara değinilmiştir.

Bölüm 5- Deneysel Yöntem: Bu bölümde yapılan deneylerde kullanılan doküman koleksiyonları, sorgu setleri, terim ağırlıklandırma modelleri ve bilgi erişiminde kullanılan yazılım çatısı hakkında bilgilere sunulmuştur.

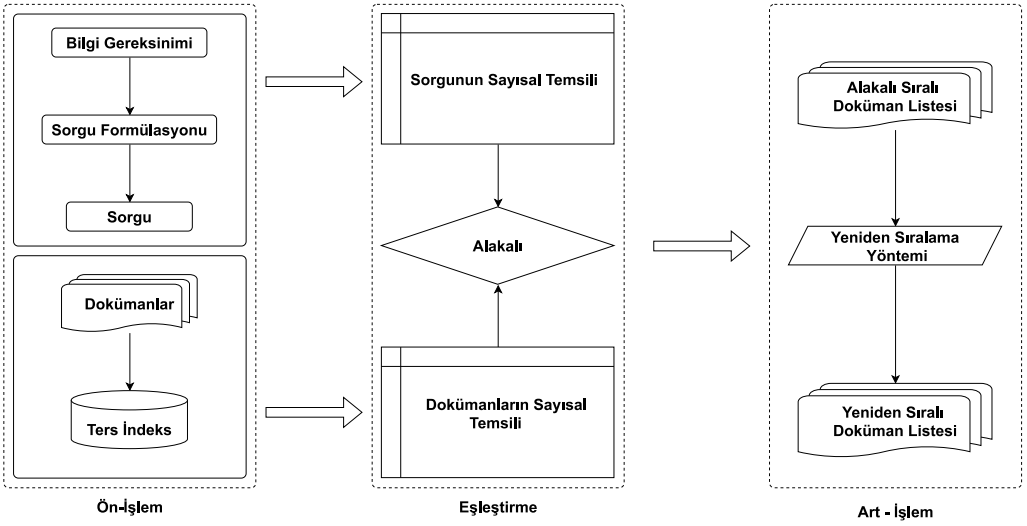
Bölüm 6- Bilgi Erişiminde Seçkili Gövdeleme: Bu bölümde sunulan seçkili gövdelemenin detayları ve performans sonuçları verilmiştir.

Bölüm 7- Sonuçlar ve Öneriler: Bu bölümde tez kapsamında yapılan çalışmaların genel değerlendirmesi sunulmuştur.

2. BİLGİ ERİŞİM SİSTEMLERİ

Bilgi erişimi, istenilen bir bilgi gereksinimi ile alakalı dokümanların bir doküman koleksiyonu arasından bulunması olarak tanımlanabilir (Dominich, 2008; Croft, Metzler, & Strohman, 2010). Bir bilgi erişim sistemi, dokümanları organize etme, depolanma ve kullanıcı tarafından verilen bir bilgi gereksinimi ile dokümanlar arasında alaka (ilgi) ölçümü yapma işlevlerini içinde barındırır. Bilgi erişim sistemi tarafından geri getirilen bir dokümanın verilen bilgi gereksinimi ile alakalı (ilgili) olabilmesi için; dokümanın, kullanıcının ihtiyacı olan bilgi gereksinimini tatmin edici derecede sağlaması gerekir. Bilgi gereksinimi, kullanıcının ihtiyacı olan bilgiyi tarif ederken, sorgu ise bilgi gereksinimini bilgi erişim sistemine verirken kullandığı ve birkaç terimden oluşan kısa halidir. Örneğin; bilgi gereksinimi "*You want to know the definition of Smart Home.*" (Akıllı evin tanımını bilmek istiyorsun) olduğu durumda, sorgu "*Smart home*" (Akıllı ev) olabilmektedir. Burada dokuz terimden oluşan bir bilgi gereksinimi iki terime indirgenmiştir. "Kelime" yerine "Terim" ifadesini kullanmamızın sebebi, terimin kelimenin yanı sıra tarih, sayı, URL, bir ürünün seri kodu gibi çeşitli karakter öbeklerini de kapsamasıdır.

Bir bilgi erişim sistemi temelde dört alt bileşenden oluşmaktadır. Bunlardan birincisi olan terim sözlüğünün belirlenmesi ve indeksleme; bilgi erişim sistemine konu olan milyonlarca dokümanların terimlerinin belirlenmesi ve üzerinde hızlı bir şekilde arama yapmaya olanak verecek bir veri yapısının oluşturulmasıdır. Bir sonraki bileşen olan sorgu formülasyonu ise kullanıcı sorgusunun bilgi erişim sisteminin işleyeceği formata çevrilmesidir. Sistemin en temel bileşeni sorgu-doküman eşleşmesi ve puan atamadır. Sistemin geri getirdiği dokümanların farklı teknikler ile yeniden sıralanması bir bilgi erişim sisteminin son bileşenidir. Şekil 2.1. ile bilgi erişim süreci ve alt bileşenleri gösterilmektedir. İlk iki bileşen, sorgunun ve indekslerin oluşturulduğu bilgi erişim sisteminin ön-işlemi; son bileşen olan yeniden sıralama ise sistemin art-işlem aşamalarıdır.

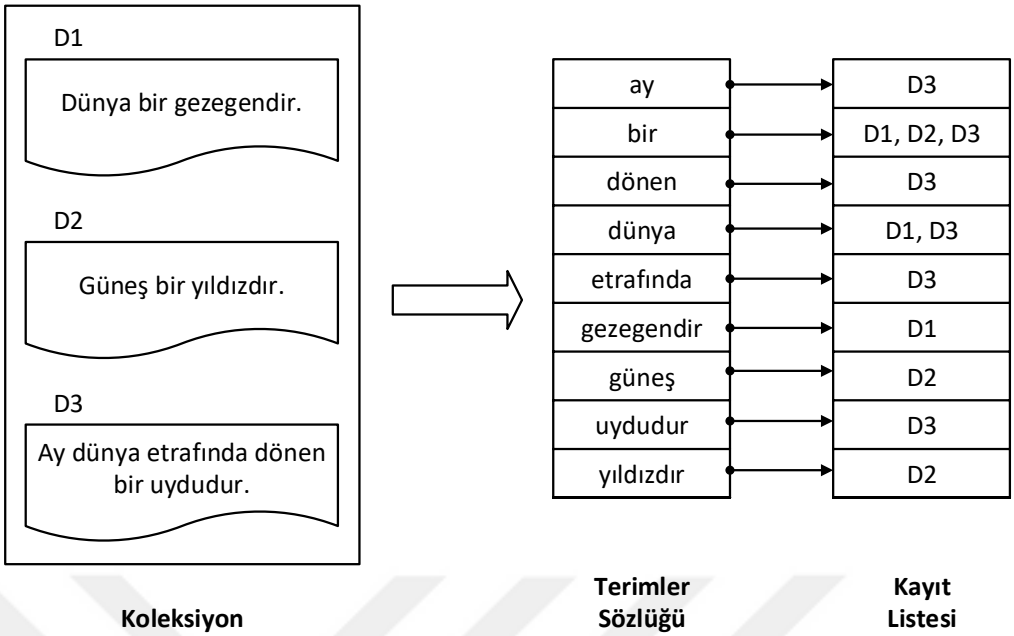


Şekil 2.1. *Bilgi erişim süreci ve alt bileşenleri*

2.1. Terimler Sözlüğünün Belirlenmesi ve İndeksleme

Bir bilgi erişim sisteminin amacı verilen bir sorgu ile alakalı dokümanları kullanıcıya geri getirmektir. Sistemin belirli bir rutin ve yüksek sorgu sayısıyla çalıştığı göz önünde bulundurulursa, her bir sorgu terimi için dokümanlar üzerinde lineer gezinerek bir terimi aramanın hesaplama maliyeti oldukça yüksektir. Bu problemin çözümü için bilgi erişim sistemlerinde ters indeksleme (inverted index) kullanılmaktadır. İndeksin yapısı genel olarak, koleksiyondaki benzersiz terimlerden ve her bir terimin işaret ettiği kayıt listesinden oluşmaktadır. Kayıt listesi ise ilgili terimin bulunduğu dokümanları içermektedir. Ayrıca, dokümanların yanı sıra terimin o doküman içindeki konumunu da içerebilir. Böylece bir sorgu teriminin hangi dokümanlarda ve dokümanın neresinde olduğu bilgisine oldukça hızlı bir şekilde ulaşılır. Şekil 2.2. ile bir ters indeksin genel yapısı gösterilmiştir.

Bir indekste benzersiz terimlerin belirlenmesi ile terimler sözlüğü oluşturulur. Terimler sözlüğü ile indekslemede kullanılacak terimler belirlenmiş olunur. Terimler sözlüğü belirlenirken dokümanlara çeşitli temel işlemler uygulanır. Bunlar, doküman içindeki metnin parçalara bölünmesi (tokenizasyon), normalizasyon, durak kelimelerin atılması ve gövdelemedir. Bu işlemlerin ayrıntılarına alt başlıklarda değinilmiştir.



Şekil 2.2. *Ters indeks şeması*

2.1.1. Tokenizasyon

Verilen bir karakter dizisini belirli bir kural çerçevesinde anlamlı küçük parçalara bölme işlemi tokenizasyon olarak tanımlanır. Parçalara ayırma işlemi uygulama alanı veya dokümanların doğal diline göre farklılık gösterebilir. Örneğin, Latin alfabesi ile yazılmış çoğu diller için beyaz boşluk karakterine göre parçalama işlemi bilgi erişim sistemlerinde genel kullanıma sahiptir. Beyaz boşluk karakterleri; yatay boşluk karakteri olan "space", dikey boşluk veya satır sonu karakteri "enter" ve birden fazla boşluk bırakan "tab" karakterleridir. Ancak, Almanca'da isim tamlamaları bitişik yazılır. Örneğin, Almanca'da "teppichwaschmaschine" kelimesi Türkçe'de "halı yıkama makinası" anlamına gelmektedir ve görüldüğü gibi üç farklı ifadeyi içinde barındırmaktadır. Bunun gibi farklı dil yapıları için tokenizasyon işlemlerinde farklı modüller kullanılır.

Bazı sistemlerde ise e-posta, URL, IP adres gibi karakter dizilerinin belirlenmesi ve bütün olarak tutulması gerekebilir. Bunun gibi durumlarda spesifik tokenizasyon işlemlerine ihtiyaç vardır. Tokenizasyon işlemi bilgi erişiminin en önemli ve kritik aşamalarından biridir. Bu işlem sonunda terimler sözlüğünde yer alabilecek terimlerin çatısı oluşturulur.

2.1.2. Normalizasyon

Dokümanlarda bir terimin farklı türde formları olabilir. Farklı formdaki terimler aynı anlamı içerseler de karakter dizileri farklı olduğu için (ör. “Portakal” ve “portakal”) bunlar farklı terimler olarak algılanır. Bilgi erişim sistemlerin bu problemin çözümünde en sık kullanılan yöntemlerden biri, büyük harfleri küçük harflere çevirmektir. Tüm koleksiyon indekslenirken küçük harflere dönüştürülür ve terimler sözlüğünde bu formuyla tutulmasında fayda vardır. Ayrıca kullanıcı tarafından gönderilen bir sorgunun terimleri, terimler sözlüğünde bulunabilmesi için sorgu da aynı işleme tabi olur.

2.1.3. Durak kelimelerin atılması

Edat, bağlaç ve zamir gibi hemen hemen tüm dokümanlarda görülme olasılığı yüksek olan terimlerin, dokümanların sahip olduğu bilgiye çok fazla bir katkısı yoktur. İngilizce’de “a”, “an”, “the”, “of”, “for” gibi terimler bunlara örnek verilebilir. Dokümanın sahip olduğu enformasyona katkısı az olan ve oldukça sık görülen bu terimlere durak kelime ve bu terimlerden oluşan listeye durak kelime listesi olarak tanımlanır. Durak kelimeler indeksleme sırasında göz ardı edilebilir ve terimler sözlüğüne eklenmeyebilir. Böylece kayıt listesinin boyutunda önemli bir düşüş gözlenir. Ancak bu işlemin uygulanması tamlamalardan oluşan sorguların performansında zararlı olabilmektedir. “President of the United State” sorgusu iki tane durak kelime içermektedir ve durak kelimeler atıldığında anlam düşüklüğüne sebep olmaktadır. Daha çarpıcı diğer bir örnek ise “To be or not to be” sorgusudur ve tamamen durak kelimelerden oluşmaktadır. Dolayısıyla durak kelimelerin atılması dikkat edilmesi gereken bir konudur.

2.1.4. Gövdeleme

Dokümanların dil yapısına bağlı olarak terimler farklı morfolojik yapılarda olabilir. “kelime, kelimeler, kelimesi, kelimenin” gibi aynı kökten türemiş kelimeler benzer anlamlara sahip olabilir. Çoğu durumda bu terimler ile yapılan aramalarda sadece sorguda belirtilen terimi içeren dokümanları değil onun morfolojik varyasyonlarını da içeren dokümanları getirmek faydalı olabilmektedir. Gövdelemenin amacı bu tür aynı köke sahip farklı varyasyonlardaki terimleri tek bir köke indirgemektir. Gövdelemede sıklıkla bu işlem, terimlerin son ekleri atılarak yapılır. Literatürde farklı dil yapıları için çeşitli gövdeleme algoritmaları mevcuttur. Bilgi erişim sisteminde gövdeleme kullanılırken

dokümanların sahip olduğu dile uygun algoritmalar seçilir. Ayrıca gövdeleme bir sisteme uygulanmış ise aynı yöntem verilen sorguya da uygulanmalıdır.

2.2. Sorguların Formülasyonu

Sorgu formülasyonu bilgi erişim sisteminin bir parçası olup, verilen bir kullanıcı sorgusunu manipüle eden bir işlemdir. Sorgu formülasyonunun amacı, kullanıcı sorgusunun niyetine uygun olarak değiştirerek, kullanıcıya daha alakalı dokümanlar sunmaktır. Sorgu formülasyonu, kullanıcılara yazdıkları sorgular ile ilgili sorgu tavsiyeleri vermek şeklinde olabilir. Ticari arama motorlarında karşılaştığımız "Bunu mu demek istediniz?" gibi geri bildirimler buna örnektir. Ayrıca sorguların yeniden yazılması veya transformasyonu da sorgu formülasyonu işlemlerindendir.

2.3. Sorgu-Doküman Eşleşmesi ve Puan Atama

Bu aşama bilgi erişim sistemlerinin en temel bileşenidir. Verilen bir kullanıcı sorgusuna göre koleksiyondaki dokümanlardan alakalı olanları belirleme özelliğine sahiptir. Ayrıca dokümanlar ile kullanıcı sorgusu arasındaki alaka derecesini belirleyen bir yöntemdir. Kullanıcılar geri getirilen tüm dokümanları inceleyemezler. Dolayısıyla kullanıcılara geri getirilen dokümanlar alaka derecesine göre sıralı olmalıdır ve bu yöntem ile kullanıcıların alakalı dokümanları seçmek için harcayacakları zaman en aza iner. Literatürde, sorgu-doküman eşleşmesi için Boole modellerden vektör-uzay modellerine ve olasılıksal yöntemlere dayalı çeşitli modeller sunulmuştur.

2.3.1. Boole modeli

Boole geri getirim modeli küme teorisine dayalı en basit modeldir. Model, verilen sorguyu Boole ifadelerine (Boole, 1854) çevirerek koleksiyon içinde arama yapar. Dijital kütüphane gibi spesifik alanlarda bu model kullanışlı olsa da dokümanlara puan atama işlemi yapmadığı için sıralamanın önemli olduğu Web tabanlı bilgi erişim sistemlerinde kullanışlı olduğu söylenemez.

Boole sorgular oluşturulurken AND, OR ve NOT operatörleri kullanılır. *A* ve *B* terimlerinden oluşturulan Boole sorguları aşağıdaki gibi ifade edilir:

(*A* AND *B*): $A \cap B$, her iki terimin geçtiği dokümanları ifade eder.

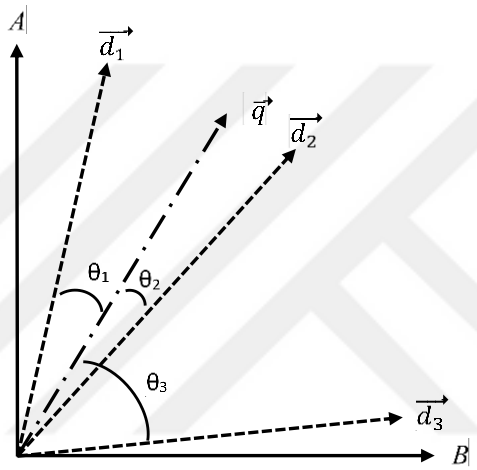
(*A* OR *B*): $A \cup B$, her iki terimden en az birinin geçtiği dokümanları ifade eder.

NOT *A*: $\bar{A}$, *A* terimini içermeyen dokümanları ifade eder.

Boole modeli sahip olduğu bu özellik sebebiyle bilgi erişim modelinden çok veri alma modeli olarak kabul edilir.

2.3.2. Vektör uzay modeli

Vektör uzay modelinde (Salton G. , 1968; Salton, Wong, & Yang, 1975; Salton & McGill, 1986) dokümanlar ve sorgular vektörel olarak temsil edilir. Yüksek boyutlu bu uzayın her bir boyutu terimler sözlüğündeki bir terimi ifade eder.



Şekil 2.3. *Vektör uzay modeli gösterimi*

Verilen bir sorgu vektörü ile doküman vektörleri arasındaki benzerlik ölçümü ile dokümanlara puan atanır. Benzerlik ölçümü sorgu vektörü ile her bir doküman vektörü arasındaki açı ile belirlenir. Sorgu vektörü ile en küçük açıya sahip olan doküman vektörü birbirine en benzer vektörlerdir. Şekil 2.3. ile vektör uzay modeli gösterimi verilmiştir. Şekilde $\theta_2 < \theta_1 < \theta_3$ olduğu için $\bar{q}$ sorgu vektörüne en yakın doküman vektörü $\bar{d}_2$ vektörüdür.

Lineer cebir hesaplamaları ile iki vektör arasındaki açı hesaplanabilir. Verilen bir doküman vektörü $\vec{d}$ ve sorgu vektörü $\vec{q}$ arasındaki açı iç çarpım ile hesaplanır. İç çarpım; dot product, inner product veya scalar product isimleri ile de adlandırılır. Denklem (2.1) bu iki vektör arasındaki iç çarpımı vermektedir.

$$\vec{q} \cdot \vec{d} = |\vec{q}| \cdot |\vec{d}| \cdot \cos(\theta) \tag{2.1}$$

$$score(\vec{q},\vec{d}) = \cos(\theta) = \frac{\vec{q}\cdot\vec{d}}{\|\vec{q}\| \times \|\vec{d}\|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2} \times \sqrt{\sum_{i=1}^{|V|} d_i^2}} \quad (2.2)$$

Denklem (2.2) vektör uzay modelde bir doküman vektörü ile sorgu vektörü arasındaki benzerliği ölçmektedir. $|V|$ terimler sözlüğünün boyutunu ve modelin sahip olduğu uzayın boyutunu ifade eder. Ayrıca denklemde $\|\cdot\|$ ifadesi Öklid veya *L2*-norm anlamına gelmektedir.

Doküman boyutu büyüdükçe, sorgu terimleri ile kesişimi doğal olarak artması beklenmektedir. İkili ağırlıklandırma, yani vektörün 0 ile 1'lerden oluştuğu durumda, ayırt edicilik gücü olan terimler diğer terimler ile ayrışmaz. Tüm terimler eşit ağırlığa ve aynı özelliğe sahiptir. Terimlerin ayırt edicilik gücünden faydalanmak için terimlere ağırlık uygulanmaktadır. Ağırlıklandırma yöntemlerinden en çok bilinen $tf \times idf$ formülüdür. Bir terimin bir dokümana olan bu ağırlık formülünde tf bir terimin doküman içi frekansı ve idf ise terimin ters doküman frekansıdır. *D* dokümanı ve *N* tane dokümandan oluşan koleksiyon için, bir *t* teriminin *D* dokümanına olan ağırlığı Denklem (2.3) ile hesaplanır. Denklemde $df(t)$ terimin koleksiyondaki doküman frekansıdır.

$$weight(t,D) = tf_{t,D} \times idf_t$$

$$idf_t = \log_2 \frac{N}{df(t)} \quad (2.3)$$

Denklem (2.4)'de görüldüğü gibi bir terim dokümanda ne kadar çok görünüyorsa ona olan ağırlığı o kadar fazladır. Ancak $tf_{t,D}$ değerinin doğrusal bir fonksiyona sahip olduğu görülmektedir. Bir dokümanda iki defa görülen bir terim bir defa görülen başka bir terime göre iki katı oranda ağırlığa sahiptir. Bu oranda bir farkın oluşmaması için $tf_{t,D}$ Denklem (2.4)'deki gibi kullanılabilmektedir.

$$tf_{t,D} = \begin{cases} \log(tf_{t,D}) + 1, & tf_{t,D} > 0 \\ 0, & tf_{t,D} = 0 \end{cases} \quad (2.4)$$

2.3.3. Olasılıksal model

Bilgi erişim sisteminde verilen bir sorgu için getirilen dokümanlar, sorgu ile dokümanlar arasındaki alakalı olma olasılığına *P(R|D)* göre azalarak sıralanırlar. Bu sıralama ile olasılığı yüksek olan dokümanlar en üst sırada yer aldığı için sistemin genel performansı maksimize edildiği düşünülür ve Probability Ranking Principle (PRP)

(Olasılık Sıralama Prensibi) (Robertson S. , 1977) olarak adlandırılır. Bu prensibin formüle edilmiş hali (Denklem (2.5)) Robertson ve Jones (1976) tarafından sunulmuştur.

$$S(Q,D) = \sum_{t \in Q \cap D} \frac{P(D_t = 1|R) \cdot P(D_t = 0|\bar{R})}{P(D_t = 0|R) \cdot P(D_t = 1|\bar{R})} \tag{2.5}$$

Denklemde R ve $\bar{R}$ "alakalı" ve "alakasız" değerlerini alan rastgele değişkenlerdir ve $P(R)$ ve $P(\bar{R})$ ise bu değişkenlerin olasılıklarını ifade etmektedir. D_t {0,1} değerlerini alan bir değişkendir. $D_t = 1$, mevcut terimin D dokümanında göründüğünü ve $D_t = 0$ ise dokümanda görünmediğini göstermektedir. $P(D_t = 1|R)$, alakalı dokümanlarda D dokümanının t terimini içerme olasılığıdır. Literatürdeki en çok bilinen Okapi-BM25 terim ağırlıklandırma modeli 2-Poisson olasılık dağılımı modeli kullanmaktadır (Robertson & Walker, 1994; Robertson & Zaragoza, 2009). Okapi-BM25 ile bir Q sorgusunun D dokümanına olan puanı Denklem (2.6)'de olduğu gibi hesaplanmaktadır.

$$S(Q,D) = \sum_{t \in Q \cap D} IDF(t) \cdot \frac{(k_1 + 1) \cdot tf_{t,D}}{k_1 \left(1 - b + b\left(\frac{|D|}{avgdl}\right)\right) + tf_{t,D}} \tag{2.6}$$

Denklem k_1 ve b olmak üzere iki serbest parametreden oluşmaktadır. Parametre $k_1$terim frekansını ve b doküman uzunluğunu normalize etmektedir.

2.3.4. Dil modeli

Dil modelinde, her bir dokümanın kendine ait bir dil modeli $M_d(t)$ bulunmaktadır. Dil modeli, her bir doküman için dil modeli üretmeye ve üretilen dil modeline göre verilen sorgunun oluşturulma olasılığını hesaplamaya dayanır. Bilgi erişim sisteminin getirdiği dokümanlar, hesaplanan bu olasılıklara göre sıralanır (Ponte & Croft, 1998; Hiemstra, 2000; Lafferty & Zhai, 2001). Denklem (2.7) bir sorgunun bir dokümana olan olasılığını hesaplar.

$$p(Q|D) = \prod_{t \in Q} p(t|D)$$

$$p(t|D) = \frac{tf_{t,D}}{|D|}$$

$$p(Q|D) = \prod_{t \in Q} M_d(t) \tag{2.7}$$

Sorgu terimlerinin dokümanda görülmediği durumda mevcut denklem olasılık olarak sıfır değeri hesaplanır. Pürüzsüzleştirmenin (smoothing) temel amacı böyle bir durumda sıfır olmayan bir olasılık değeri atamaktır. Dirichlet (Zhai & Lafferty, 2004) ve Jelinek-Mercer (Jelinek, 1980) bu amaçla sıklıkla kullanılan yöntemlerdir.

2.3.5. Rastlantısallıktan farklılık modeli (Divergence from randomness)

Rastlantısallıktan farklılık modeli (Divergence From Randomness (DFR)) (Amati & van Rijsbergen, 2002) parametrik olmayan bir terim ağırlıklandırma modelidir. Mevcut terim dağılımının rastgele bir işlemle elde edilen dağılımdan farklılığını baz alır. Bir dokümandaki önem derecesi yüksek olan terimler rastlantısal model tarafından üretilen frekanstan farklılaşmasına dayanır. Dokümanlara puan atama işleminde, terim ağırlıklandırma yapılırken terim frekanslarına iki tür normalizasyon uygulanır. Birinci normalizasyonda dokümanlar aynı uzunluğa sahip olduğu varsayılır. Ayrıca mevcut doküman için terimin iyi bir açıklayıcı veya tanımlayıcı terim olduğu varsayılarak bilgi kazancı ölçülür. İkinci normalizasyonda ise doküman uzunlukları ile ilgili normalizasyon işlemleri ele alınır.

$$S(Q,D) = \sum_{t \in Q} (1 - P_2) \cdot (-\log P_1) \tag{2.8}$$

Denklem (2.8) bir dokümanın puanı belirlenirken kullanılan formülün genel yapısını vermektedir. P_1, verilen bir terim için rastlantısal modelinden elde edilen olasılıktır. Rastlantısal model olarak Poisson dağılımı, binom dağılım, Bose-Einstein istatistiği, ters doküman frekansı gibi modeller kullanılabilmektedir. P_2 ise terimin mevcut bir doküman için ne kadar iyi bir tanımlayıcı olduğunu ifade eder. Bu işlem için Bernoulli ilkesi veya Laplace kuralı uygulanabilir. İkinci normalizasyonda ise normalize terim frekans dağılımları için Denklem (2.9) uygulanır. Denklemde avg_l bir koleksiyondaki ortalama doküman uzunluğunu; l ise mevcut dokümanın uzunluğunu ifade etmektedir.

$$tfn = tf \cdot \log_2 \left(1 + c \cdot \frac{avg\_l}{l}\right) \tag{2.9}$$

$$S(Q,D) = \sum_{t \in Q} \frac{1}{tfn + 1} \left(tfn \cdot \log_2 \frac{tfn}{\lambda} + (\lambda - tfn) \log_2 e + 0.5 \log_2 (2\pi \cdot tfn)\right) \tag{2.10}$$

PL2 rastlantısallıktan farklılık modellerinden biridir ve Denklem (2.10)'da formülü verilmiştir. Denklemde λ Poisson dağılımının varyansını ve ortalamasını vermektedir. Bir terimin bir dokümandaki ortalama frekansını ifade etmektedir. PL2 terim ağırlıklandırma, Poisson dağılımı ile Laplace ilkesini birlikte kullanan ve terim frekansını normalize eden bir modeldir. DLH ve DPH terim ağırlıklandırma modelleri (Amati G. , 2006) hipergeometrik dağılıma sahip ve parametresiz DFR modelleridir. DLH, Laplace; DPH ise Popper normalizasyonunu kullanmaktadır.

2.3.6. Bağımsızlıktan farklılık modeli (Divergence from independence)

Parametrik olmayan bir terim ağırlıklandırma modeli olan bağımsızlıktan farklılık modelinde (Divergence from independence (DFI)) önem derecesi yüksek terimler kullanılan bağımsızlık modelinin ürettiği frekans değerinden farklılaşmasına dayanır (Kocabaş, Dinçer, & Karaoğlan, 2014). DFI modeli Pearson's Ki-Kare istatistiğinden faydalanarak bağımsızlıktan ne ölçüde farklılaştığını ölçer. Bir terimin bir dokümanda beklenen terim frekansı (e_t) Denklem (2.11)'de olduğu gibi hesaplanır. TF_t terimin koleksiyon frekansını, N koleksiyonun büyüklüğünü (koleksiyondaki toplam terim sayısı) ve $|D|$ dokümanın uzunluğunu ifade eder.

$$e_t = TF_t \cdot \frac{|D|}{N} \tag{2.11}$$

DFI modeli ile doymuş bağımsızlık modeli (saturated model of independence) (Denklem (2.12)), normalize Ki-Kare uzaklığı temelli bağımsızlık modeli (normalized chi-squared distance from independence) (Denklem (2.13)) ve standardize edilmiş (standardization) (Denklem (2.14)) uzaklık olmak üzere üç farklı ölçüm yapılabilir.

$$Saturated\ model\ of\ independence: DFIB = \log_2\left(\frac{tf_{t,D} - e_t}{e_t} + 1\right) \tag{2.12}$$

$$Normalized\ chi - squared\ distance : DFIC = \log_2\left(\frac{(tf_{t,D} - e_t)^2}{e_t} + 1\right) \tag{2.13}$$

$$Standardization: DFIZ = \log_2\left(\frac{tf_{t,D} - e_t}{\sqrt{e_t}} + 1\right) \tag{2.14}$$

2.3.7. Bilgi-tabanlı model

Doküman ve koleksiyon düzeyinde bir terimin karakterindeki farklılık doküman için terimin önem derecesi hakkında bilgi üretir. Clinchant ve Gaussier (2010; 2011) bilgi erişimi terim ağırlıklandırma modellerine yeni bilgi-tabanlı modeller önermiştir. En çok bilinen ve başarılı modellerinden biri LGD modelidir. Bu model normalize terim dağılımları için log-logistic dağılımı temel alır.

$$LGD(Q,D) = \sum_{t \in Q} -\log\left(\frac{\lambda_t}{\lambda_t + tfn}\right) \tag{2.15}$$

Denklem (2.15) LGD terim ağırlıklandırma modelini vermektedir. Denklemde normalize terim frekansı PL2'de kullanılan Denklem (2.9) ile aynı normalizasyonu kullanmaktadır. λ_t, koleksiyonda t teriminin olasılık dağılımının bir parametresidir. Bu parametre koleksiyonda t teriminin ortalama görünme frekansı ya da terimin göründüğü ortalama doküman frekansıdır.

2.4. Sorgu Sonuçlarının Yeniden Sıralanması

Bilgi erişim sisteminde terim ağırlıklandırma modellerinin dokümanlara puan atamasından bahsedilmiştir. Bu aşamadan sonra bir terim ağırlıklandırma modeli (BM25, DFIC, PL2, LGD vs.) dokümanları atadıkları puana göre sıralar ve bu sıralı listenin top-*K* dokümanı geri getirilir. Bilgi erişim sürecinin son aşaması geri getirilen sonuçlarının yeniden sıralanmasıdır. Yeniden sıralama işlemi çeşitli makine öğrenmesi teknikleri kullanılarak yapıldığı için Learning-to-Rank (LTR) olarak bilinir (Liu, 2009). Geri getirilen top-*K* dokümandan elde edilen öznitelikler ile öğrenilmiş bir model oluşturulur ve bu model geri getirilen dokümanları yeniden sıralamada kullanılır (Macdonald, Santos, & Ounis, 2013).

Öğrenilmiş model oluşturulurken çeşitli türde öznitelikler kullanılmaktadır. Bunlardan bazıları sorgulardan bağımsız sadece dokümanlardan elde edilmiş özniteliklerdir (Entropi, URL derinliği, SpamScore, Favicon, Keywords vs.). Bazıları ise sorgu terimlerinden oluşan öznitelikler olabilmektedir (Gamma, Omega, AvgPMI, SCS vs.) (Macdonald C. , Santos, Ounis, & He, 2013; Bendersky, Croft, & Diao, 2011; Cormack, Smucker, & Clarke, 2011; Hauff, Azzopardi, & Hiemstra, 2009; He & Ounis, 2006; Aydın, Göksel, Arslan, & Dinçer, 2020).

Bilgi erişimi alanında LTR günümüzde oldukça ilgi görmektedir. Bu alanda artan oranda araştırmalar yapılmakta, yöntemler geliştirilmekte ve kıyaslanmaktadır (Tax, Bockting, & Hiemstra, 2015).

2.5. Bilgi Erişim Sistemlerinin Değerlendirilmesi

Bilgi erişim sistemlerinin değerlendirilmesindeki temel amaç, bir bilgi erişim sisteminin kendisinden beklenen performansı ne kadar yerine getirdiğinin ölçülmesidir. Performansın değerlendirilmesi kullanıcıların bilgi gereksiniminin ne ölçüde karşılandığı ile ilgilidir (Voorhees E. , 2001). Bilgi erişimi alanında performans ölçümünü standart bir hale getirmek için çeşitli konferanslarda veya forumlarda test koleksiyonları oluşturulmuştur. Bu konferansların önde gelenleri Text REtrieval Conference (TREC, http-1), Cross Language Evaluation Forum (http-2) ve NII Testbeds and Community for Information access Research (NTCIR, http-3) organizasyonlarıdır.

Bilgi erişim sistemlerinin değerlendirilmesinde Cranfield Paradigması benimsenir. Cranfield Paradigmasına göre bir test koleksiyonunun üç temel bileşeni bulunmaktadır. Bunlardan birincisi bir grup dokümanlar, ikincisi bilgi gereksinimleri (sorgular) ve sonuncusu ise her bir sorgu için dokümanların alaka yargılarıdır (query relevance judgments) (Voorhees E. , 2007; 2019). Test koleksiyonları ile yapılan soyut bilgi erişim operasyonları ile farklı bilgi erişim stratejilerinin değerlendirilmesi mümkün kılınmıştır.

Test koleksiyonları genel olarak iki tür dosya ile yayınlanır. Bunlardan birinci bilgi gereksinimlerinin olduğu genelde XML formatında olan dosya, diğeri ise sorgu alaka yargılarının bulunduğu formatlı bir metin dosyasıdır. Şekil 2.4. bir bilgi gereksinimi örneğini vermektedir. Şekil 2.5. ise sorgu alaka yargılarının bulunduğu dosyadan bir örnektir. Bu dosyada dört adet veri alanı bulunmaktadır (http-4). İlk sütun sorgu numarası (TOPIC), ikinci sütun kullanılmayan ve genelde 0 değerini alan (ITERATION), üçüncü sütun ilgili koleksiyondaki dokümanların doküman numarası (DOCUMENT#) ve son sütun alakasız dokümanlar için 0, alakalı dokümanlar için 1 ve üzeri değer alan bir veri alanından (RELEVANCY) oluşmaktadır. Ayrıca bilgi erişim sistemi tarafından getirilen her doküman sorgu alaka yargılarında bulunmayabilir. Milyonlarca dokümanın olduğu koleksiyonda, her bir sorgu için tüm dokümanları alakalı ve alakasız olarak işaretlemek mümkün değildir. Dokümanları işaretleme işi, iş gücü yüksek bir operasyondur. Dolayısıyla her doküman sorgu alaka yargılarında bulunmaz. Seçilen dokümanlar ise birden fazla bilgi erişim sisteminin getirdiği ortak dokümanlardan oluşmaktadır. Bilgi

erişim sistemi tarafından getirilen bir doküman bu yargılardaki dokümanlar arasında bulunmuyorsa, o doküman alakasız olarak kabul edilir.

```
<topic number="266" type="single">
    <query>
        symptoms of heart attack
    </query>
    <description>
        What are the symptoms of a heart attack in both men and women?
    </description>
</topic>
```

Şekil 2.4. *Bir bilgi gereksinimi örneği*

TOPIC	ITERATION	DOCUMENT#	RELEVANCY
266	0	clueweb12-0202wb-03-27470	2
266	0	clueweb12-0202wb-27-23569	0
266	0	clueweb12-0202wb-60-22297	2
266	0	clueweb12-0202wb-67-33253	1
266	0	clueweb12-0203wb-45-20370	0
266	0	clueweb12-0203wb-60-15441	-2
266	0	clueweb12-0203wb-76-25161	2
266	0	clueweb12-0204wb-23-10546	-2
266	0	clueweb12-0204wb-89-07849	3

Şekil 2.5. *Sorgu alaka yargıları örneği*

Şekil 2.5. ile verilen sorgu alaka yargıları örneğinde alaka etiketleri (son sütun) sadece 0 veya 1 değerlerini alabildiği gibi; şekilde olduğu gibi farklı değerlerde alabilir.

-2 ile belirtilen değer dokümanın spam olduğunu ifade eder. 0 değeri alakasız, 1 ve üzeri ise alakalı düzeyinin artan oranda olduğunu göstermektedir.

QID	ITER	DOC_NO	RANK	SIM	RUN_ID	TAG
300	Q0	clueweb12−1712wb−85−10084	1	4.7044	BM25k1.2b0.75	KStem
300	Q0	clueweb12−0102wb−39−28701	2	4.6279	BM25k1.2b0.75	KStem
300	Q0	clueweb12−0307wb−58−22743	3	4.4540	BM25k1.2b0.75	KStem
300	Q0	clueweb12−0109wb−82−21760	4	4.4472	BM25k1.2b0.75	KStem
300	Q0	clueweb12−0307wb−28−01469	5	4.3891	BM25k1.2b0.75	KStem

Şekil 2.6. *TREC formatında hazırlanan gönderim dosyasından bir örnek*

Bir sistemin değerlendirilebilmesi için TREC formatında bir gönderim dosyası oluşturulur. Bu dosya sorgu alaka yargı dosyasına benzemektedir. Değerlendirilen bilgi erişim sistemi tarafından her bir sorgu için top-*N* (genelde 1,000) doküman geri getirilir. Her bir sorgu için geri getirilen dokümanlar alaka olasılıklarına göre azalan sıra ile belli bir formatta dosyaya yazılır. Şekil 2.6. ile TREC formatında hazırlanan gönderim dosyasından bir örnek verilmiştir. İlk sütun bilgi gereksiniminin sorgu numarasını (QID) vermektedir. İkinci sütun (ITER) kullanılmayan ve "Q0" girilen sabit bir alandır. Üçüncü sütun (DOC_NO) geri getirilen dokümanın numarasıdır. Dördüncü sütun (RANK) dokümanın kaçıncı sırada getirildiğini ifade eder. Beşinci sütun (SIM) dokümanın alaka puanını ve son iki sütun ise sırasıyla kullanıcı tarafından verilen geri getirim stratejisi ile etiketini göstermektedir.

2.5.1. Bilgi erişim sistemleri performans ölçüm araçları

Bilgi erişim sistemlerinin performans değerlendirilmesinin yapılabilmesi için TREC tarafından standart bir performans ölçüm aracı sunulmuştur. Bu alanda çalışan araştırmacılar için ortak bir aracın olması performans ölçümlerinin karşılaştırılabilmesi için önemlidir.

Bu amaçla oluşturulan ilk ölçüm aracı `trec_eval` (http-5), bir bilgi erişim sisteminin ürettiği sonuç listesi ile sorgu alaka yargılarını (`qrels`) kullanır. Aracın ürettiği çıktı oldukça geniş bir yelpazede performans ölçümleri yapar.

Diğer bir araç ise `gdeval.pl` (http-6), nDCG@*k* ve ERR@*k* ölçülerinin hesaplanması için kullanılan bir araçtır.

Son olarak `statAP_MQ_eval_v4.pl`, TREC Million Query için sunulmuş bir araçtır (http-7). Bu ölçüm aracının `trec_eval` aracından farkı; `trec_eval` dört sütunlu `qrels` dosyası alırken, `statAP_MQ_eval_v4.pl` bir sütun fazla bir alana sahiptir ve sırası ile şu alanları içerir (Allan, ve diğerleri, 2007):

- Topic ID
- Document ID
- Judgement
- Algorithm
- Inclusion Probability

2.5.2. Bilgi erişim sistemleri performans ölçüleri

Bilgi erişim sistemlerinin performans ölçümü için literatürde çeşitli ölçüler sunulmuştur. İlk performans ölçümleri küme tabanlı olan *Kesinlik* (*Precision*) ve *Duyarlılık* (*Recall*) üzerinden yapılmıştır. Kesinlik; bilgi erişim sisteminin getirdiği dokümanlarındaki alakalı doküman oranıdır. Duyarlılık ise; getirilen alakalı dokümanların koleksiyondaki alakalı dokümanlara oranıdır.

Bilgi erişimi performans ölçümlerinde, dokümanları dört grupta düşünebiliriz:

1. Alakalı Dokümanlar
2. Alakasız Dokümanlar
3. Sistem Tarafından Getirilen Dokümanlar
4. Sistem Tarafından Getirilmeyen Dokümanlar

Bu doküman gruplarını bir *olumsallık tablosunda* (*contingency table*) ifade ettiğimizde ve bunu bir sınıflandırma problemi gibi ele aldığımızda Tablo 2.1 ile verilen olumsallık tablosunu elde ederiz.

Tablo 2.1. *Olumsallık tablosu*

	Alakalı	**Alakasız**
Getirilen	gerçek pozitif (*tp*)	yanlış pozitif (*fp*)
Getirilmeyen	yanlış negatif (*fn*)	gerçek negatif (*tn*)

Tablodan yararlanarak elde edilen kesinlik formülü Denklem (2.16) ve Denklem (2.17) ile ifade edilmiştir. Kesinlik bilgi erişim sisteminin getirdiği dokümanlardaki alakalı doküman sayısının oranıdır. Duyarlılık ise koleksiyondaki alakalı dokümanlardan ne kadarı sistem tarafından getirildiğinin bilgisidir.

$$\text{precision} = \frac{tp}{tp + fp}$$

$$\text{precision} = \frac{\#\ getirilen\ alakalı\ belgeler}{\#\ getirilen\ belgeler} \tag{2.16}$$

$$\text{recall} = \frac{tp}{tp + fn}$$

$$recall = \frac{\#\ getirilen\ alakalı\ belgeler}{\#\ koleksiyondaki\ alakalı\ belgeler} \tag{2.17}$$

Duyarlılık, bilgi erişim sisteminin getirdiği doküman sayısı arttıkça artar. Sebebi ise duyarlılık tüm koleksiyondaki alakalı doküman sayısına bağlıdır. Duyarlılığın "1" olması için tüm koleksiyondaki tüm alakalı dokümanların sonuç listesinde bulunması gerekir. Ancak böyle bir sistem büyük veri setleri için oldukça güç bir durumdur.

Kesinlik ve duyarlılık ölçülerinin birleştirilerek tek bir ölçü olarak kullanıldığı ölçü *F_1 puanıdır* (*F-puanı veya F ölçüsü*). F_1 puanı kesinlik ve duyarlık değerlerinin harmonik ortalamasıdır ve bu değerler Denklem (2.18)'de eşit ağırlığa sahiptir.

$$F_1\text{puanı} = \frac{2 \times precision \times recall}{precision + recall} \tag{2.18}$$

Küme tabanlı olan kesinlik ve duyarlılık ölçümleri, getirilen dokümanların sırasının önemli olduğu bilgi erişim sistemlerinde performansı gerektiği gibi yansıtamamaktadır. Örneğin R_1 = {0,0,0,1,1} ve R_2 = {1,1,0,0,0} olmak üzere 2 farklı sonuç listesidir. Listede "0" alakasız "1" ise alakalı dokümanları ifade etmektedir. Her iki sonuç listesi için kesinlik değerleri 0.4'dür. Ancak R_2 sonuç listesinde alakalı dokümanlar ilk sırada gelmektedir. Literatürde sıralamayı da hesaba katan ve bilgi erişim sistemlerinin performansında kullanılan sıralama tabanlı ölçüler vardır.

2.5.2.1. Mean Average Precision (MAP)

Ortalama kesinlik (Average Precision (AP)), verilen bir sorgu için her bir alakalı doküman alındıktan sonra top-*k* doküman seti için elde edilen kesinlik değerlerinin

ortalamasıdır. *Average precision,* precision/*recall* eğrisinin altında kalan alana olan yaklaşık değerdir. Genel olarak Denklem (2.19) deki gibi hesaplanır.

$$AP = \frac{1}{|Rel|} \cdot \sum_{i=1}^{k} rel(i) \times P@i \tag{2.19}$$

AP genel olarak top-1000 doküman üzerinden tek bir sorgu için hesaplanır. Bilgi erişim sistemlerinin performans ölçüme tek bir sorgu üzerinden değil; bir grup sorgu üzerinden (Q) hesaplanır. Sorgu setindeki her bir sorgu için hesaplanan AP değerlerinin ortalaması ise *Mean Average Precision (MAP)* (Denklem (2.20)) ölçümünü vermektedir.

$$MAP = \frac{1}{|Q|} \cdot \sum_{q \in Q} AP(q) \tag{2.20}$$

2.5.2.2. Normalized Discounted Cumulative Gain (NDCG)

Bilgi erişimi alanında en güncel ve kabul görmüş metriklerden biridir. Top-*k* sonuç listesi üzerinden hesaplanan metrik derecelendirilmiş bir alaka düzeyi ölçeği kullanır (Järvelin & Kekäläinen, 2002). Diğer bir deyişle, listenin alt sıralarına inildikçe alakalı dokümanların sorgu için bilgi erişim sistemi performansına olan etkisi logaritmik olarak azalır. *DCG* top-*k* doküman için Denklem (2.21)'deki gibi hesaplanır. Denklemde g_i, *i.* sıradaki dokümanın alaka derecesidir.

$$DCG@k = \sum_{i=1}^{k} \frac{2^{g_i} - 1}{\log_2(1+i)} \tag{2.21}$$

nDCG@k ise *DCG*'nin normalize edilmiş halidir ve *idealDCG@k* bölünmesi ile elde edilir (Denklem (2.22)). *idealDCG* alakalı belgelerin alaka düzeylerine göre büyükten küçüğe göre sıralanmasıyla elde edilebilecek maksimum DCG değerini ifade eder. Diğer bir ifadeyle oluşturulabilecek en mükemmel sıralamadan elde edilecek DCG değeridir.

$$nDCG@k = \frac{DCG@k}{idealDCG@k} \tag{2.22}$$

2.5.2.3. Expected Reciprocal Rank (ERR)

Bilgi erişim sistemlerinde kullanıcının gezinme davranışları çeşitli faktörler tarafından belirlenir. Örneğin bir kullanıcı sıralı bir listedeki *i.* dokümanı seçmesi

kullanıcının bu dokümandan önceki dokümanlar hakkında bilgi ihtiyacını tatmin edecek seviyede karşılayıp karşılamadığı hakkında bilgi verir. ERR (Chapelle, Metlzer, Zhang, & Grinspan, 2009) bu davranışı göz önünde bulunduran bir metriktir (Denklem (2.23)). Denklemde g_i, top-*k* doküman için *i.* dokümanın alaka derecesini ifade etmektedir.

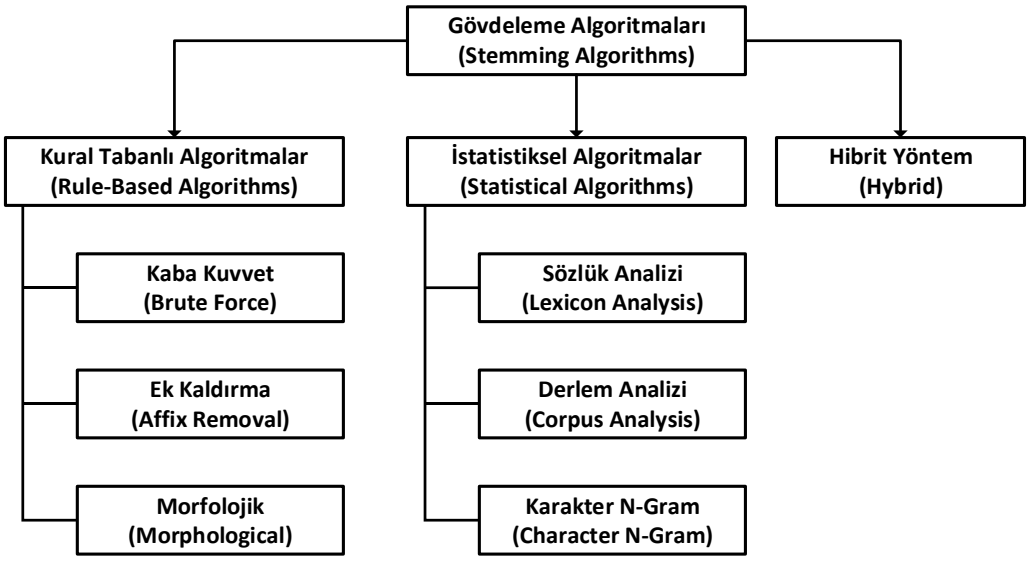
$$ERR@k = \sum_{i=1}^{k} \frac{R(g_i)}{i} \prod_{j=1}^{i-1} (1 - R(g_i)) \tag{2.23}$$

$$R(g) = \frac{2^g - 1}{2^{g_{max}}}$$

3. GÖVDELEME ALGORİTMALARI

Bir kelime, cümle içinde aldığı göreve göre çeşitli formlarda görülebilmektedir. Dil yapısına göre kelimeler ön-ek, son-ek alarak veya farklı dil kuralları ile bir biçimden farklı biçimlere dönüşebilmektedir. Ancak bu farklı formdaki kelimeler benzer anlamasal ilişkiye sahip olabilir. Örneğin İngilizcedeki "works", "worked", "working" ve "worker" kelimelerinin kökü "work" kelimesidir. Diğer bir örnek ise "taken", "taking" ve "took" kelimelerinin kökü "take" kelimesidir. Türkçe kelimelerden örnek verecek olursak "geliyorum", "geliyorsun", "geliyor", "geliyoruz", "geliyorsunuz" ve "geliyorlar" kelimelerinin kökü "gel" kelimesidir. Örneklerden de anlaşılacağı gibi gövdeleme, bir kelimenin morfolojik türevlerinin o kelimenin köküne indirgenmesi olarak adlandırılabilir.

Gövdeleme algoritmaları literatürde oldukça zengin bir yere sahiptir ve çeşitli yöntemler ile birçok gövdeleme algoritmaları sunulmuştur. Basit bir yöntem olan çoğul eklerinin kaldırılmasından, derlemden çıkarılan istatistiklere göre son-eklerin kaldırılmasına kadar kompleks algoritmalar mevcuttur. Gövdeleme algoritmaları genel olarak iki gruba ayrılır ((Paik J. H., Parui, Pal, & Robertson, 2013; Brychcín & Konopík, 2015)): (*i*) Kural tabanlı ve (*ii*) istatistiksel algoritmalar. Ayrıca Singh & Gupta (2016) yaptığı çalışmada (*iii*) hibrit yaklaşımlardan da bahsetmiştir ve Şekil 3.1. Gövdeleme algoritmalarının sunulmuştur.



Şekil 3.1. *Gövdeleme algoritmalarının taksonomisi*

3.1. Kural Tabanlı Algoritmalar

Kural tabanlı gövdeleme algoritmaları bir terimin türevini kök formuna çevirmek için terimin dil yapısının kurallarına göre tanımlanmış gövdeleme kuralları kullanır. Bu kurallar dil bilim konusunda uzman kişiler tarafından oluşturulmasına ihtiyaç duyar. Kurallar ile birlikte sözlük gibi ek kaynaklarda bu tür gövdeleme algoritmalarında kullanılabilir.

Kural tabanlı algoritmaların kullanımı kolaydır ve oldukça hızı sonuç üretirler. Ayrıca belirlenmiş kurallar ile çalışması onların derlemden bağımsız olmalarını ve dil yapısına uygun her derlemde çalışma imkânı vermektedir.

3.1.1. Kaba kuvvet (Brute force) algoritmaları

Kaba kuvvet algoritmaları bir arama tablosu kullanarak verilen bir kelimenin kökünü tablodan geri döndürür. Bir kelimenin kökünü bulmak için o kelime tabloda olup olmadığına bakılır, eğer tabloda bulunursa kök algoritma tarafından üretilmiş olunur. Dil kurallarına uymayan kelimeler için bu algoritma kullanışlıdır. Örneğin, bir son-ek kaldırma algoritması ile İngilizcedeki “eating” kelimesi “eat” kelimesine dönüştürülebilir. Ancak aynı kelimenin bir başka biçimi olan “ate” son-ek algoritması ile kökü belirlenemez. Kaba kuvvet algoritmaları ile kök bulmak oldukça zordur. Çünkü böyle bir tablonun oluşturulması büyük oranda iş yüküne ve bilgisayar hafızasına ihtiyaç duyar.

3.1.2. Ek kaldırma (Affix removal) algoritmaları

Ek kaldırma algoritmaları ön-ek veya son-eklerin kelimden kaldırarak gövdeleme yapma yöntemidir. Algoritmanın temelinde ön-ek veya son-ek listeleri kullanılarak belirli kurallar çerçevesinde kelimenin kökünün bulunması yatar. Sondan eklemeli diller düşünüldüğünde son-ek kaldırma yöntemleri ön-ek kaldırma yöntemlerine kıyasla daha çok tercih edilir. Belirli kurallar ile kaldırılan ekler ile üretilen kök her zaman dile uygun gerçek bir kelime olmayabilir. Ayrıca yapılan ek kaldırmalar ile oldukça agresif birleşim kümeleri (aynı köke ait kelimelerin oluşturduğu küme) oluşabilir (Baeza-Yates & Ribeiro-Neto, 2011).

3.1.3. Morfolojik (Morphological) algoritmalar

Morfolojik gövdeleme algoritmaları dilin çekim ve yapım eklerinin morfolojik analizlerini kullanarak gövdeleme yapmaktadır. Bu kapsamda yayınlanmış ilk çalışmalardan biri Lovins (Lovins, 1968) gövdeleme algoritmasıdır. Krovetz (Krovetz, 1993) tarafından geliştirilen KStem ise bu kapsamda geliştirilmiş önemli algoritmalardan birisidir. Diğer bir algoritma ise Porter (Porter, 1997) gövdeleme algoritmasıdır. Ayrıca Porter bu algoritma ile *Snowball* (http-8) isminde bir yazılım çerçevesi geliştirmiştir.

3.2. İstatistiksel Algoritmalar

İstatistiksel gövdeleme algoritmaları bir dildeki derlemden üretilecek gövdeleme kurallarını öğrenmede gözetimsiz veya yarı gözetimli eğitimleri kullanır. Derlemi kullanarak, morfolojik olarak birbirleri ile ilişkili kelimeleri gruplar. Bu yöntem dil bilim konusunda uzman bilgisi veya dil ile ilgili herhangi bir kaynak gerektirmez. Bu sebeple yeni bir dile daha az iş yükü ile uyarlanabilir.

3.2.1. Sözlük analizi (Lexicon analysis)

Sözlük analizine dayalı algoritmalar sözlük bilimsel olarak birbiri ile ilişkili kelimeleri gruplamak için derlemden elde edilen kelimeleri analiz eder. Çeşitli istatistiksel analizler ile olası kökleri ve son-ekleri belirler. Kelimelerin karakter dizileri arasındaki uzaklık (Majumder, ve diğerleri, 2007) veya alt karakter dizileri frekansları kullanılan (Paik & Parui, 2011) yöntemlerdendir. Derlemdeki kelimelerin son-ek çiftleri ile oluşturulan çizge (Paik, Mitra, Parui, & Järvelin, 2011), aynı köke sahip kelime kümelerini bulmayı amaçlamak için kullanılmıştır. $G = (V, E)$ çizgesi için V, düğüm, sözlükteki en uzun ortak ön-eke sahip kelimeler; E, kenar, ise düğümler arasındaki bağlantıları ifade eder. Örneğin, $kelime_1 = rs_1$, $kelime_2 = rs_2$ ve r kelime_1 ve kelime_2 için en uzun ortak ön-ek için $kelime_1$ ve $kelime_2$ komşu düğümlerdir. $<s_1, s_2>$ son-ek ikililerinin frekansları ise bu iki düğüm arasındaki ağırlıktır. Belli bir eşik değer üzerindeki komşuların oluşturduğu kelimeler aynı birleşim kümeleri yani aynı sınıfa aittir.

Varis çeşitliliği (successor variety) gövdeleme algoritması olarak sunulan yöntemlerden birinde; bir kelimede, herhangi bir konumdaki harfin kendisinden önceki harflere bağlı olması ve kelimenin sonuna doğru hareket ettikçe bağımlılığın artması fikrine dayanır (Hafer & Weiss, 1974). Ayrıca derlem üzerinden elde edilen istatistiksel

bilgilere göre gövde belirlenir. Terimin parçaları bir ağaç veri yapısı üzerinde ifade edilir. Derlemden elde edilen bilgiler ile her kelime için ek ve gövde sınırını belirleyen spesifik bir *k* değeri hesaplanır. *k* değerinin belirlenmesinde terimin varislerinin sayısına bir eşik değer uygulanır (Hafer & Weiss, 1974). Eşik değer, terimin ataları (kelimenin alt karakter dizilerine) ile hesaplanan entropi değerine (Al-Shalabi, Kannan, Hilat, Ababneh, & Al-Zubi, 2005; Hafer & Weiss, 1974) veya atanın derlemde görünme durumuna göre (Stein & Potthast, 2007) belirlenir.

3.2.2. Derlem analizi (Corpus analysis)

Kelime formlarının derlemde birlikte görünme olasılıklarını kullanan yöntemlerdir. Xu ve Croft (1998) çalışmasında, çekim eki almış kelimelerin birbirine yakın görünme kabullenmesine dayandırmıştır. Bu yakın görünme aynı doküman veya aynı metin penceresinde olabilir. Agresif bir gövdeleme algoritması ile aynı köke sahip kelime kümeleri belirlenir ve bu kümelerin içinde bulunan kelimeler beklenen birlikte görünme bilgisine (Expected Mutual Information) göre tekrar incelenir ve kümelerin son hali belirlenir. Paik, Pal ve Parui (2011) çalışmalarında benzer bir yaklaşım sunarak bir birleşim kümesindeki kelimelerin birlikte görünme bilgisini bir dokümandaki terim frekanslarına göre belirler. Bir sonraki çalışmada ise (Paik J. H., Parui, Pal, & Robertson, 2013) birlikte görünme bilgi için kelime çiftleri arasındaki ilişkiyi ve sorgu terimlerinin sahip olduğu konsepti göz önünde bulundurmayı temel alan yeni bir metrik sunmuştur. Gözetimsiz makine öğrenme ile derlem üzerinde sözlük analizi, semantik bilgi ve maksimum entropi sınıflandırıcısı kullanılarak morfolojik olarak birbiri ile alakalı kelimeler bulunmaya çalışılarak bir başka gövdeleme algoritması sunulmuştur (Brychcín & Konopík, 2015). Çalışmada kesinlik (precision) performansının yüksek tutulmasına önem verilmiştir. Bir başka gözetimsiz makine öğrenmesi ile yapılan derlem tabanlı gövdeleme algoritması (Singh & Gupta, 2019) çizge veri yapısı kullanarak sözlük ve son-ek analizleri ile kelimelerin morfolojik türevlerini bulmaya yönelik çalışma yapılmıştır.

3.2.3. Karakter *N*-Gram (Character *N*-Gram)

Bu tür gövdeleme algoritmaları gövdeleme kurallarını *n*-gram ile elde edilen kelime öbeklerinin frekans veya olasılıkları ile oluşturulur. *N*-gram ile oluşturulan gövdeleme algoritmaları ile çekim ekleri veya yapım eklerinin belirlenmesinin yanı sıra birlikte yan yana geçen kelimeleri göz önünde bulundurur ve yazım hatalarından oluşabilecek

durumları da yakalayabilmektedir (McNamee & Mayfield, 2004; Ahmed & Nürnberger, 2009).

3.3. Hibrit Algoritmalar

Birden fazla yaklaşımı birleştiren yöntemlere hibrit yaklaşımlar olarak tanımlanır. Hibrit yaklaşımlar gövdeleme çalışmalarında da yer almaktadır. Birden fazla kural tabanlı yaklaşımların veya bir kural tabanlı bir istatistik tabanlı yaklaşımın bileştirilmesi ile hibrit gövdeleme algoritmaları sunulabilmektedir (Weiss, 2005; Adam, Asimakis, Bouras, & Poulopoulos, 2010; Mishra & Prakash, 2012).

4. İLGİLİ ÇALIŞMALAR

Seçkili bilgi erişimi, verilen her bir sorgu için belirli bir bilgi erişim stratejisi uygulanmasıdır. Bu stratejiler, uygulanan tekniğe bağlı olarak erişim öncesi aşamadan erişim sonrası aşamaya kadar değişir. Bu bölüm, temel olarak, çok çeşitli tekniklerle ve özellikle gövdelemeye yönelik seçkili yaklaşımlarla ilgilenen seçim stratejileri üzerine literatürdeki çalışmaları gözden geçirmektedir.

4.1. Seçkili Bilgi Erişimi

Seçkili bilgi erişimine yönelik bazı çalışmalar erişim sistemi içindeki konfigürasyonları optimize etmeye yöneliktir (Bigot, Dejean, & Mothe, 2015; Deveaud, Mothe, Ullah, & Nie, 2018; Arora & Yates, 2019; Mothe & Ullah, 2021). Yapılan çalışmalarda indekslemeden dokümanların yeniden sıralamasına kadar uzanan sistem bileşenlerinin en uygun kombinasyonunu tahmin etmeyi amaçlamaktadır.

Bilgi erişim sistemin bir bileşeni olarak terim ağırlıklandırma modelleri de sorgu başına performansı etkiler. Bu nedenle, modeller arasında performans düşüşünü hafifletmek için terim ağırlıklandırma modeli seçimine yönelik seçkili yaklaşımlar sunulmaktadır (He & Ounis, 2003; He & Ounis, 2004a; He & Ounis, 2004b; Arslan & Dinçer, 2019).

Farklı bir çalışmada (Tonellotto, Macdonald, & Ounis, 2013) araştırmacılar seçkili bir budama yöntemi önermektedir. Çalışmaları, sonuç listesinin agresif bir şekilde budanması gereken sorguları belirler. Ayrıca, sorgu genişletme teknikleri, daha fazla doküman eşleştirerek bilgi erişiminde *recall* performansını artırmak için sorguya yeni terimler ekler. Ancak, herhangi bir sorgunun, bilgi erişim performansı açısından eklenen terimlerden pozitif yönde yararlanması her zaman geçerli değildir. Bu nedenle, sorgu genişletmenin uygulanıp uygulanmayacağını belirlemek için seçkili yaklaşımlar önerilmektedir (Amati, Carpineto, & Romano, 2004; Cronen-Townsend, Zhou, & Croft, 2004; Yom-Tov, Fine, Carmel, & Darlow, 2005; Peng, He, & Ounis, 2009; Peng, Macdonald, He, & Ounis, 2009; Hauff, Azzopardi, Hiemstra, & de Jong, 2010; Chen, Chunyan, & Xiaojie, 2012). Ayrıca sorguya eklenecek terimleri seçen yaklaşımlar da mevcuttur (Cao, Nie, Gao, & Robertson, 2008; Saleh & Pecina, 2019).

Bilgi erişimine yönelik diğer bir seçkili yaklaşım, doküman koleksiyonlarının indekslenmesiyle ilgilidir. Büyük ölçekli doküman koleksiyonları, *shard* olarak adlandırılan topikal olarak homojen gruplara bölünür. Arama işlemi, yalnızca belirli bir

sorgu için ilgili dokümanları içerdiği varsayılan birkaç shard üzerinde yürütülür ve bu işleme seçkili arama adı verilir. Bu strateji, doküman koleksiyonunun küçük bir dilimini kullanarak ve aynı zamanda tüm doküman koleksiyonunda gerçekleştirilen bir aramadaki kadar performansına sahip olarak, sorgu için bilgi erişim sisteminin maliyetini düşürmeyi amaçlar. Bu bağlamda, üzerinde arama yapılacak doküman kaynağını seçen algoritmalar ve teknikler önerilmiştir (Kulkarni & Callan, 2015; Kim, Callan, Culpepper, & Moffat, 2016; Kim, Callan, Culpepper, & Moffat, 2016; Chuang & Kulkarni, 2017; Kim, Callan, Culpepper, & Moffat, 2017; Siedlaczek, Rodriguez, & Suel, 2019).

Seçkili yaklaşımın başka bir uygulama alanı, geri getirilen dokümanları, öğrenilen bir modelle yeniden sıralayan Sıralamayı Öğrenme (Learning-to-Rank (LTR))'dir. Ağırlıklandırma modellerinde olduğu gibi, farklı sorgular her bir sıralama metodundan farklı şekilde yararlanır ve sorgu bazında uygun metoda karar vermek için seçkili yöntemler geliştirilmiştir (Peng, Macdonald, & Ounis, 2010; Balasubramanian & Allan, 2010; Lin, Yu, & Chen, 2011; Ghanbari & Shakery, 2019; Mansoub, Ercan, & Çiçekli, 2020). Literatürde yapılan çalışmalar göz önünde bulundurulduğunda, bilgi erişimine yönelik seçkili bir yaklaşım, sorgu bazında bilgi erişim sisteminin herhangi bir aşamasında benimsenebilir.

4.2. Seçkili Gövdeleme

Bilgi erişim alanında gövdeleme oldukça uzun bir geçmişe sahiptir. Bu alanda çeşitli yöntemler sunulması, literatürde zengin bir yer edinmesini sağlamıştır. Gövdeleme yöntemlerini önemli kılan şey, özellikle bilgi erişimi alanında, morfolojik olarak birbiri ile alakalı terimleri tek bir terimde (kökte) toplayarak sistemin *recall* değerini artırabilmesidir. Bu tez kapsamında seçkili gövdeleme; verilen her bir sorgu için sisteme ve sorguya gövdeleme uygulanıp uygulanmayacağına ön-işlem aşamasında karar verme yöntemidir.

Harman (1987) Cranfield koleksiyonu üzerinde bir seçkili gövdeleme çalışması yapmıştır. Bu çalışmada seçki yöntemi sorgu uzunluğuna dayanmaktadır. Gövdeleme sadece sorgu uzunluğu en çok dokuz terimden oluşan sorgulara uygulanmıştır. Ancak bu yöntem ortalama performansı geliştirememiştir. Ayrıca diğer bir çalışmasında (Harman, 1991) sorgudaki önemli terimler üzerinde seçkili gövdelemenin performansını incelemiştir. Bu çalışmanın sonucunda önemli terimler üzerine gövdeleme uygulandığında seçkili gövdelemenin başarısız olduğu sonucuna varmıştır. Farklı bir

çalışmada (Chin, DeCook, Street, & Eichmann, 2010) makine öğrenmesi tabanlı seçim yöntemleri önerilmiştir. Çalışmadaki yöntem, gövdeleme, çoğullaştırma metin normalleştirme ve gövdelemenin uygulanmaması teknikleri arasından uygun bir metin normalleştirme tekniği seçmek için sorgu öznitelikleri üzerine kurulu Destek Vektör Regresyon modellerini kullanır. Ancak bu çalışmada birden fazla teknik arasından seçim yapılmaya çalışılmıştır. Gövdelemenin uygulanıp uygulanmayacağına karar vermede genetik algoritmalar da kullanılmıştır. Sorgu performans öznitelikleri ile yapılan çalışmada genetik algoaritma yöntemleri ile bu öznitelikler kullanılarak bir formül elde edilmiş ve bu formül seçki mekanizmasında kullanılmıştır. Yapılan bu çalışmada seçkili gövdeleme için etkili öznitelikler belirlenememiştir (Wood, 2013). Verilen bir sorgunun diline göre seçkili gövdeleme gerçekleştirilen çalışmada (Macdonald, Lioma, & Ounis, 2007) ise tutarlı sonuç alınamamıştır.

Sorgu terimlerinin morfolojik türevleri ile sorgunun genişletilmesi de bir gövdelemenin bilgi erişim sistemlerine uygulanma çeşididir. Croft & Xu (1994) agresif bir gövdeleme algoritması ile oluşturulan birleşim kümelerini (conflation set) derlem analizi ile yeniden oluşturmuştur. Yaptığı çalışma ile agresif gövdeleme ile elde edilen morfolojik türevlerden hangilerinin sorgu genişletmede kullanılacağına böylelikle karar verir. Tudhope (1996) ise birleşim kümesindeki morfolojik türevlerin seçimini, sorgu terimi ile küme içindeki terimlerin anlamsal olarak yakınlığına göre yapmıştır. Diğer bir çalışmada (Cao, Robertson, & Nie, 2008) ise terimler dil modelleri uygulanarak sorgu ile aynı bağlamda olabilecek terimler seçilmiştir. Morfolojik türevlerin seçimine yönelik buna benzer çalışmalara literatürde mevcuttur (Peng, Ahmed, Li, & Lu, 2007; Paik J. H., Parui, Pal, & Robertson, 2013; Zervanou, Iosif, & Potamianos, 2014). Bu çalışmalara ek olarak son yıllarda kelime temsilleri (word embeddings) de morfolojik türevler arasındaki anlamsal yakınlığı ölçmek için kullanılmıştır (Basu, Roy, Ghosh, Bandyopadhyay, & Ghosh, 2017; Roy, Ghorai, Ghosh, & Ghosh, 2017).

4.3. Sunulan Seçkili Gövdelemenin Önceki Çalışmalardan Farkı

Bu yazıda sunulan seçkili gövdeleme yöntemi, aynı araştırma çizgisini takip ettiğimiz önceki çalışmaları (Harman, 1987; Harman, 1991; Chin, DeCook, Street, & Eichmann, 2010; Wood, 2013) önemli ölçüde genişletmektedir. Bu çalışmaların kısıtlarından biri de sorgu kümelerinin boyutudur: Test edilen sorgu sayısı, son zamanlarda literatüre kazandırılan toplam sorgu kümelerinden önemli ölçüde daha azdır.

Diğer bir kısıt ise karar yöntemleridir: Bazı çalışmalar bir makine öğrenme modeli oluştursa bile, modelde kullanılan özniteliklerin sorguları olabildiğince ayırt edebilmesi gerekir. Çalışmalardaki öznitelikler, gövdeleme uygulandığında oluşan terimdeki frekans dağılımlarındaki değişimleri genellikle kapsamaz. Ancak, gövdeleme ile bazı terimlerin frekansları diğer terimlere görece daha çok artabileceğinden, gövdeleme her terim için derlemdeki terim ve doküman frekanslarına aynı oranda etki etmez.

Bu çalışmada önerilen yöntemimiz, yalnızca sorgu performansı tahmin edici özniteliklerini değil, aynı zamanda gövdeleme uygulanmış ve uygulanmamış sistemler için terim frekanslarındaki artışlarda görülen dalgalanmadan türetilen öznitelikleri de kullanan bir ikili sınıflandırıcıdır. Yapılan araştırmalar sonucunda sahip olduğumuz bilgiye göre, çalışmamız sorgu terimlerinin özgüllüğündeki (Spärck Jones, 1972; Robertson S. , 2004; Church & Gale, 1999) değişimi göz önünde bulunduran ve kullanan öncü bir seçkili gövdeleme çalışmadır. Bu öznitelikler, terimlerin hem gövdelemeli hem de gövdelemesiz terim özgüllüğü farklılıklarından yararlanır. Sunulan seçkili yöntem, pratikte ve sıklıkla kullanılan bilgi erişim konfigürasyonları (kural tabanlı gövdeleme algoritmaları ve BM25 terim ağırlıklandırma modeli) ve çeşitli standart IR test koleksiyonları üzerinde değerlendirilir. Deneylerde kullanılan test koleksiyonları daha çok Web dokümanlarını ve ayrıca gazete dergilerini içeren dört kapsamlı doküman koleksiyonu üzerinde hazırlanmış sekiz sorgu setinden oluşur. Ayrıca, önerilen seçkili gövdeleme yönteminin gürbüzlüğünü belirlemek için riske duyarlı bir değerlendirme yapılmıştır. Bu değerlendirme, yöntemin ortalama performanstaki artışın sorgulara ne ölçüde yayıldığını gösterir.

5. DENEYSEL YÖNTEM

Önceki bölümlerde bilgi erişim sisteminin yapısından, performans ölçümünden ve gövdeleme algoritmalarından bahsedilmiştir. Bu bölümde sunmuş olduğumuz seçkili gövdeleme yönteminin performans ölçümünde ve yaptığımız analiz çalışmalarında kullandığımız deneysel yöntem sunulmuştur. Bu başlık altında; kullanılan veri setleri, sorgu setleri, terim ağırlıklandırma modelleri ve deneyleri gerçekleştirdiğimiz ortam olan Apache Lucene bilgi erişim çatısı detaylandırılmıştır.

5.1. Doküman Koleksiyonları

Sunulan seçkili gövdeleme yönteminin performans ölçümünde ve gövdeleme algoritmalarının analizinde oldukça zengin ve çeşitli doküman koleksiyonları kullanılmıştır. Doküman koleksiyonları çeşitli boyut ve içerikte olan TREC Web koleksiyonlarıdır. Ayrıca kullanılan bir koleksiyon ise gazete makalelerinden oluşmaktadır.

5.1.1. ClueWeb09

ClueWeb09 (http-9) bilgi erişimi alanında çalışan araştırmacıları desteklemek için oluşturulmuş Web dokümanlarından oluşan doküman koleksiyonlarından biridir. Koleksiyon yaklaşık olarak bir milyar Web dokümanı içermektedir ve yaklaşık 25TB büyüklüğündedir (Sıkıştırılmış hali 5TB'dır). Koleksiyonun oluşturulması 2009 Ocak ve Şubat ayları arasında gerçekleşmiştir. Koleksiyonda İngilizce dokümanların yanı sıra farklı dillerde de dokümanlar içermektedir. Tüm bu koleksiyon "Category A" olarak adlandırılmaktadır. "Category B" ise yaklaşık 50 milyon İngilizce dokümandan oluşan bir alt kategorisidir.

5.1.2. ClueWeb12

ClueWeb12 (http-10), ClueWeb09'da olduğu gibi Web dokümanlarından oluşmaktadır. Koleksiyon Şubat 2012 ve Mayıs 2012 tarihleri arasında toplanan 733,019,372 İngilizce Web dokümanından oluşmaktadır. "Category B" ise yaklaşık 50 milyon dokümandan oluşan bir alt kümesidir ve ClueWeb12-B13 ile ifade edilmektedir.

5.1.3. GOV2

GOV2 koleksiyonu 2004 yılında büyük oranda `.gov` alan adının içerdiği Web dokümanlarından oluşmaktadır. Koleksiyon Web dokümanlarının yanı sıra PDF, Word

ve Postscript dosyalarından elde edilmiş metinleri de içermektedir. Yaklaşık 25 milyon doküman içeren 426 GB büyüklüğünde bir koleksiyondur (http-11).

5.1.4. Wall Street Journal

TIPSTER projesi kapsamında üretilen derlem birden fazla doküman koleksiyonu içeren bir derlemdir (http-12). Bu derlemde Associated Press, Wall Street Journal, Department of Energy, Federal Register, Ziff-Davis Publishing ve San Jose Mercury News gibi kaynaklardan elde edilen dokümanlar içermektedir. Derlem 3 CD-ROM'dan oluşmaktadır ve Wall Street Journal birinci ve ikinci diskte bulunmaktadır. Disk-1'den 98,732 ve Disk-2'den 74,520 doküman olmak üzere Wall Street Journal derlemi toplam 173,252 doküman içermektedir (http-13).

5.2. Sorgu Setleri

Bilgi erişim sistemlerinin değerlendirilmesi için doküman koleksiyonlarına bağlı olarak test sorgu testlerine (bilgi ihtiyaçları) ihtiyaç duyulmaktadır. Sorgu setleri çeşitli konferanslarda veya çalıştaylarda katılımcılara bilgi erişim sistemlerinin performans değerlendirilmesi için yayınlanır.

5.2.1. Ad Hoc Track

TREC-1, TREC-2 ve TREC-3 (Harmon, 1996) kapsamında her yıl 50 sorgu olmak üzere toplamda 150 sorgu sunulmuştur. Bu sorgular Wall Street Journal'ın da bulunduğu TIPSTER Disk-1 ve Disk-2'deki doküman koleksiyonları için oluşturulmuştur.

5.2.2. Terabyte Track

TREC 2004, TREC-2005 ve TREC 2006 yıllarında bu kapsamda toplam 150 sorgu GOV2 koleksiyonu için oluşturulmuştur (Clarke, Craswell, & Soboroff, 2004). Buradaki amaç büyük boyutlardaki koleksiyonlar üzerinde performans ölçümü geliştirmektir. Ayrıca katılımcılar büyük boyutlardaki koleksiyonlar üzerinde bilgi erişim sistemlerinin ne ölçüde performans sergilediğini inceleyebilmektedir.

5.2.3. Million Query Track

TREC 2007 (Allan, ve diğerleri, 2007), TREC 2008 ve TREC 2009 (Carterette, Pavlu, Fang, & Kanoulas, 2009)'da yer alan Million Query Track'deki amaçlardan biri

büyük koleksiyonlar üzerinde araştırma yapmak ve diğeri ise sığ sorgu alaka yargılarında veya her sorgu için az sayıda dokümanın alaka düzeyinin belirlendiği durumda sistemlerin performansının nasıl değiştiğini incelemektir. Diğer Track'lerin aksine burada çok sayıda sorgu için sığ sayıda doküman alaka düzeyi belirlenmiştir. Million Query 2007 ve Million Query 2008 sorgu setleri GOV2 koleksiyonunu, Million Query 2009 sorgu seti ise ClueWeb09 Category-B koleksiyonunu kullanmaktadır. Tez kapsamında kullanılan sorgu sayısı Million Query 2007 için 1524, Million Query 2008 için 564 ve Million Query 2009 için 562'dir. Million Query Track'leri üzerinde yapılan bişgi erişim sistemi performans değerlendirmesi için `statAP MQ eval v4.pl` (http-7) kodu kullanılmaktadır.

5.2.4. Web Track

TREC Web Track, 2009 ile 2014 yılları arasında gerçekleşmiştir. 2009-2012 yılları arasında yapılan dört Track, ClueWeb09 doküman koleksiyonunu kullanmıştır. Sonraki iki Track için ise ClueWeb12 doküman koleksiyonu benimsenmiştir. Her bir yıl için 50 sorgu katılımcılara sunulmuştur. Toplamda 297 sorgu tez kapsamında kullanılabilmiştir. Bunlardan 197 tanesi ClueWeb09 koleksiyonları üzerinde çalışan TREC Web Track 2009 (Clarke, Craswell, & Soboroff, 2009) - TREC Web Track 2012; 100 tanesi ise ClueWeb12 koleksiyonu üzerinde çalışan TREC Web Track 2013 (Collins-Thompson, Bennett, Diaz, Clarke, & Voorhees, 2013) - TREC Web Track 2014'dür.

5.2.5. Tasks Track

TREC Task Track 2015 (Yilmaz, ve diğerleri, 2015) ve 2016, ClueWeb12 üzerinde çalışan 100 sorgu üretilmiş ancak bunlardan 85 tanesi tez kapsamında kullanılabilmiştir. Bu Track'deki amaçlardan biri sisteme verilen sorguları en kapsamlı şekilde anlayabilmektir. Örneğin "hotels in London" için olası hedefler "hotels in London [price]", "hotels in London [location]" veya "hotels [reviews] in London" olabilir. Londra'daki otelleri arayan bir kullanıcının fiyat, konum veya otel yorumları gibi hedefleri mümkün olduğu kadar kapsayan sonuçlar ister. Kısaca amaçlardan biri mümkün olduğu kadar tüm hedefleri kapsayan dokümanları kullanıcıya sunabilmektir. Diğer bir amaç ise sistemin bu kapsamda ne kadar faydalı olduğunun ölçülmesidir. Ayrıca sistemlerin klasik Ad-Hoc bilgi erişim performanslarının karşılaştırılabilmesi için de kullanılmıştır.

Tablo 5.1. *Koleksiyonlara göre Track'lerin sahip olduğu sorgu sayıları*

Koleksiyon	Track	Deneylerde temsil eden Etiket	Sorgu Sayısı
WSJ	Ad Hoc 1,2 & 3	WSJ	150
GOV2	Terabyte 2004, 2005 & 2006	GOV2	149
	Million Query 2007	MQ07	1524
	Million Query 2008	MQ08	564
CW09B	Million Query 2009	MQ09	562
	Web 2009, 2010, 2011 & 2012	CW09B	197
CW12-B13	Web 2013 & 2014 Tasks 2015 & 2016	CW12B	185
	We Want Web 1 & 2	NTCIR	180

5.2.6. We Want Web Task

We Want Web Task, kısaca WWW Task, klasik Ad-Hoc Web bilgi erişim değerlendirilmesi için NII Testbeds and Community for Information access Research (NTCIR) tarafından yürütülen projedir. İlki 2017 yılında gerçekleştirilen WWW halen devam etmektedir. WWW-1 (Luo, ve diğerleri, 2017) ve WWW-2 (Mao, ve diğerleri, 2019) de toplam 180 sorgu bulunmaktadır.

5.2.7. Özet

Bu tezde dört farklı doküman koleksiyonu, sekiz farklı sorgu seti kullanılmıştır. CW09B, CW12-B13 ve GOV2 doküman koleksiyonları Web dokümanlarından, WSJ ise metin dokümanlarından oluşmaktadır. Dolayısıyla Web dokümanlarından oluşan koleksiyonlardan doküman metninin çıkarılması ek bir işlem gerektirir. Bu işlem için `jsoup` (http-14) kütüphanesi kullanılarak HyperText Markup Language (HTML) Web dokümanlardan indeksleme aşamasından önce doküman metni çıkarma işlemi uygulanmıştır. HTML dokümanının başlık ve gövde kısmı kütüphane kullanılarak ham metni alınmıştır. İki metin bloğu birleştirilerek indekslemede kullanılabilecek doküman metni haline çevrilmiştir. Tablo 5.1. Koleksiyonlara göre Track'lerin sahip olduğu sorgu sayıları göstermektedir. Tablo bu bölümde değinilen koleksiyonlar ve sorgu setleri özetlenmiştir.

5.3. Terim Ağırlıklandırma Modelleri

Araştırma sorularımızdan *R1* için yapılan oracle analizimizde farklı terim ağırlıklandırma ailelerinden sekiz terim ağırlıklandırma modeli kullanılmıştır. Dil tabanlı (DLM), bilgi tabanlı (LGD), rastlantısallıktan (PL2, DFRee, DPH, DLH13) ve bağımsızlıktan (DFIC) farklılık modelleri ile 2-Possion dağılımına dayalı (BM25) terim

ağırlıklandırma modelleri kullanılmıştır. Tablo 5.2. Terim ağırlıklandırma modelleri ve varsayılan parametre değerleri ile olasılık dağılımları verilmiştir. Ayrıca parametrik olan modeller için deneylerde kullanılan parametre değerleri Terrier (http-15) varsayılan değerleri olarak belirlenmiştir ve tablonun son sütununda sunulmuştur.

Tablo 5.2. *Terim ağırlıklandırma modelleri ve varsayılan parametre değerleri*

No	Model	Olasılık Dağılımı	Varsayılan Parametre
1	DLM (Zhai & Lafferty, 2004)	Binomial/ multinomial	$\mu = 2500$
2	LGD (Clinchant & Gaussier, 2010; 2011)	Log-logistic regression	$c = 1$
3	PL2 (Amati & van Rijsbergen, 2002)	Possion	$c = 1$
4	BM25 (Robertson & Zaragoza, 2009)	2-Poisson	$k_1 = 1.2$, $b = 0.75$
5	DFIC (Kocabaş, Dinçer, & Karaoğlan, 2014)	Ampirik	-
6	DFRee (Amati G. , 2009)	Hipergeometrik	-
7	DPH (Amati G. , 2006)	Hipergeometrik	-
8	DLH13 (Amati G. , 2006)	Hipergeometrik	-

5.4. Bilgi Erişim Sistemi Yazılım Çatısı – Apache Lucene Framework

Bilgi erişim alanında, deneylerinin tekrarlanabilir olması ve bu konuda olabildiğince standartları takip ederek deneylerin yürütülmesi teşvik edilir (Arguello, Crane, Diaz, Lin, & Trotman, 2016; Lin, ve diğerleri, 2016; Voorhees, Rajput, & Soboroff, 2016). Bilgi alanında önde gelen konferanslarda deneylerin için benimsenen prensipler şu şekilde listelenebilir:

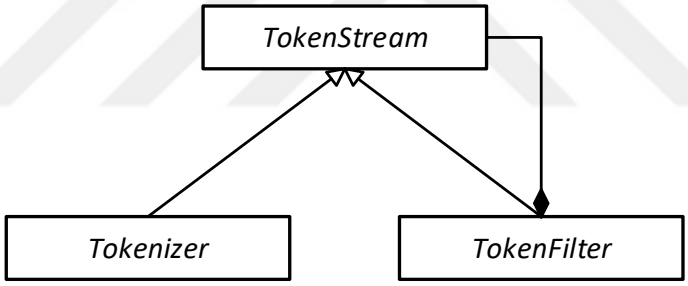
- Tekrarlanabilirlik (Repeatability): Önceki çalışmalarda yapılan deneylerin sonuçlarının aynı sistem konfigürasyonları ve veri setleri ile tekrarlanabilir olmasıdır.
- Yeniden Üretilebilirlik (Reproducibility): Önceki çalışmaların farklı veri setleri ile yeniden üretilebilir ve sonuçların önceki çalışmalar ile karşılaştırılabilir olmasıdır.
- Genelleştirilebilirlik (Generalizability): Mevcut tekniğin farklı bilgi erişim alanlarına uygulanabilir olmasıdır.

Bu amaçla, bu bölümde deneylerin tekrarlanabilirliği için gerekli olan sistem bilgilerinin detayları verilmiştir. Yürütülen deneylerin kaynak kodları GitHub deposunda (http-16) açık kaynak kod olarak servis edilmiştir (http-17).

Apache Lucen (Białecki, Muir, & Ingersoll, 2012), açık kaynak kodlu indeksleme ve bilgi erişimi yazılımıdır (http-18). Apache Lucene deneylerimizin yürütülmesi için bir yazılım çatısı olarak kullanılmıştır. Java tabanlı olan bu kütüphane yazım denetimi,

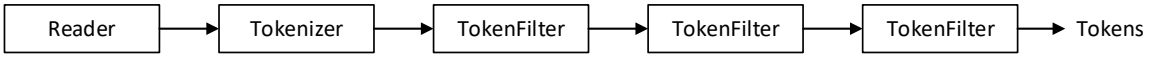
vurgulama (highlighting) ve gelişmiş metin çözümleme ve tokenizasyon özelliğine sahiptir. Apache Lucene'in asıl amacı ticari arama motorları için geliştirilmiş olsa da akademik amaçlar ile kullanımı da artan oranda yaygınlaşmaktadır (Azzopardi, ve diğerleri, 2017).

Çözümleme (Analysis), Lucene ekosisteminde en önemli işlemlerden biridir. Çözümleme ile metin indekslenebilir temel küçük parçalara (token) bölünür (McCandless, Hatcher, & Gospodnetić, 2010). Bir çözümleyici kendi içinde noktalama işlemlerinin atılması, küçük harflere çevirme, gövdeleme gibi birçok işlemi içinde barındırır. `TokenStream` bir çözümleyicinin çıktısıdır ve bir seri token içerir. `TokenStream` iki alt sınıfa sahiptir: (*i*) `Tokenizer` ve (*ii*) `TokenFilter`. `Tokenizer` sınıfı karakterleri okur ve bir seri token oluşturur. `TokenFilter` sınıfı ise oluşan token serisini alır ve üzerinde çeşitli işlemler yaparak token serisini modifiye eder. Ayrıca kendisi `TokenStream` barındırdığı için birden fazla `TokenFilter` kullanılabilir. Şekil 5.1. Token'ları üreten sınıfların hiyerarşisi verilmiştir.



Şekil 5.1. *Token'ları üreten sınıfların hiyerarşisi*

Bir `TokenStream` oluşumu `Tokenizer` ile başlar ve herhangi bir sayıda `TokenFilter` alır. Böylece bir çözümleyici zinciri oluşur. Şekil 5.2. Çözümleyici zincirinin genel gösterimi sunulmuştur.



Şekil 5.2. *Çözümleyici zinciri*

Tez kapsamın `StandardTokenizer` kullanılmıştır. Beyaz boşlukların yanı sıra e-mail adresleri gibi üst seviye türleri de algılayabilen dil tabanlı bir `Tokenizer`'dır.

Ayrıca normalizasyon için `LowerCaseFilter` zincire eklenmiştir. Sorgu zamanında gövdeleme yöntemi benimsendiği için herhangi bir gövdeleme ile ilgili `TokenFilter` uygulanmamıştır. Ancak bu özelliği sağlamak için `SynonymGraphFilterFactory` kullanılmıştır. `SynonymGraphFilterFactory` gövdelenecek kelime için morfolojik türevlerinin birleşim kümesine ihtiyacı vardır. Dolayısıyla her bir sorgu terimi için önceden oluşturulmuş birleşim kümeleri çalışma zamanında sisteme beslenmiştir.

Akademik çalışmamızı Apache Lucene ortamında sürdürebilmek için çeşitli terim ağırlıklandırma modellerin de yazılması gerekmektedir. Terrier kütüphanesi içinde farklı terim ağırlıklandırma modellerini barındırır. Tez kapsamında kullandığımız modeller Lucene versiyon 7.7.0 için adapte edilmiştir.

6. BİLGİ ERİŞİMİNDE SEÇKİLİ GÖVDELEME

Yapılan analizlerde seçkili gövdelemenin, tekil bir sistemin performansından daha yüksek performans üretebilecek potansiyele sahip olduğunu önceki bölümde belirtilmiştir. Bu çıkarımdan faydalanarak sunulan seçkili gövdeleme yaklaşımı gözetimli sınıflandırmaya dayanmaktadır. Gözetimli sınıflandırma algoritmaları etiketlenmiş veri ile bu verilere ait bir grup öznitelikler gerektirir.

6.1. Seçki Mekanizması

Tez kapsamında sunulan seçkili gövdeleme yaklaşımı verilen bir sorguya gövdeleme uygulanıp uygulanmayacağına karar veren bir modeldir. Diğer bir ifadeyle gövdeleme uygulanmamış sistem ile gövdeleme uygulamış sistem arasından, verilen bir sorgu için seçim yapan bir yaklaşımdır.

Sınıflandırma modelimiz ikili bir sınıflandırmadır ve erişim öncesi tipi öznitelikler kullanılarak makine öğrenmesi yöntemiyle üretilmiştir. Sınıf etiketleri için sorguların performans puanlarından yararlanılmıştır. Bir sorguya sınıf etiketi seçime katılan gövdelemesiz sistem ile gövdeleme uygulanmış sistemin performans puanları göz önünde bulundurularak atanır. Puanı büyük olan sistem, sorgunun etiketini belirler. Örneğin bir sorgu için NoStem nDCG@20 metriğinde 0.2 ve KStem aynı metrikte 0.17 puanını üretmiş ise, bu sorgunun etiketi NoStem olacaktır.

6.2. Kullanılan Gövdeleme Algoritmaları

Literatürde oldukça fazla gövdeleme algoritmaları sunulmuştur. Bu algoritmalar kullandıkları teknikler ve diller bakımından oldukça çeşitlidir. Tez kapsamında kullanılan doküman koleksiyonları göz önünde bulundurularak İngilizce dil yapısına uygun ve pratikte sıklıkla kullanılan kural tabanlı algoritmalar ele alınmıştır.

Kural tabanlı gövdeleme algoritmaları bilgi erişim sistemlerinde en sıklıkla kullanılan algoritmalardır. Porter (Porter, 1997), Lovins (Lovins, 1968) ve KStem (Krovetz, 1993) tez çalışmasında kullanılan algoritmalar arasındadır. Apache Lucene'nin `SnowballPorterFilterFactory` sınıfı Porter ve Lovins gövdeleme algoritmalarının uygulamasıdır ve deneylerde yapılan çalışmalarda Snowball olarak adlandırılmaktadır. Benzer bir şekilde `KStemFilter` sınıfı Krovetz gövdeleme algoritmasının Apache Lucene'deki uygulamasıdır. Her iki algoritmanın Apache Lucene varsayılan uygulaması deneylerde kullanılmıştır.

Bilgi erişim sistemlerinde gövdeleme ön-işlem aşamasında önemli bir yere sahiptir. İndeksleme aşamasında uygulanan gövdeleme, terimler sözlüğünde bulunan benzersiz terimlerin miktarını azaltmaktadır. Bu durum indeks boyutunun küçülmesine olanak vermektedir. Ancak terimlerin kayıt listeleri uzayabilmektedir. Gövdeleme ile sorgu terimlerinin dokümanlarda mutlak olarak eşleşmediği durumlar en az orana indirgenir.

Tablo 6.1. *NoStem ve gövdeleme algoritmalarının BM25 terim ağırlıklandırma modeli için DCG@20 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1689	1.79E-04	WSJ	KStem	0.3833	1.95E-05
	Lovins	0.1725	1.15E-05		Lovins	0.3642	5.36E-04
	Porter	0.1723	1.55E-05		Porter	0.3937	3.55E-06
CW12B	KStem	0.0907	5.76E-05	MQ07	KStem	0.2530	<1.00E-20
	Lovins	0.0933	3.69E-04		Lovins	0.2584	<1.00E-20
	Porter	0.0946	2.01E-04		Porter	0.2632	<1.00E-20
NTCIR	KStem	0.3167	2.07E-05	MQ08	KStem	0.2896	8.60E-17
	Lovins	0.3142	1.30E-07		Lovins	0.2939	1.44E-15
	Porter	0.3197	2.45E-06		Porter	0.3014	1.20E-19
GOV2	KStem	0.4018	3.05E-06	MQ09	KStem	0.2733	9.56E-14
	Lovins	0.4027	1.08E-06		Lovins	0.2752	3.74E-17
	Porter	0.4160	2.12E-07		Porter	0.2794	1.84E-15

Gövdeleme algoritmaları ile yapılan analiz çalışması ise NoStem ve her bir algoritma için üretilen oracle modelinin istatistiksel anlamlılık düzeyinin ölçülmesidir. Burada amaç üretilecek olan seçkili bir modelin ulaşabileceği en üst limitinin yeterli anlamlılık düzeyine sahip olup olmadığını görmektir. EK-1 ile verilen tablolarda, bu bağlamda yapılan analizler sunulmuştur. Yapılan analizler ile $p<0.1$ anlamlılık düzeyinde yapılan *t*-testi için, oracle modellerinin istatistiksel olarak anlamlı bir fark sahip olduğu görülmektedir. Örnek olarak verilen Tablo 6.1. NoStem ve gövdeleme algoritmalarının BM25 terim ağırlıklandırma modeli için DCG@20 metriğinde Oracle performansı ve *t*-test sonuçlarının *p*-değerleri verilmiştir. Tabloya göre gövdeleme algoritmaları ve NoStem'den oluşturulan ikililer ile elde edilen Oracle model, istatistiksel olarak anlamlı düzeyde fark yaratmıştır. Bu durum seçkili yaklaşım üzerine çalışma yapmanın potansiyelini ortaya koymaktadır. Ayrıca gövdeleme algoritmalarının tüm sorgu setleri ve terim ağırlıklandırma yöntemleri için MAP, nDCG@20, nDCG@100 metriklerinde geri getirim performans puanları EK-2 ile verilen tablolarda sunulmuştur. Sonuç olarak tez kapsamında Bölüm-1'de verilen araştırma sorularımızdan *R1* için cevaplar aranmış ve bu sorulara olan hipotezlerden *H1*'in sağlandığı görülmektedir:

R1 Seçkili yaklaşım bir sistemin performansını istatistiksel olarak anlamlı düzeyde artırma potansiyeline sahip midir?

H1 Seçkili gövdeleme yaklaşımının üretebileceği en yüksek performans mevcut tekil sistemin ürettiği performansa göre istatistiksel olarak anlamlı düzeyde farklıdır.

6.3. Bilgi Erişim Öncesi Tipi Öznitelikleri

Sınıflandırmada kullanılan özniteliklerin bir kısmı sorgu performans belirleyicilerden (Carmel & Yom-Tov, 2010) oluşmaktadır. Deneyimizde literatürde sunulan minimum IDF'in maksimum IDF'e oranı, sorgu kapsamı (He & Ounis, 2006), maksimum IDF, ortalama SCQ (Zhao, Scholer, & Tsegay, 2008) öznitelikleri kullanılmıştır. Ayrıca, bir sorguya gövdeleme uygulamanın etkisi, sorgu terimlerinin doküman ve terim frekansları üzerinden (terim özgüllüğü) açıklanabilir. Bu çıkarım ile deneyimizde frekanslardaki değişimi temel alan öznitelikler de öznitelik kümesine eklenmiştir:

- **AvgIncDF:** Gövdeleme, bir IR sisteminde indeksteki morfolojik türevlerin kayıt listesini birleştirerek bir terimin doküman frekansını (DF) artırır. Doküman frekansındaki artış her terim için aynı oranda olmayabilir. Bu öznitelik temel olarak belirli bir sorgu için doküman frekansındaki ortalama artışı ölçer ve şu şekilde temsil edilebilir:

$$\frac{1}{|q|}\sum_{i=1}^{|q|}\frac{DF_{i_{Stem}} - DF_{i_{NoStem}}}{DF_{i_{NoStem}}} \tag{6.1}$$

  Denklemde q sorguyu, $DF_{i_{Stem}}$ bir terimin gövdeleme uygulanmış sistemdeki doküman frekansını ve $DF_{i_{NoStem}}$ gövdeleme uygulanmamış sistemdeki doküman frekansını vermektedir.

- **MaxWeightedIncDF:** Ters doküman frekansı bir terimin bir doküman koleksiyonundaki terim özgüllüğünü ölçer (Spärck Jones, 1972; Robertson S. , 2004). Doküman frekansındaki artış ile terim özgüllüğü çarpılarak hesaplanan bir önem değerinin sorgu terimlerine gövdelemenin etkisinin bir göstergesi olacağı düşünülebilir. Her sorgu terimi için hesaplanan önem değerleri arasındaki maksimum önem değeri öznitelik olarak kullanılır. Denklem (6.1) ile belirtilen artış oranı terimlerin gövdelemesiz sistemdeki

özgüllüğü ile ağırlıklandırılmış ve bunların maksimumu öznitelik olarak kullanılmıştır:

$$max\left\{Idf_{i_{NoStem}} \times \frac{DF_{i_{Stem}} - DF_{i_{NoStem}}}{DF_{i_{NoStem}}}, \ldots, Idf_{n_{NoStem}} \times \frac{DF_{n_{Stem}} - DF_{n_{NoStem}}}{DF_{n_{NoStem}}}\right\} \quad (6.2)$$

- **CorrIctfRank:** Gövdeleme uygulandıktan sonra bir sorguda, bir terim için ters doküman frekansı (*Idf*) veya ters koleksiyon frekansı (*Ictf*) gibi bir terimin özgüllüğünü ifade eden öznitelik değerleri, diğer terimlere göre göreceli olarak değişebilir. Tezimizde bu durumu ters koleksiyon frekansı (*Ictf*) kullanarak bir özniteliğe dönüştürdük. Buna göre her iki sistem için sorgu terimleri özgüllük değerlerine göre sıralandıktan sonra oluşacak listelerdeki korelasyona bakılır. Korelasyonda kullanılan listeler ise terimlerin sorgu içindeki pozisyonlarıdır. Örneğin terimler sıralandığında her iki sistemdeki sorgu terimleri aynı sırada diziliyorsa doğal olarak korelasyon 1 olacaktır. Burada Spearman sıralama korelasyonunu kullanıyoruz ve korelasyon katsayısı değeri 0,7'den büyükse listeler arasında korelasyon olduğunu varsayıyoruz. 0.7'den büyük korelasyon görünmesi durumunda öznitelik 1; aksi durumda 0 değerini alır.
- **Mst-Lst-Change:** Gövdeleme uygulamak, bir sorgudaki özgüllüğü en az ve en fazla olan terimi değiştirebilir. Değişimdeki bu durum bir öznitelik olarak kullanılmıştır. Değişimin gözlenmesi durumunda öznitelik 1; aksi durumda 0 değerini alır.
- **Chi2-DF-TF:** Gövdeleme uygulandığında bazı terimlerin doküman frekansı ve terim koleksiyon frekansı diğer terimlere kıyasla farklılık gösterebilir. Bu etkiyi görmek ve bunu bir özniteliğe dönüştürmek için Pearson's Ki-Kare uyum iyiliği testinin *p*-değeri kullanılmıştır. İlk olarak her iki sistem için verilen sorgu terimlerinin doküman ve koleksiyon terim frekanslarından oluşan iki liste oluşturulur. Bu liste oluşturulurken terimlerin sorgudaki pozisyon sıraları korunmuştur. Ki-Kare uyum iyiliği testi kesikli değerler gerektirdiği için frekanslara Freedman-Diaconis gruplama stratejisi uygulanmıştır. Böylece sınırlı sayıda grup oluşturulmuş ve terimler gruplanmıştır. Bu testten elde edilen *p*-değeri öznitelik kümesine eklenmiştir.

- **ModifiedSCS:** Bu öznitelik literatürde olan "*simplified query clarity score*" özniteliğinin (Cronen-Townsend, Zhou, & Croft, 2002), gövdeleme uygulanması ile farklılaşan terim frekanslarına göre adapte edilmiş şeklidir (Denklem (6.3)).

$$\sum_{Q} P_{ml}(w|Q) \cdot \ln\left(\frac{P_{ml}(w|Q)}{P_{coll}(stem(w))}\right) \tag{6.3}$$

Denklemde $P_{ml}(w|Q)$, bir sorgunun literatürde mevcut olan maksimum olabilirlik metodunu ifade eder. Bu metot qtf/ql olarak verilmiştir. İlk değişken, qtf, bir sorgudaki terimin sorgu içi frekansıdır. Diğer değişken ql ise sorgunun terim uzunluğudur. Çalışmamızda $P_{coll}(stem(w))$, koleksiyon modeli olarak tanımlanmıştır. Bu yöntem literatürde $P_{coll}(w)$ olarak sunulmuştur ve $tf_{coll}/token_{coll}$ formülü ile hesaplanmaktadır. Ancak denklemimizde $P_{coll}(stem(w))$ haliyle kullanılmıştır ve tf_{coll} / $tf_{coll}(stem(w))$ hesaplanmaktadır. tf_{coll} , $token_{coll}$ ve $tf_{coll}(stem(w))$ sırası ile terimin koleksiyon frekansı, koleksiyondaki terim sayısı ve gövdelenmiş sorgu teriminin koleksiyon frekansıdır.

6.4. *K*-En Yakın Komşu Sınıflandırma Algoritması

K-En yakın komşu algoritması (*k*-NN) etkili, anlaması ve uygulaması oldukça kolay temel sınıflandırma algoritmalarından biridir. Verilen etiketsiz bir girdi, etiketli veriler ile karşılaştırılır. Benzerliği en yüksek olan top-*k* verinin etiketlerinden faydalanarak verilen etiketsiz girdinin etiketi belirlenir. Etiket, benzerliği en yüksek olan top-*k* verinin etiketlerinden oy çokluğu ile karar verilir. Bu sebepten dolayı "k" değeri genelde tek sayı olarak belirlenir. Bu algoritma veriler hakkında bir kabullenmeye sahip değildir. Sayısal ve nominal öznitelikler için uygundur (Harrington, 2012).

6.5. Seçkili Gövdeleme Yönteminin Değerlendirmesi

Seçkili gövdeleme literatürde en çok bilinen gövdeleme algoritmalarından kural tabanlı gövdeleme algoritmaları KStem (Krovetz, 1993), Lovins (Lovins, 1968) ve Porter (Porter, 1997) algoritmalarına genelleştirilmiştir. Bu algoritmalar literatürde sıklıkla kullanılan gövdeleme algoritmadır.

Web bilgi erişiminde kullanıcılar genelde alakalı dokümanları ilk bir veya iki sayfada bulmaya çalışır ve daha ileriki sayfalara göz atma eğilimde değildirler. Bu genel bir kullanıcı davranışıdır. Dolayısıyla nDCG@20 performans ölçümü bu davranışa uygun bir metriktir.

BM25 terim ağırlıklandırma metodu da literatürde ve pratikte oldukça sık kullanılan ve en çok bilinen temel bir terim ağırlıklandırma modelidir. Bu bağlamda, seçkili gövdelememiz, Web bilgi erişimine en uygun performans metriği olan nDCG@20 ve BM25 terim ağırlıklandırma modeli kombinasyonu üzerine inşa edilmiş ve test edilmiştir.

Sunulan seçkili gövdeleme modeli, üretilen öznitelikler kullanılarak *k*-NN algoritması ile gerçekleştirilmiştir. Sınıf belirlemede k değeri 11 ve bu komşuların belirlenmesinde Minkowski mesafesi (Denklem (6.4)) p=3 olarak sabitlenmiştir.

$$Q_1 = (x_{1,} x_{2,} \dots, x_n)$$

$$Q_2 = (y_{1,} y_{2,} \dots, y_n)$$

$$Minkowski(Q_1, Q_2) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (6.4)$$

Makine öğrenmesinin performansı bir çapraz doğrulama (cross validation) ile belirlenir. Deneyimiz Leave-one-out çapraz doğrulama ile sınıflandırmanın performansı ölçülmüştür. Leave-one-out çapraz doğrulama ile bir sorgu test ve geri kalanlar ise eğitim kümesi olarak ayrılır ve bu işlem tüm sorgular bir kere test verisi olana kadar tekrarlanır. Sonuçta her sorgu bir kere teste dahil edilmiş olunur. Ancak eğitim kümesinde her iki sistem için aynı performans puanına sahip sorgular bulunabilir. Bu durumda bu sorgulara sınıf atamak mümkün değildir. Test sorgusunun en yakın komşuları belirlenirken bu sorgular mesafe hesaplamasına dahil edilmemiştir; diğer bir ifade ile eğitim verilerine eklenmemiştir. Bu sayede daha temiz ve belirsizlikten uzak bir eğitim kümesi oluşturulmuş olunur.

6.5.1. Seçkili gövdelemenin bilgi erişim sistemi üzerindeki performansı değerlendirmesi

Sorgu terimlerinin bazıları sorgunun doğası gereği birden fazla terim içerebilmektedir. Ancak sorgu terimlerinin bazıları koleksiyonda bulunmayabilir. Bu

sebeple bu sorgular için özniteliğin değeri üretilemez. Bu sorgular seçkili metodunun değerlendirilmesinde yer almamıştır.

Tablo 6.2, Tablo 6.3 ve Tablo 6.4 ile seçkili gövdelemenin nDCG@20 metriğinde geri getirim performans sonuçlarını vermektedir. Tabloların ikinci sütununda sekiz sorgu seti için deneyde kullanılan sorgu sayıları verilmiştir. "Seçkili Gövdeleme" ile belirtilen sütun ise sunulan seçkili gövdeleme metodunun performans sonuçlarıdır. Her sorgu seti için en yüksek performans üreten sistem koyu renk ile işaretlenmiştir. Değerlerdeki "*B*" ve "*N*" karakterleri seçkili metodun sırasıyla temel gövdeleme algoritması (Base Stemmer) ve NoStem sistemlerine $p<0.1$ anlamlılık düzeyi için istatistiksel olarak anlamlı olduğunu ifade etmektedir.

Tablo 6.2. *KStem için seçkili gövdeleme performans sonuçları*

Sorgu Seti	Sorgu Sayısı	Geri Getirim Performans Puanları		
		NoStem	KStem	Seçkili Gövdeleme
CW09B	197	0.1606	0.1523	**0.1640**^{N,B}
CW12B	185	0.0781	0.0858	**0.0872**^{N}
NTCIR	178	0.2946	0.2956	**0.3051**^{N,B}
GOV2	149	0.3486	0.3697	**0.3707**^{N}
WSJ	148	0.3313	**0.3661**	0.3576^{N}
MQ07	1520	0.2253	0.2313	**0.2321**^{N}
MQ08	562	0.2623	0.2607	**0.2631**
MQ09	562	0.2508	0.2543	**0.2546**

Tablo 6.3. *Lovins için seçkili gövdeleme performans sonuçları*

Sorgu Seti	Sorgu Sayısı	Geri Getirim Performans Puanları		
		NoStem	Lovins	Seçkili Gövdeleme
CW09B	197	0.1606	0.1460	**0.1634**^{B}
CW12B	185	0.0781	0.0807	**0.0849**^{N}
NTCIR	178	*0.2946*	0.2681	**0.2934**^{B}
GOV2	149	0.3486	**0.3670**	0.3611
WSJ	148	0.3313	**0.3485**	0.3417^{N}
MQ07	1520	0.2253	0.2227	**0.2264**
MQ08	562	*0.2623*	0.2431	**0.2586**^{B}
MQ09	562	*0.2508*	0.2433	**0.2479**

NoStem, Lovins gövdeleme algoritması için NTCIR, MQ08 ve MQ09 koleksiyonlarında ve Porter gövdeleme algoritması için ise CW09B koleksiyonu için en yüksek puanlara sahiptir. KStem ile gerçekleştirilen seçkili yöntem, WSJ dışında kalan diğer sorgu setleri için ortalama olarak en yüksek puanları üretir. Ayrıca, CW09B, CW12B, NTCIR, GOV2, WSJ, MQ07 sorgu setlerinde NoStem'e; CW09B ve NTCIR

sorgu setleri için ise KStem'e göre ortalama performansta istatistiksel olarak anlamlı bir iyileşme sağlar. Bununla birlikte, önerilen yöntem, GOV2 ve WSJ koleksiyonları dışındaki sorgu setlerinde, temel gövdeleme algoritması (Lovins) daha yüksek performans puanları üretmiştir. Performans artışları, CW09B, NTCIR ve MQ08 için temel gövdeleme algoritmasından istatistiksel olarak anlamlı ölçüde daha yüksek iken, CW12B ve WSJ için NoStem'den anlamlı ölçüde yüksektir. Porter gövdeleme algoritması ile önerilen yöntem, CW12B, NTCIR ve GOV2 için en yüksek puanları verir. CW12B, GOV2, WSJ ve MQ07 koleksiyonları için seçkili yöntem performansı NoStem'den; NTCIR için ise temel gövdeleme algoritmasından istatistiksel olarak anlamlı ölçüde daha yüksektir.

Tablo 6.4. *Porter için seçkili gövdeleme performans sonuçları*

Sorgu Seti	Sorgu Sayısı	Geri Getirim Performans Puanları		
		NoStem	Porter	Seçkili Gövdeleme
CW09B	197	*0.1606*	0.1497	**0.1563**
CW12B	185	0.0781	0.0872	**0.0874**^{N}
NTCIR	178	0.2946	0.2839	**0.2972**^{B}
GOV2	149	0.3486	0.3839	**0.3885**^{N}
WSJ	148	0.3313	**0.3684**	0.3657^{N}
MQ07	1520	0.2253	**0.2366**	0.2315^{N}
MQ08	562	0.2623	**0.2631**	0.2609
MQ09	562	0.2508	**0.2552**	0.2549

Tablo 6.2, Tablo 6.3 ve Tablo 6.4'den elde edilen en önemli sonuç, seçkili gövdelemenin, sorgu setlerinin büyük bir kısmında tekil bir sistemin ortalama geri getirim başarımından, sistematik olarak daha yüksek puanlar üretmesidir. Diğer bir önemli sonuç, yöntemin, çeşitli sorgu setleri kullanılmış olmasına rağmen, herhangi bir tekil sistemden önemli ölçüde daha kötü bir performans puanı üretmemesidir. Diğer bir ifade ile, seçkili gövdeleme tekil sistemlerin mevcut başarımına oranla istatistiksel olarak daha düşük bir başarım sunmamasıdır. Öte yandan, seçkili gövdeleme WSJ sorgu seti için gövdeleme algoritmalarından daha yüksek bir başarım üretememiştir. Bu durumun nedeni Tablo 6.5 ve Tablo 6.6 birlikte yorumlanarak açıklanabilir. Tablo 6.5, her sorgu kümesi için sorgulardaki farklı terimlerin ortalama uzunluğunu listeler. Terimlerin ortalama uzunluklarının birbirine yakın olduğu rahatlıkla görülebilir.

Tablo 6.6 doküman koleksiyonlarındaki farklı terimlerin ortalama uzunluğunu listeler. WSJ doküman koleksiyonu ile buna karşılık gelen sorgu seti olan Ad Hoc Track'ler ortalama olarak yakın terim uzunluğuna sahiptir, ancak diğer doküman koleksiyonları, bunlara karşılık gelen sorgu setlerinden daha yüksek ortalama terim uzunluğuna sahiptir. WSJ ile diğer doküman koleksiyonları arasındaki farklardan bir diğeri, WSJ'nin bir gazete koleksiyonu olması, diğerlerinin ise Web dokümanlarını içermesidir. Ayrıca, doküman koleksiyonlarındaki farklı terimlerin sayıları da aralarındaki farklardan biridir. WSJ, diğer doküman koleksiyonlarına kıyasla oldukça az sayıda farklı terimler içerir. Bu nedenle, WSJ koleksiyonu, dokümanların kıtlığı ve farklı terimlerin sayısı göz önüne alındığında, gövdeleme algoritmaları sayesinde daha fazla doküman puanlanabilir ve alakalı dokümanları geri getirme olasılığı daha yüksektir. Böylece WSJ, gövdeleme algoritmalarının terim eşleştirme problemini hafifletmesinden daha çok yararlanmaktadır. Diğer doküman koleksiyonları çok sayıda doküman içerdiğinden, terim eşleştirme ve daha az sayıda alakalı dokümana erişim problemlerinin daha az olasılıkla ortaya çıkacağı düşünülebilir.

Tablo 6.5. *Sorguların ortalama karakter uzunlukları*

Koleksiyon	Track	Terim Uzunlukları Toplamı	Terim Sayısı	Ortalama
WSJ	Ad Hoc 1,2 & 3	3,159	447	7.07
GOV2	Terabyte 2004, 2005 & 2006	2,791	413	6.76
	Million Query 2007	18,616	2,824	6.59
	Million Query 2008	9,408	1,475	6.38
CW09B	Million Query 2009	7,113	1,140	6.24
	Web 2009, 2010, 2011 & 2012	2,457	419	5.86
CW12B	Web 2013 & 2014 Tasks 2015 & 2016	2,705	456	5.93
	We Want Web 13 & 14	2,458	396	6.21

Tablo 6.6. *Koleksiyonların ortalama karakter uzunlukları*

Koleksiyon	Terim Uzunlukları Toplamı	Terim Sayısı	Ortalama
WSJ	1,651,354	212,146	7.78
GOV2	95,309,991	10,440,851	9.13
CW09B	443,897,471	44,114,780	10.06
CW12B	556,007,008	52,065,545	10.68

Gövdeleme uygulandığında, genellikle çok doküman içeren bir koleksiyonda alakasız dokümanların sonuç listesinde yer alması beklenir ve bu durum bazı sorgularda

performansı olumsuz etkiler. Bu argüman, Tablo 6.2, Tablo 6.3 ve Tablo 6.4'de italik olarak vurgulanan durumlardan bir çıkarım olarak yapılabilir. Örneğin NoStem; CW09B sorgu seti için Porter ve NTCIR, MQ08 ve MQ09 sorgu setleri için ise Lovins gövdeleme algoritmalarından ve seçkili gövdelemeden daha yüksek puanlara sahiptir. Bu durum bilgi erişimi sistemi oluştururken gövdeleme uygulamanın ortalama başarımı artıracağı kabulünün her zaman doğru olmadığını da göstermektedir.

6.5.2. Seçkili gövdelemede gürbüzlük değerlendirmesi

Gövdeleme önceki bölümlerde de değinildiği gibi bazı sorgularda performansı artırırken bazı sorgularda ise performansı düşürmektedir. Bu durum sorgular arasında yüksek performans değişimine ve dalgalanmaya sebep olmaktadır. Örneğin bir sistemin bir sorgu seti için ortalama performansı, karşılaştırılan sisteme göre düşük olabilir. Performanstaki bu kayıp bazı sorgulardaki dramatik düşüşten ve dalgalanmadan kaynaklı olabilir. Performans kaybı istatistiksel olarak anlamlı olmasa da bu dalgalanma sistem performansının sorgular üzerinde stabil olmadığını göstermektedir. Bilgi erişiminde gürbüzlük; bir sistemin, sistemdeki herhangi bir sorgu için karşılaştırılan sistemden daha düşük performans sergileyip sergilemeyeceğini ölçmektedir. Gürbüzlük olarak adlandırılan bu ölçüm sistem performansının sorgular arasında performansın ne ölçüde dağıldığını ve sistemin sorgular için ne kadar stabil bir performans sergileyebileceğini ifade etmektedir.

Risk-sensitive olarak adlandırılan bu metriğin temelde iki test amacı vardır:

1. Ortalama bilgi erişim sisteminin performansının sorgular arasında da stabil olduğunu test eder (Collins-Thompson, 2009a; Collins-Thompson, 2009b)
2. Bilgi erişim sisteminin performansını ve gürbüzlüğünü birlikte test eder (Wang, Bennett, & Collins-Thompson, 2012).

Performansın karşılaştırılan sisteme göre düşük olması *risk,* yüksek olası ise *ödül* olarak ifade edilir. Denklem (6.5) ile verilen F_{Risk}, karşılaştırmaya temel olan sistemin (b_i) verilen sistem performansından (r_i) yüksek olduğu sorgulardaki (c) performans farklarını dikkate alır. Denklem (6.6) ile verilen F_{Reward} (ödül) ise verilen sistemin temel sistem performansından yüksek olduğu durumlardaki performans farkını dikkate alır. *Kazanç* (Denklem (6.7)) ise *ödül* ile *risk* arasındaki farktır. Denklem (6.8) ile verilen U_{Risk} sistem performansları arasındaki mutlak farkı hesaplar. Denklemdeki α parametresi riskin hassasiyet derecesidir. Değerin sıfır olması kazanç ile ödülün eşit ağırlığa sahip

olduğunu ifade eder. Değerin sıfırdan yük olması ise risk ağırlığını ödüle oranla artırdığı için sistemi cezalandıran bir kontrol parametresi olmaktadır.

$$F_{Risk} = \frac{1}{c}\sum_{i=1}^{c} max(0, b_i - r_i) \tag{6.5}$$

$$F_{Reward} = \frac{1}{c}\sum_{i=1}^{c} max(0, r_i - b_i) \tag{6.6}$$

$$U_{Gain} = F_{Reward} - F_{Risk} \tag{6.7}$$

$$U_{Risk} = U_{Gain} - \alpha \cdot F_{Risk} \tag{6.8}$$

T_{Risk} ise Student's *t*-test istatistiğine dayanan ve bilgi erişim sistemlerinin ortalama etkinliğindeki kaybın önemini test eden bir metriktir (Dinçer, Macdonald, & Ounis, 2014). T_{Risk}, U_{Risk} (Wang, Bennett, & Collins-Thompson, 2012) hesaplamasını temel almaktadır. T_{Risk} ölçümünde kullanılan kazanç Denklem (6.9) ile verilmiştir. Denklemde $SE(U_{Gain})$ standart hatayı ifade etmektedir. T_{Risk} ölçümünün U_{Risk} ölçümüne dayanan ifadesi Denklem (6.10)'da görüldüğü gibidir.

$$T_{Gain} = \frac{U_{Gain}}{SE(U_{Gain})} \tag{6.9}$$

$$T_{Risk} = \frac{U_{Risk}}{SE(U_{Risk})} \tag{6.10}$$

T_{Risk} ile hesaplanan değerlerde +2 ve -2 değerleri önemli eşik değerlerdir. İki yönlü eşleştirilmiş *t*-testi (paired *t*-test) temel alındığı göz önünde bulundurulduğunda; 95% güven aralığında istatistiksel olarak anlamlılık çıkarımı yapılabilmektedir. Örneğin, $\alpha = 0$ parametresi için -2'den küçük veya +2'den büyük değerler istatistiksel olarak anlamlıdır. $T_{Risk} > 0$ için sistemin ödülün riskine göre daha yüksek olduğu; diğer bir ifade ile riskten uzak olduğu söylenebilir. $T_{Risk} > 2$ olduğu durumlarda bir sorgunun karşılaştırılan sisteme göre daha düşük performans sergilemeyeceği sonucu çıkarılabilir. Ayrıca sorgu bazında sistem performansı, beklenen sorgu bazındaki sistem performansına göre yüksektir ve sistem risk altında değildir.

Şekil 6.1 ile seçkili gövdelemenin CW09B ve CW12B sorgu setleri için tekil sistemlere karşı gürbüzlüğü artan α değerleri için gösterilmiştir. Şekilde "SelKStem",

KStem algoritması kullanılarak sonuç alınan seçkili gövdelemeyi ifade etmektedir. Benzer şekilde "SelLovins" ve "SelPorter", sırası ile seçkili gövdelemede temel olarak kullanılan Lovins ve Porter gövdeleme algoritmalarıdır. Örneğin "NoStem vs KStem", KStem gövdeleme algoritmasının NoStem'e olan gürbüzlüğünü gösteren, 0 ile 5 arasındaki α değerleri için üretilen T_{Risk} puanının değişimidir. "NoStem vs SelKStem" ise, KStem temel gövdeleme algoritmasını kullanan seçkili gövdelemenin NoStem'e olan değişen α değerlerine göre gürbüzlük (Robustness) değişimidir. Grafiklerden de görüleceği üzere seçkili yöntem, gövdeleme algoritmasının saf bir şekilde kullanıldığı bilgi erişim sistemine göre gürbüzdür. GOV2 ve NTCIR için gürbüzlük değişim grafikleri Şekil 6.2 ile sunulmuştur. Şekil 6.3 ve Şekil 6.4 sırası ile WSJ - MQ07 ve MQ08 - MQ09 sorgu setlerinin gürbüzlük değişimlerini içermektedir. Şekil 6.1'de olduğu gibi diğer gürbüzlük grafiklerinde seçkili yöntem temel gövdeleme algoritmalarına göre gürbüzlüğü artırmaktadır.

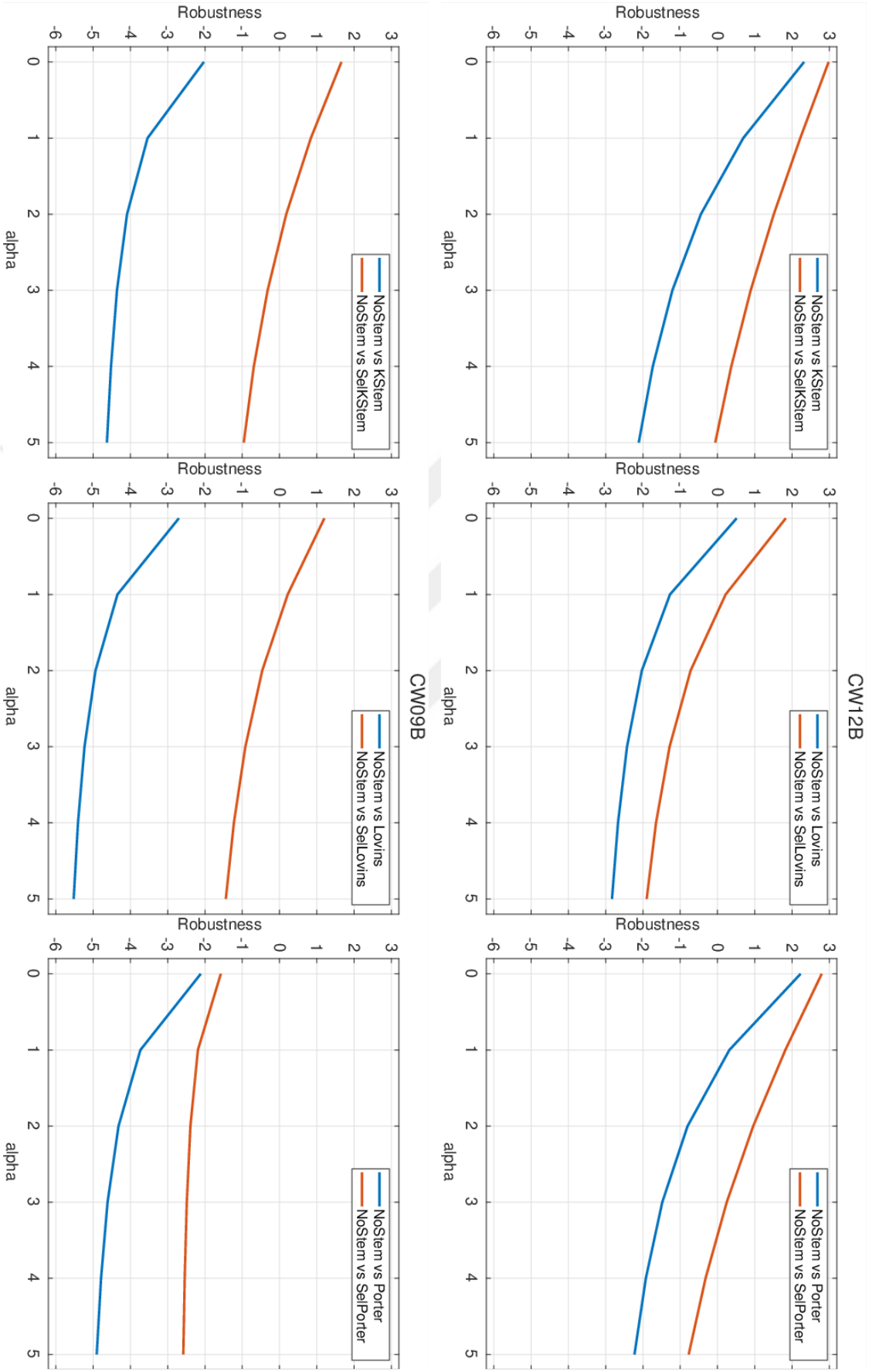
Şekil 6.5, Şekil 6.6 ve Şekil 6.7 ile gösterilen grafikler sorgu başına geri getirim performans puan farklarını vermektedir. Grafiklerin sol alt kısmında kalan sorgular NoStem'in, sağ üst kısmında kalan sorgular ise gövdeleme algoritmasının ve seçkili gövdelemenin başarılı olduğu sorgulardır. Grafiklerin *y*-eksenindeki değerleri ise gövdeleme algoritmasının ve seçkili yöntemin NoStem'e olan nDCG@20 performans farkıdır (Diff. in nDCG@20). Değerler sola yaklaştıkça küçülmektedir ve NoStem o sorgularda fark açmaktadır. Sağ tarafa yaklaştıkça değerler büyümekte ve gövdeleme veya seçkili yöntem NoStem'e o sorgularda fark açmaktadır. Seçkili yöntem, temel gövdeleme algoritmalarıyla genel olarak karşılaştırıldığında solda kalan sorgular ve sorgu başına fark azalmaktadır. Benzer şekilde sağa kalan sorgularda ise sorgu başına fark artmaktadır. Solda kalan sorguların azalması ve değerlerin küçülmesi, bilgi erişim sisteminin gövdelemeden kaynaklı, sorguların geri getirim performansındaki başarısızlık riskini azalttığının ve sistemin gürbüzlüğünü artığının göstergelerinden biridir.

Sonuç olarak tez kapsamında Bölüm-1'de verilen araştırma sorularımızdan *R2* için cevaplar aranmış ve bu sorulara olan hipotezlerden *H2*'ün sağlandığı görülmektedir:

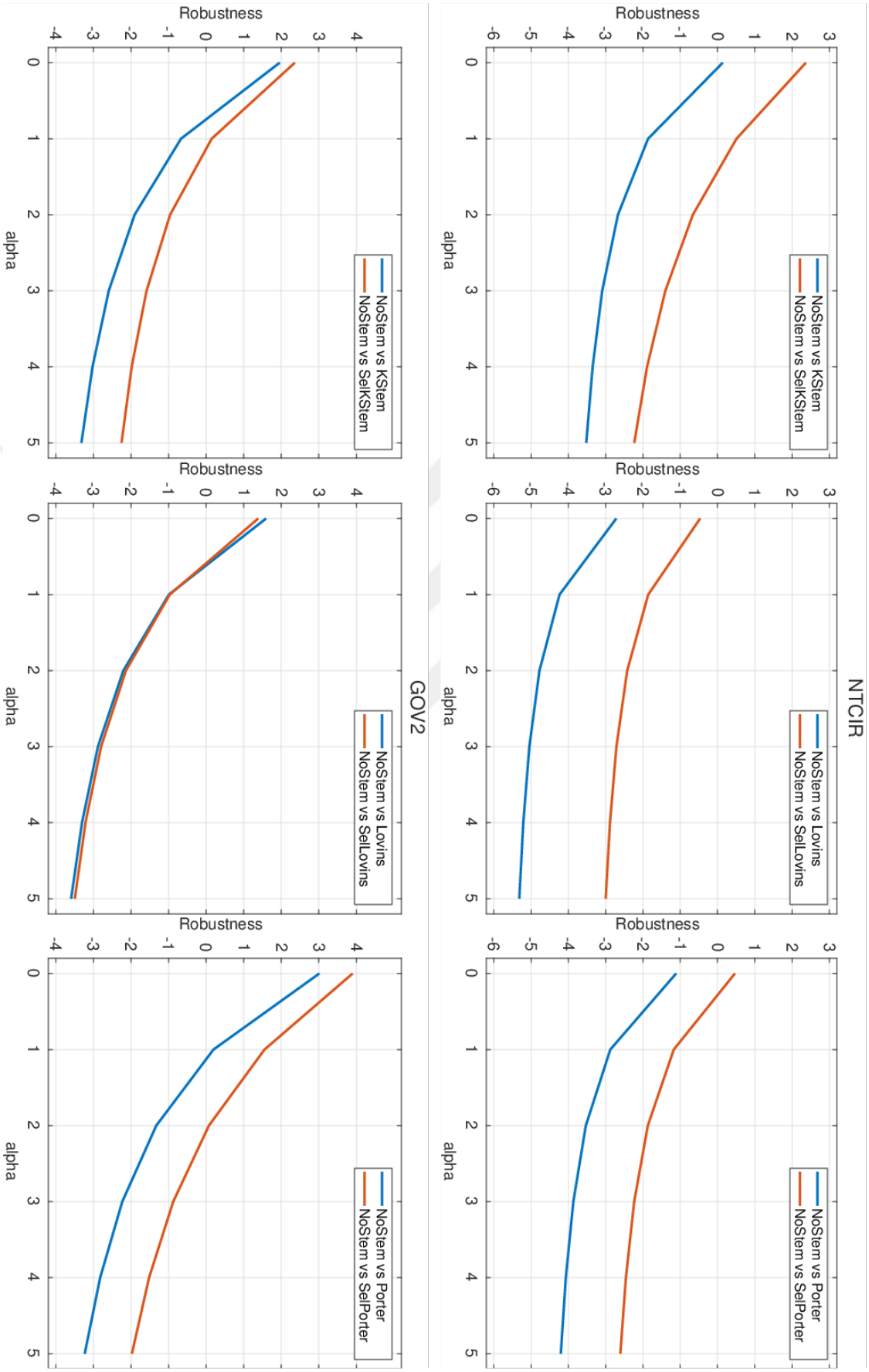
R2 Seçkili yaklaşım, gövdeleme algoritması kullanan bilgi erişim sistemlerinin gürbüzlüğüne katkıda bulunuyor mu?

H2 Verilen bir sorgu için; gövdeleme uygulanmış bir bilgi erişim sisteminin, anlamsal olarak ilgili dokümanları geri getirmedeki başarısızlık riskini en

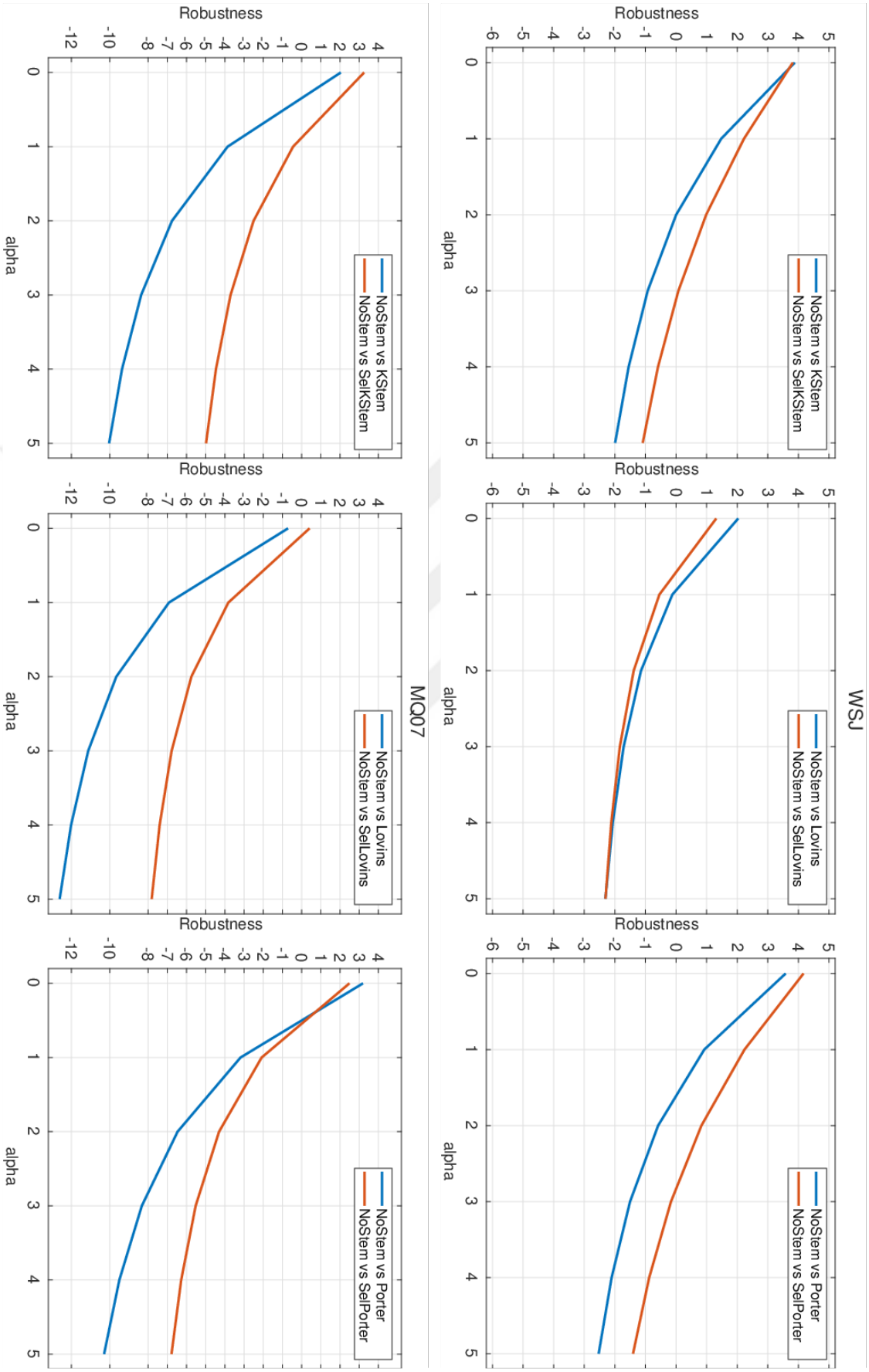
aza indiren ve gürbüzlüğü artıran bir seçkili gövdeleme modeli oluşturulabilir.



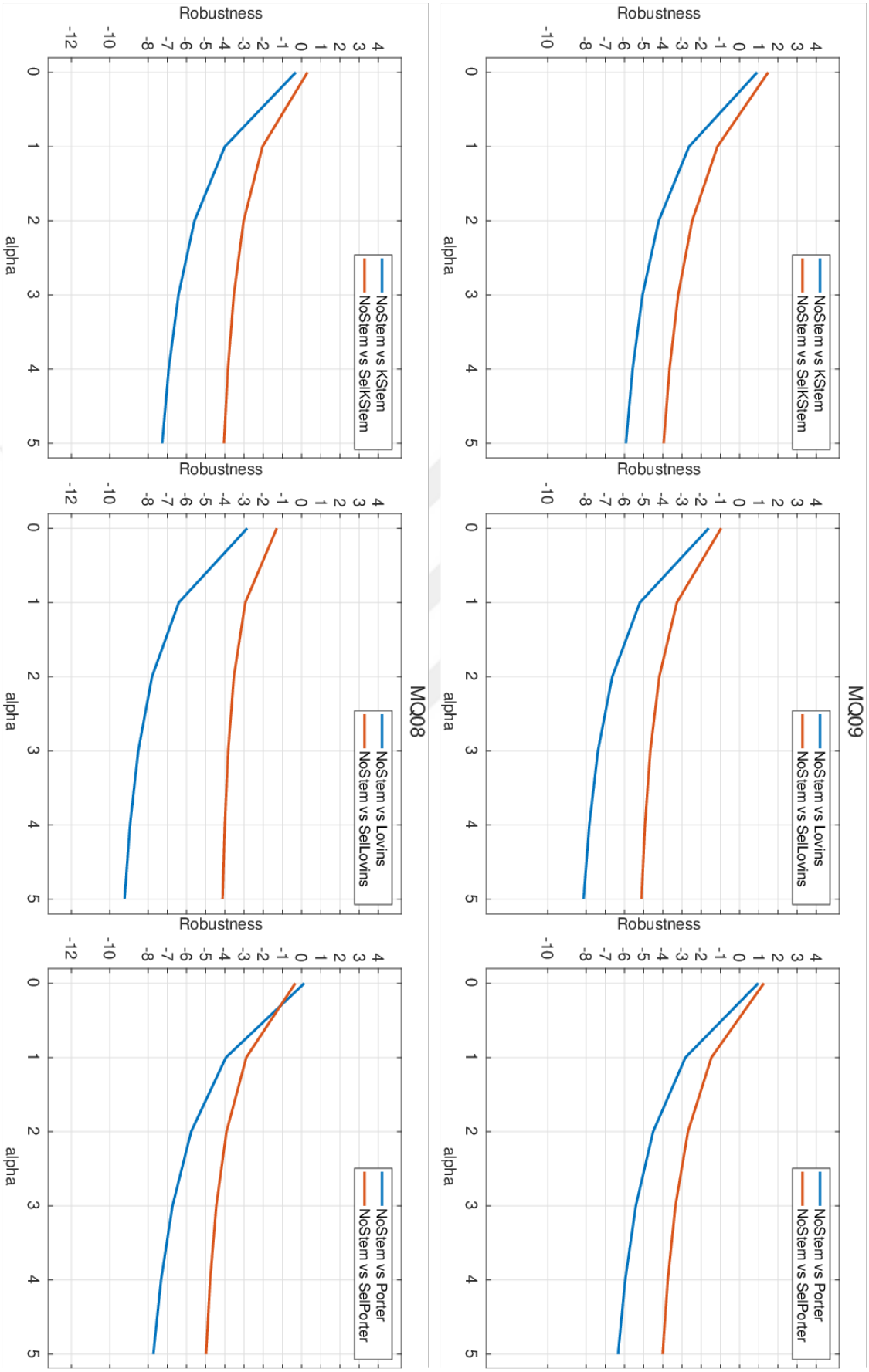
Şekil 6.1. *CW09B ve CW12B için TRisk grafikleri*



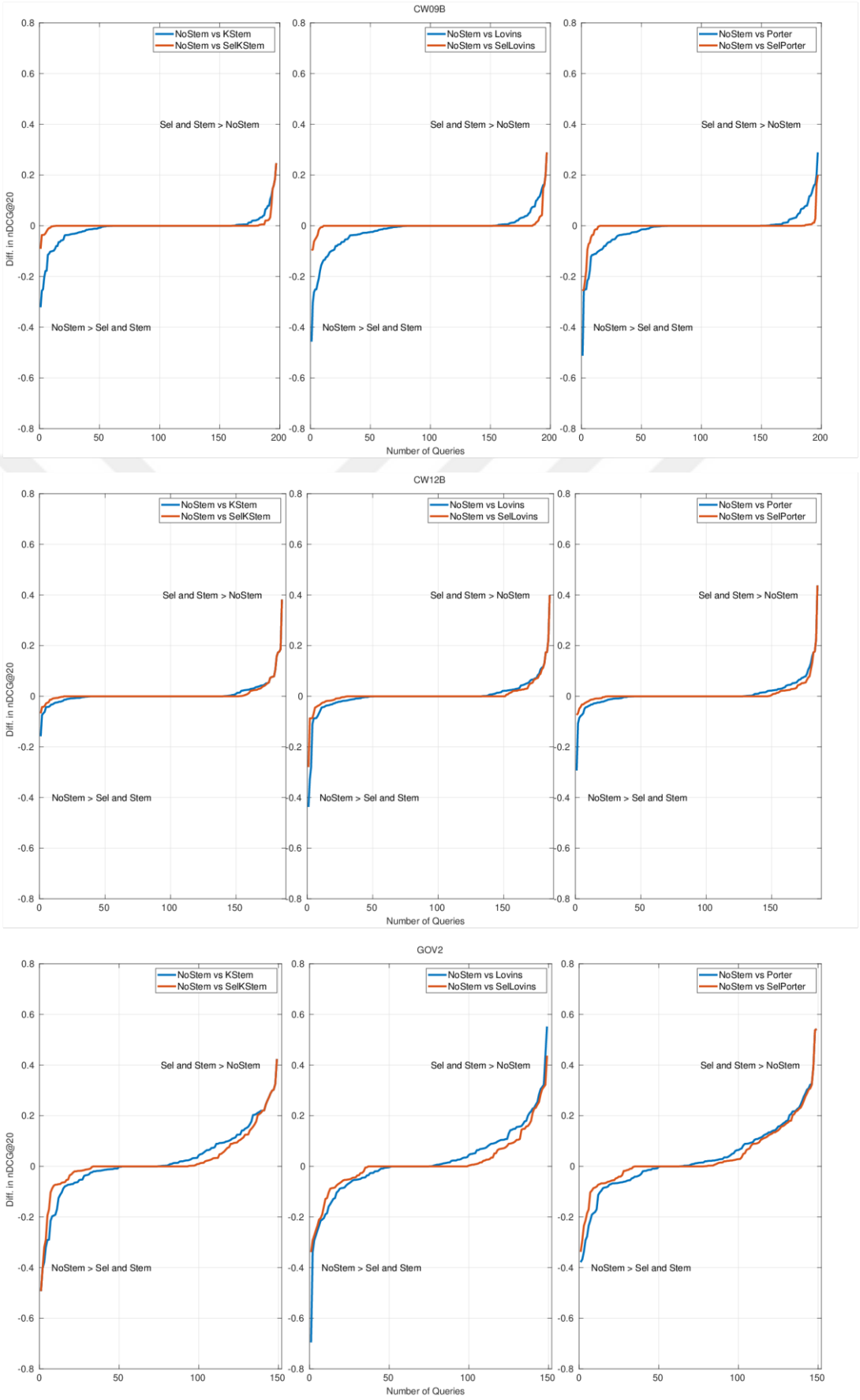
Şekil 6.2. *GOV2 ve NTCIR için TRisk grafikleri*



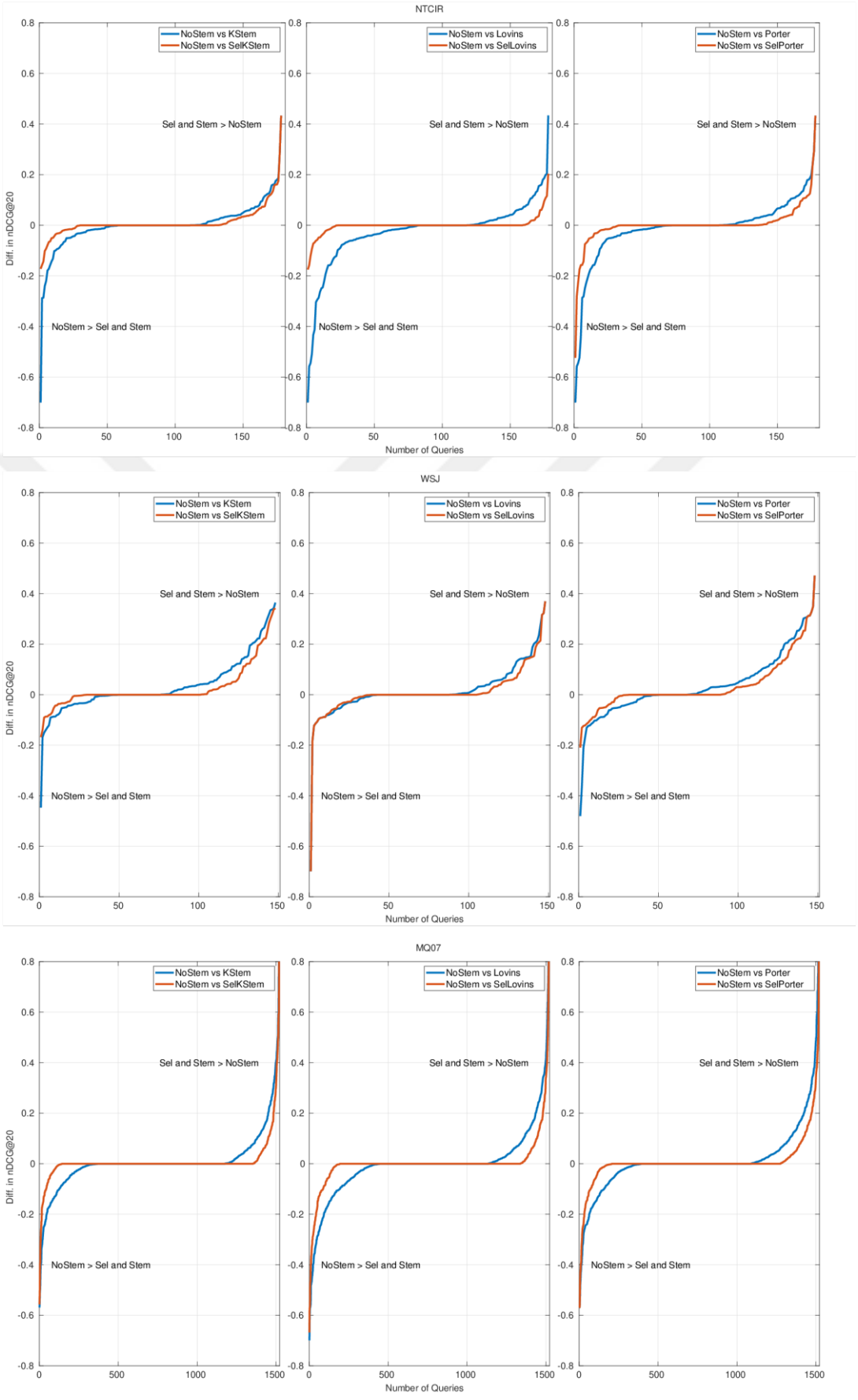
Şekil 6.3. *WSJ ve MQ07 için TRisk grafikleri*



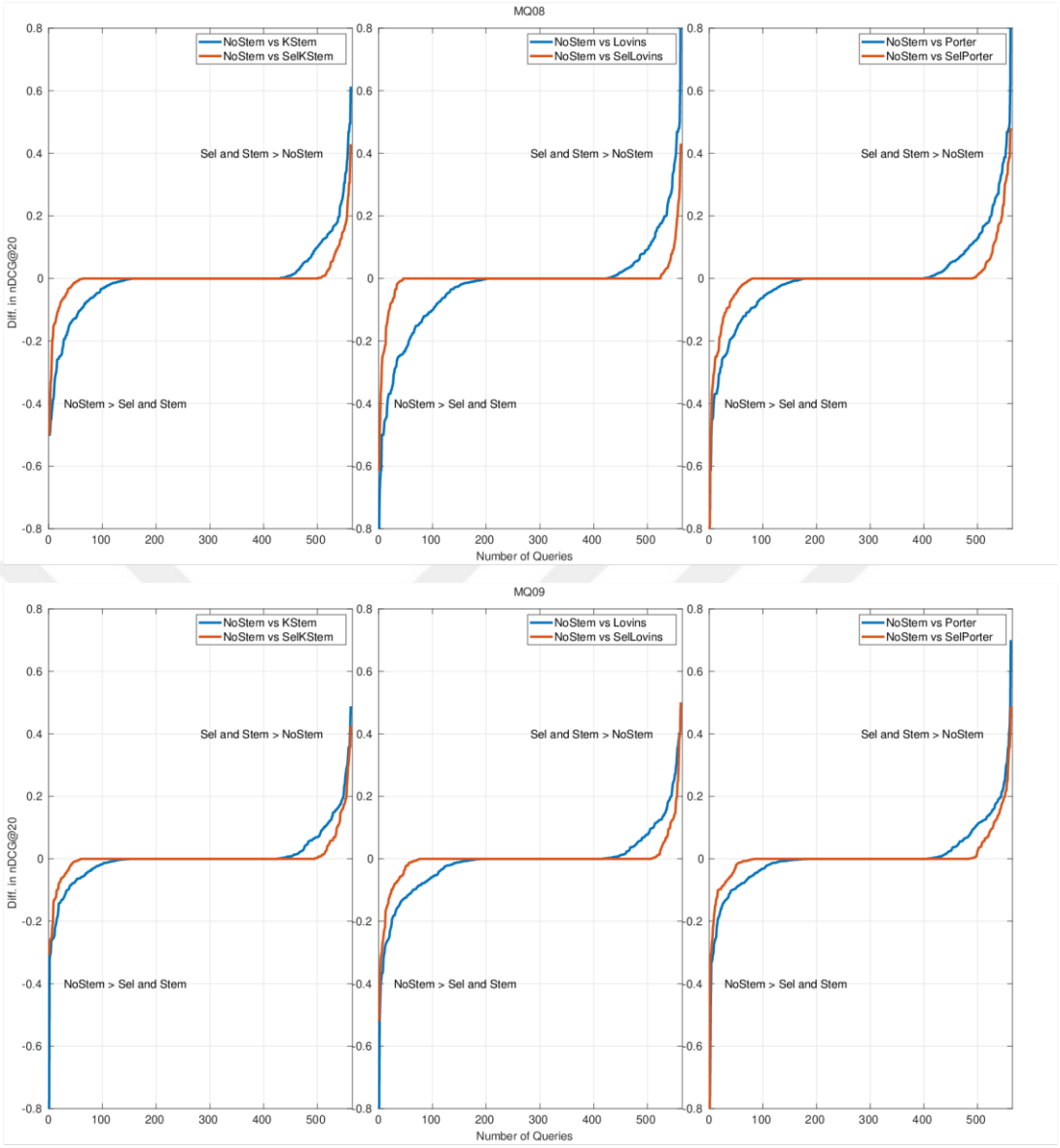
Şekil 6.4. *MQ08 ve MQ09 için TRisk grafikleri*



Şekil 6.5. *CW09B, CW12B ve GOV2 için sorgu başına performans farkları*



Şekil 6.6. *NTCIR, WSJ ve MQ07 için sorgu başına performans farkları*



Şekil 6.7. *MQ08 ve MQ09 için sorgu başına performans farkları*

6.5.3. Seçkili gövdelemede sınıflandırmanın değerlendirilmesi

Her bir sorgu seti için sınıflandırmanın doğruluğu Tablo 6.7, Tablo 6.8 ve Tablo 6.9 ile listelenmiştir. Tablolar, NoStem ve her bir gövdeleme algoritması için doğru tahmin edilen ve olması gereken (hedef) sorgu sayısını içerir. Araştırma sorularından *R3*'de, önerilen seçkili yaklaşımın, gövdelemenin uygulanıp uygulanmayacağı sorguları tahmin etmede ne derece doğru olduğunu sorgular. Sınıflandırıcı, gövdelemenin uygulanıp uygulanmayacağını tahmin etmede orta düzeyde doğruluk oranları üretir. Ancak doğruluk oranları ile, seçim yapılan sistemlerdeki sorguların geri getirim başarım puanları arasındaki farklar ihmal edilerek, sadece sınıflandırmanın performansı

hesaplanır. Çünkü, NoStem ve gövdeleme arasındaki geri getirim performansı yüksek olan sorgu ile az olan sorgular doğruluk hesaplanırken eşit ağırlığa sahiptir. Bu nedenle, ortalama bilgi erişim sistemi performans puanı ve sınıflandırmanın doğruluğu birlikte değerlendirilmesi gerekmektedir. Sonuç olarak tez kapsamında Bölüm-1'de verilen araştırma sorularımızdan *R3* için cevaplar aranmış ve bu sorulara olan hipotezlerden *H3*'ün sağlandığı görülmektedir:

R3 Seçkili yaklaşım, gövdelemenin uygulanması gereken sorguları tahmin etmede ne derecede başarılıdır?

H3 Sınıflandırma problemi olarak ele alınan seçkili yöntemin doğruluk oranı, sorgulara gövdeleme uygulanıp uygulanmayacağını tahmin etmede başarılıdır.

Tablo 6.7. *Seçkili gövdelemede KStem için sınıflandırma başarımı*

Sorgu Seti	Doğru Tahmin Edilen Sorgu Sayıları		Hedef Sorgu Sayısı		Eşit Puana Sahip Sorgu Sayıları	Doğruluk Oranı (%)
	NoStem	KStem	NoStem	KStem		
CW09B	50	17	64	37	96	66
CW12B	20	33	38	45	102	64
NTCIR	28	49	60	67	51	61
GOV2	19	56	52	75	22	59
WSJ	20	47	49	72	27	55
MQ07	223	165	366	343	811	55
MQ08	91	61	154	132	276	53
MQ09	89	67	149	136	277	55

Tablo 6.8. *Seçkili gövdelemede Lovins için sınıflandırma başarımı*

Sorgu Seti	Doğru Tahmin Edilen Sorgu Sayıları		Hedef Sorgu Sayısı		Eşit Puana Sahip Sorgu Sayıları	Doğruluk Oranı (%)
	NoStem	Lovins	NoStem	Lovins		
CW09B	72	12	83	45	69	66
CW12B	21	34	53	52	80	52
NTCIR	62	19	87	60	31	55
GOV2	15	50	51	74	24	52
WSJ	4	43	42	60	46	46
MQ07	250	182	444	385	691	52
MQ08	156	38	201	136	225	58
MQ09	113	55	189	145	228	50

Tablo 6.9. *Seçkili gövdelemede Porter için sınıflandırma başarımı*

Sorgu Seti	Doğru Tahmin Edilen Sorgu Sayıları		Hedef Sorgu Saysı		Eşit Puana Sahip Sorgu Sayıları	Doğruluk Oranı (%)
	NoStem	Porter	NoStem	Porter		
CW09B	59	13	75	48	74	59
CW12B	22	38	47	57	81	58
NTCIR	40	44	75	71	32	58
GOV2	16	71	51	86	12	64
WSJ	19	59	50	80	18	60
MQ07	185	244	392	430	698	52
MQ08	96	71	176	162	224	49
MQ09	97	78	181	155	226	52

6.5.4. Öznitelik analizi

Öznitelik analizi, her bir özniteliğin performansa ne derece katkıda bulunduğunu araştırır. Önerilen yöntem, seçkili gövdeleme için ikili sınıflandırıcılarda kullanılan çeşitli özniteliklerden yararlanmaktadır. Öznitelikler, çalışmada 24 deneyde kullanılmıştır (sekiz sorgu seti ve üç gövdeleme algoritmasının kombinasyonu), bu nedenle bu bölümde özniteliklerin bilgi erişimi etkinliği üzerindeki etkisini inceledik. Burada yapılan analiz, en çok ve en az öneme sahip öznitelikleri keşfetmeyi ve bunları seçkili gövdelemenin performans iyileştirmesindeki katkılarına göre sıralamayı amaçlamaktadır.

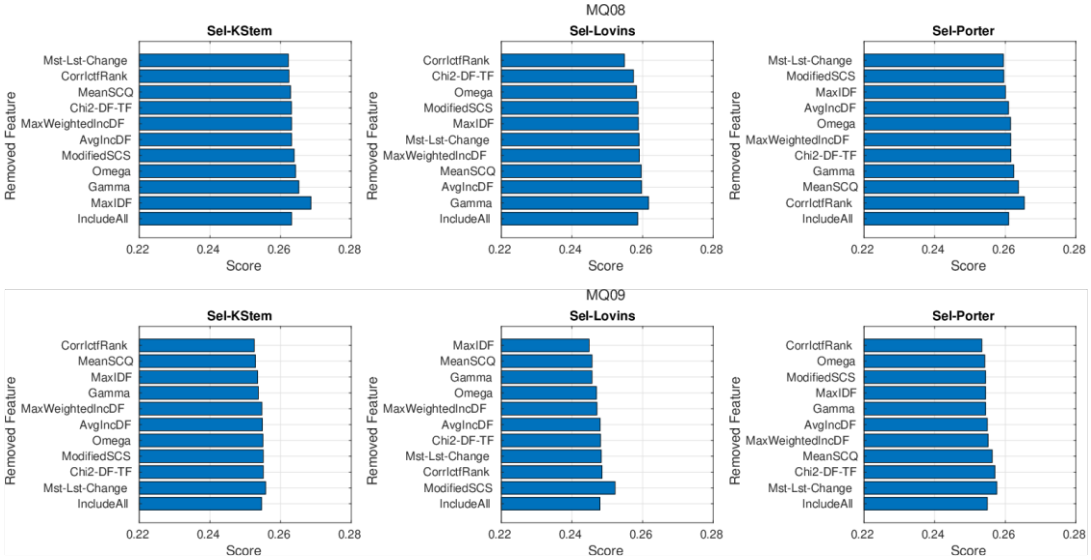
Öznitelik analizi deneylerini yürütürken bir ablasyon çalışması (ablation study) kullandık. Bu değerlendirme, öznitelikleri önem derecelerine göre sıralar. Öznitelikler, öznitelik kümesinden sırayla kaldırılır. Her yineleme için sınıflandırma modeli eğitilir ve bilgi erişim sisteminin ortalama performansı elde edilir. Son olarak, her bir indirgenmiş öznitelik seti tarafından elde edilen sistem performansları, kaldırılan özniteliğin önem derecesini belirler. Öznitelikler önem derecesine göre artan düzende sıralandığında, ilk özellik en önemli özelliktir. Çünkü bu öznitelik olmadan öznitelik seti tarafından eğitilen sınıflandırma modeli en düşük performans puanını üretmiştir.

Şekil 6.8 ve Şekil 6.9 ile sorgu setleri ve her bir gövdeleme algoritması özniteliklerin çıkarılması durumunda (Removed Feature) sistemin performansının sahip olduğu değerleri göstermektedir. Grafiklerdeki en alt satır "IncludeAll" tüm öznitelikler kullanıldığındaki sistem başarımını vermektedir. Her öznitelik, deneylere farklı bir önem derecesi ile katkı sağlar. Örneğin, MaxIDF özniteliği KStem ile CW09B için en önemli öznitelik iken, Porter için en az öneme sahip özniteliklerden biridir ve bunun kaldırılması, tüm öznitelik setini kullanan bilgi erişim sisteminden daha fazla performans üretir.

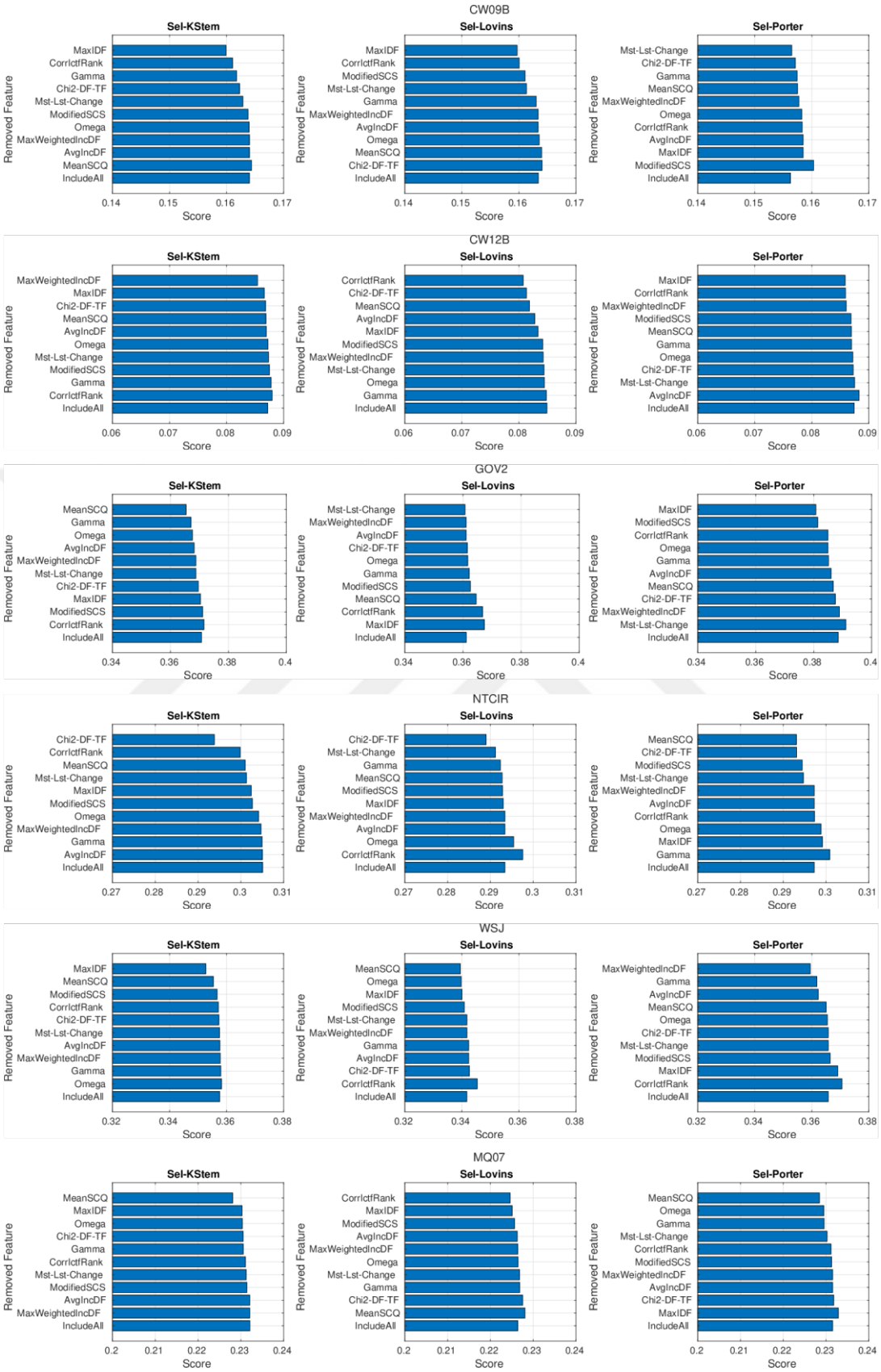
Tablo 6.10. *Özniteliklerin ortalama ve ortanca sıraları*

Öznitelik	Ortalama	Ortanca
MaxIDF	4.6	4
MeanSCQ	4.6	4
CorrIctfRank	5.2	5
Chi2-DF-TF	5.5	6
Mst-Lst-Change	5.5	6
ModifiedSCS	5.6	6
Omega	5.7	6
MaxWeightedIncDF	5.8	6
Gamma	6	6
AvgIncDF	6.5	6

Tablo 6.10 özniteliklerin önemine göre olan grafiklerin özetidir. Tablo, deneylerdeki her özniteliğin ortalama ve ortanca sıralamasını içerir. Tablodan birkaç çıkarım yapabiliriz. En önemli ve en az önemli öznitelikler açısından sırasıyla MaxIDF ve AvgIncDF aday özniteliklerdir. Genelleme yapıldığında, özniteliklerin ortalama sıraları on öznitelik göz önünde bulundurulduğunda yaklaşık olarak beşinci sıradır. Ortalamada yaklaşık ikinci sırada bir özniteliğimiz olsaydı, bunun temel öznitelik olduğu sonucuna varabilirdik. Benzer şekilde, ortalamada dokuzuncu sırada bir özniteliğimiz olsaydı, öznitelik setinden çıkarılabilir veya en az öneme sahip öznitelik olduğu sonucuna varabilirdik. Ancak, yaptığımız deneylere göre, en önemli çıkarım, özniteliklerin yaklaşık beşinci sırada oldukları için neredeyse eşit öneme sahip olmalarıdır. Tablo, özniteliklerin seçkili bilgi erişim sistemlerinin gürbüzlüğünü ve performans iyileştirmelerini sağladığını göstermektedir. Ek olarak, öznitelik setinden bazı özelliklerin kaldırılması, Şekil 6.9 ve Şekil 6.8 görüldüğü gibi geri getirim puanlarını iyileştirir. Bu öznitelikler her deney için farklıdır. Örneğin, KStem ile yürütülen deney için öznitelik kümesinden MaxIDF'nin kaldırılması performansı düşürür, ancak CW09B'de Porter ile yürütülen deney için performans artışı sağlanır. Bu tipik bir durumdur, çünkü her gövdeleme algoritması, bir doküman koleksiyonundaki terimlerin terim ve doküman frekanslarını farklı bir dereceye kadar değiştirir. Dolayısıyla terimler doküman koleksiyonu üzerinde farklı frekans dağılımlarına sahip olur. Makine öğrenimi yöntemleri verilerden öğrendiği için, terimlerin frekans dalgalanmaları her deneydeki öznitelikleri ve bunların önemini etkiler. Bu nedenle, her doküman koleksiyonu ve terimleri, uygulanan gövdeleme algoritmasına göre bireyseldir; kendine özgü frekans dağılımına sahiptir. Bu etki, geri getirim performansı için olumlu veya olumsuz olabilir. Ancak tüm öznitelikler deneylere genelleştirildiğinde, her bir özniteliğin seçkili yaklaşıma katkı sağladığı görülmektedir.



Şekil 6.8. *MQ08 ve MQ09 için öznitelik analiz grafikleri*



Şekil 6.9. *CW09B, CW12B, GOV2, NTCIR, WSJ ve MQ07 için öznitelik analiz grafikleri*

7. SONUÇ VE ÖNERİLER

Tez kapsamında gövdeleme algoritmaları üzerinde geniş ve kapsamlı analiz çalışmaları yapılmıştır. Mevcut kural tabanlı gövdeleme algoritmaları ele alınarak sorgu temelli bir seçkili gövdeleme yaklaşımının, bir bilgi erişim sisteminin performansı ne kadar artırabileceği ve gürbüzlüğü ne derece sağladığı incelenmiştir. Seçkili yaklaşım uygulanması ile elde edilebilecek performansta üretilen artış, tekil sisteme göre çoğu durumda istatistiksel olarak anlamlıdır. Bu istatistiksel çıkarım seçkili gövdelemenin potansiyelini ortaya koymaktadır.

Seçkili gövdelemenin potansiyeline dayanarak böyle bir sistem geliştirilmeye çalışılmıştır. Bu amaçla bilgi erişim sistemleri çalışmalarında standart olarak kullanılan konfigürasyonlar temel alınmıştır. BM25 terim ağırlıklandırma, kural tabanlı gövdeleme algoritmaları ve nDCG@20 metriği konfigürasyonları ile bir seçkili gövdeleme yaklaşımı sunulmuştur. Makine öğrenmesi problemi olarak ele alınan seçkili yaklaşımında, en temel makine öğrenmesi algoritmalarından olan en yakın komşuluk algoritması (*k*-NN) kullanılmıştır. Gözetimli bir sınıflandırıcı olan bu algoritma için literatürde sunulan özniteliklerden faydalanılmıştır. Bu özniteliklere ek olarak gövdelemeden doğan terim frekansları arasındaki farklılıklardan ve buna bağlı olarak terimlerin özgüllüğündeki farklılıklardan yola çıkarak üretilen öznitelikler de sınıflandırma modelinin oluşturulmasında kullanılmıştır. Deneylerde kullanılan öznitelikler, sorgu kümelerine göre geri getirim etkinliğini farklı düzeylerde iyileştirmesinde rol oynamaktadır. Bununla birlikte, ortalama olarak, öznitelikler, neredeyse eşit oranda seçkili gövdelemenin geri getirim etkinliğine katkıda bulunur. Sekiz sorgu seti üzerinde denenen seçkili gövdeleme, gövdeleme uygulanmış ve uygulanmamış tekil sistemlere göre yüksek performans üretmiştir. Üretilen performansın gürbüzlük ölçümü yapılmış ve riske duyarlı olduğu gözlemlenmiştir. Özellikle büyük ölçekli Web koleksiyonları (CW09B ve CW12B) için yöntem, sorgular arasındaki performans dalgalanmalarına karşı oldukça gürbüzdür. Ayrıca, seçkili gövdeleme, gövdelemenin tüm sorgulara sistematik olarak uygulandığı tek bir sistemden daha etkilidir. Bu, önerilen seçkili yaklaşımımızın hem gürbüzlüğü hem de performansı artırmada etkili olduğunu göstermektedir.

KAYNAKÇA

Adam, G., Asimakis, K., Bouras, C., & Poulopoulos, V. (2010). An Efficient Mechanism for Stemming and Tagging: The Case of Greek Language. *adam2010hybrid* (s. 389-397). Berlin, Heidelberg: Springer-Verlag.

Ahmed, F., & Nürnberger, A. (2009). Evaluation of n-gram conflation approaches for Arabic text retrieval. *Journal of the American Society for Information Science and Technology, 60*(7), 1448-1465.

Allan, J., Carterette, B., Aslam, J., Pavlu, V., Dachev, B., & Kanoulas, E. (2007). Million Query Track 2007 Overview. *TREC.*

Al-Shalabi, R., Kannan, G., Hilat, I., Ababneh, A., & Al-Zubi, A. (2005). Experiments with the Successor Variety Algorithm Using the Cutoff and Entropy Methods. *Information Technology Journal, 4*(1), 55-62.

Amati, G. (2006). Frequentist and Bayesian Approach to Information Retrieval. M. Lalmas, A. MacFarlane, S. Ruger, A. Tombros, T. Tsikrika, & A. Yavlinsky (Dü) içinde, *Advances in Information Retrieval* (Cilt 3936, s. 13–24). Springer Berlin / Heidelberg.

Amati, G. (2009). Divergence from Randomness Models. *Encyclopedia of Database Systems* (s. 929–932). içinde Boston, MA: Springer US.

Amati, G., & van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS), 20*, 357–389.

Amati, G., Carpineto, C., & Romano, G. (2004). Query Difficulty, Robustness, and Selective Application of Query Expansion. S. McDonald, & J. Tait (Dü.), *Advances in Information Retrieval, 26th European Conference on IR Research, ECIR 2004, Sunderland, UK, April 5-7, 2004, Proceedings.* içinde *2997*, s. 127–137. Berlin: Springer Berlin Heidelberg.

Arguello, J., Crane, M., Diaz, F., Lin, J., & Trotman, A. (2016). Report on the SIGIR 2015 Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR). *SIGIR Forum, 49*, 107–116.

Arora, S., & Yates, A. (2019). Investigating Retrieval Method Selection with Axiomatic Features. J. Beel, & L. Kotthoff (Dü.). içinde *2360*, s. 18-31. CEUR-WS.org.

Arslan, A., & Dinçer, B. T. (2019). A selective approach to index term weighting for robust information retrieval based on the frequency distributions of query terms. *Information Retrieval Journal, 22*(6), 543-569.

Aydın, A., Göksel, G., Arslan, A., & Dinçer, B. T. (2020). ESTUCeng at the NTCIR-15 WWW-3 Task: Experimenting with new document quality features. *Proceedings of the 15th NTCIR Conference on Evaluation of Information Access Technologies,* (s. 253-257).

Azzopardi, L., Crane, M., Fang, H., Ingersoll, G., Lin, J., Moshfeghi, Y., . . . Zuccon, G. (2017). The Lucene for Information Access and Retrieval Research (LIARR) Workshop at SIGIR 2017. *Proceedings of the 40th International ACM SIGIR*

Conference on Research and Development in Information Retrieval (s. 1429–1430). Shinjuku: ACM.

Bacon, F. (1605). *The Advancement of Learning.* London: Cassell & Company.

Baeza-Yates, R., & Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology behind Search* (2nd b.). USA: Addison-Wesley Publishing Company.

Balasubramanian, N., & Allan, J. (2010). Learning to Select Rankers. *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (s. 855–856). Geneva: Association for Computing Machinery.

Basu, M., Roy, A., Ghosh, K., Bandyopadhyay, S., & Ghosh, S. (2017). A Novel Word Embedding Based Stemming Approach for Microblog Retrieval During Disasters. J. M. Jose, C. Hauff, İ. Ş. Altıngövde, D. Song, D. Albakour, S. Watt, & J. Tait (Dü.), *Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings.* içinde *10193*, s. 589–597. Cham: Springer International Publishing.

Bendersky, M., Croft, W., & Diao, Y. (2011). Quality-biased ranking of web documents. *Proceedings of the fourth ACM international conference on Web search and data mining*, (s. 95-104).

Białecki, A., Muir, R., & Ingersoll, G. (2012). Apache Lucene 4. *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval*, (s. 17–24). Portland.

Bigot, A., Dejean, S., & Mothe, J. (2015). Learning to Choose the Best System Configuration in Information Retrieval: the case of repeated queries. *21*, s. 1726-1745. Graz University of Technology, Institut für Informationssysteme und Computer Medien.

Boole, G. (1854). *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities.* Walton and Maberley.

Brychcín, T., & Konopík, M. (2015). HPS: High precision stemmer. *Information Processing & Management, 51*, 68–91.

Buckley, C. (2004). Why Current IR Engines Fail., (s. 584-585). Sheffield, United Kingdom. https://doi.org/10.1145/1008992.1009132 adresinden alındı

Buckley, C. (2009). Why current IR engines fail. *Information Retrieval, 12*, 652–665.

Cao, G., Nie, J.-Y., Gao, J., & Robertson, S. (2008). Selecting Good Expansion Terms for Pseudo-Relevance Feedback. *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (s. 243–250). Singapore: Association for Computing Machinery.

Cao, G., Robertson, S., & Nie, J.-Y. (2008). Selecting query term alternations for web search by exploiting query contexts. *Proceedings of ACL-08: HLT*, (s. 148–155).

Carmel, D., & Yom-Tov, E. (2010). *Estimating the query difficulty for information retrieval.* Morgan & Claypool Publishers.

Carterette, B., Pavlu, V., Fang, H., & Kanoulas, E. (2009). Million Query Track 2009 Overview. *TREC.*

Casson, L. (2001). *Libraries in the Ancient World.* New Haven, UNITED STATES: Yale University Press.

Chapelle, O., Metlzer, D., Zhang, Y., & Grinspan, P. (2009). Expected reciprocal rank for graded relevance. *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (s. 621-630). Hong Kong, China: Association for Computing Machinery.

Chen, C., Chunyan, H., & Xiaojie, Y. (2012). Relevance Feedback Fusion via Query Expansion. *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology. 3*, s. 122-126. IEEE Computer Society.

Chin, S.-C., DeCook, R., Street, W. N., & Eichmann, D. (2010). Query-Based Text Normalization Selection Models for Enhanced Retrieval Accuracy. *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search* (s. 19–26). Los: Association for Computational Linguistics.

Chuang, M. S., & Kulkarni, A. (2017). Improving Shard Selection for Selective Search. W.-K. Sung, H. Jung, S. Xu, K. Chinnasarn, K. Sumiya, J. Lee, . . . S. Lee (Dü.), *Information Retrieval Technology - 13th Asia Information Retrieval Societies Conference, AIRS 2017, Jeju Island, South Korea, November 22-24, 2017, Proceedings* içinde (s. 29-41). Cham: Springer International Publishing.

Church, K., & Gale, W. (1999). Inverse Document Frequency (IDF): A Measure of Deviations from Poisson. S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann, & D. Yarowsky (Dü) içinde, *Natural Language Processing Using Very Large Corpora* (s. 283–295). Dordrecht: Springer Netherlands. doi:10.1007/978-94-017-2390-9_18

Clarke, C., Craswell, N., & Soboroff, I. (2004). Overview of the TREC 2004 Terabyte Track. *TREC.*

Clarke, C., Craswell, N., & Soboroff, I. (2009). Overview of the TREC 2009 Web Track. *TREC.*

Clinchant, S., & Gaussier, É. (2010). Information-based models for ad hoc IR. *Proceeding of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'10)*, (s. 234–241).

Clinchant, S., & Gaussier, É. (2011). Retrieval Constraints and Word Frequency Distributions a Log-logistic Model for IR. *Information Retrieval, 14*, 5–25.

Collins-Thompson, K. (2009a). Accounting for Stability of Retrieval Algorithms using Risk-Reward Curves. *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, (s. 27-28).

Collins-Thompson, K. (2009b). Reducing the risk of query expansion via robust constrained optimization. *Proceedings of the 18th ACM conference on Information and knowledge management*, (s. 837-846).

Collins-Thompson, K., Bennett, P., Diaz, F., Clarke, C., & Voorhees, E. (2013). TREC 2013 Web Track Overview. *TREC.*

Cormack, G., Smucker, M., & Clarke, C. (2011). Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval, 14*(5), 441-465.

Croft, B., & Xu, J. (1994). *Corpus-specific stemming using word form co-occurence.* Departement of Computer Science.

Croft, W., Metzler, D., & Strohman, T. (2010). *Search engines: Information retrieval in practice.* Addison-Wesley Reading.

Cronen-Townsend, S., Zhou, Y., & Croft, W. B. (2002). Predicting Query Performance. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (s. 299–306). Tampere: Association for Computing Machinery. doi:10.1145/564376.564429

Cronen-Townsend, S., Zhou, Y., & Croft, W. B. (2004). A Framework for Selective Query Expansion. *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management* (s. 236–237). Washington, D.C., USA: Association for Computing Machinery.

Deveaud, R., Mothe, J., Ullah, M., & Nie, J.-Y. (2018). Learning to Adaptively Rank Document Retrieval System Configurations. *ACM Trans. Inf. Syst., 37*(1).

Dinçer, B., Macdonald, C., & Ounis, I. (2014). Hypothesis testing for the risk-sensitive evaluation of retrieval systems. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, (s. 23-32).

Dominich, S. (2008). *The Modern Algebra of Information Retrieval.*

Ghanbari, E., & Shakery, A. (2019). Query-dependent learning to rank for cross-lingual information retrieval. *Knowledge and Information Systems, 59*, 711-743.

Hafer, M., & Weiss, S. (1974). Word segmentation by letter successor varieties. *Information Storage and Retrieval, 10*(11), 371-385.

Harman, D. (1987). A failure analysis of the limitation of suffixing in an online environment. *Proceedings of the 10th annual international ACM SIGIR conference on Research and development in information retrieval*, (s. 102–107).

Harman, D. (1991). How Effective Is Suffixing? *Journal of the American Society for Information Science, 42*, 7–15.

Harman, D. (2019). Information Retrieval: The Early Years. *Foundations and Trends in Information Retrieval, 13*(5), 425-577.

Harman, D., & Buckley, C. (2004). The NRRC Reliable Information Access (RIA) Workshop. *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (s. 528–529). Sheffield: ACM.

Harman, D., & Buckley, C. (2009). Overview of the Reliable Information Access Workshop. *Information Retrieval, 12*, 615–641.

Harmon, D. (1996). *Overview of the Third Text Retrieval Conference (TREC-3).*

Harrington, P. (2012). *Machine Learning in Action.* Manning Publications Co.

Hauff, C., Azzopardi, L., & Hiemstra, D. (2009). The Combination and Evaluation of Query Performance Prediction Methods. *ECIR '09 Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, (s. 301-312).

Hauff, C., Azzopardi, L., Hiemstra, D., & de Jong, F. (2010). Query Performance Prediction: Evaluation Contrasted with Effectiveness. C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, . . . K. van Rijsbergen (Dü.), *Advances in Information Retrieval - 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010.Proceedings.* içinde *5993*, s. 204–216. Berlin: Springer Berlin Heidelberg.

He, B., & Ounis, I. (2003). University of Glasgow at the Robust Track- A Query-based Model Selection. E. Voorhees, & L. Buckland (Dü.). içinde *500-255*, s. 636-645. National Institute of Standards and Technology (NIST).

He, B., & Ounis, I. (2004a). A Query-Based Pre-Retrieval Model Selection Approach to Information Retrieval. *Proceedings of RIAO 2004 - Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval* (s. 706-719). Vaucluse, France: LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

He, B., & Ounis, I. (2004b). Inferring Query Performance Using Pre-retrieval Predictors. In *String Processing and Information Retrieval* (Vol. 3246, pp. 43-54). Springer Berlin Heidelberg.

He, B., & Ounis, I. (2006). Query performance prediction. *Information Systems, 31*(7), 585-594.

Hiemstra, D. (2000). Using Language Models for Information Retrieval. *Ph.D. thesis. University of Twente, Netherlands*.

Hull, D. A. (1996). Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science, 47*, 70–84.

Järvelin, K., & Kekäläinen, J. (2002). Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst., 20*, 422–446.

Jefferson, T. (2010). *Thomas Jefferson's library: A catalog with the entries in his own order.* (J. Gilreath, & D. L. Wilson, Dü) Clark, New Jersey: The Lawbook Exchange, Ltd.

Jelinek, F. (1980). Interpolated estimation of Markov source parameters from sparse data. *Proc. Workshop on Pattern Recognition in Practice, 1980*, 381-397.

Kim, Y., Callan, J., Culpepper, J. S., & Moffat, A. (2016). Does Selective Search Benefit from WAND Optimization? N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, . . . G. Silvello (Dü.), *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings.* içinde *9626*, s. 145-158. Cham: Springer International Publishing.

Kim, Y., Callan, J., Culpepper, J. S., & Moffat, A. (2016). Load-Balancing in Distributed Selective Search. *Proceedings of the 39th International ACM SIGIR Conference*

on Research and Development in Information Retrieval (s. 905–908). Pisa: Association for Computing Machinery.

Kim, Y., Callan, J., Culpepper, J. S., & Moffat, A. (2017). Efficient distributed selective search. *Information Retrieval Journal, 20*, 221-252.

Kocabaş, İ., Dinçer, B. T., & Karaoğlan, B. (2014). A nonparametric term weighting method for information retrieval based on measuring the divergence from independence. *Information Retrieval, 17*, 153-176.

Krovetz, R. (1993). Viewing Morphology As an Inference Process. *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (s. 191–202). Pittsburgh: ACM.

Kulkarni, A., & Callan, J. (2015). Selective Search: Efficient and Effective Search of Large Textual Collections. *ACM Trans. Inf. Syst., 33*, 17:1–17:33.

Lafferty, J., & Zhai, C. (2001). Document Language Models, Query Models, and Risk Minimization for Information Retrieval. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, *51*, s. 111-119.

Lin, H.-Y., Yu, C.-H., & Chen, H.-H. (2011). Query-Dependent Rank Aggregation with Local Models - 7th Asia Information Retrieval Societies Conference, 2011, Dubai, United Arab Emirates, December 18-20, 2011. Proceedings. M. V. Salem, K. Shaalan, F. Oroumchian, A. Shakery, & H. Khelalfa (Dü.), *Information Retrieval Technology* içinde (s. 1-12). Berlin: Springer Berlin Heidelberg.

Lin, J., Crane, M., Trotman, A., Callan, J., Chattopadhyaya, I., Foley, J., . . . Vigna, S. (2016). Toward Reproducible Baselines: The Open-Source IR Reproducibility Challenge. *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings* (s. 408–420). içinde Cham: Springer International Publishing.

Liu, T. Y. (2009). *Learning to Rank for Information Retrieval* (Cilt 3). Foundations and Trends® in Information Retrieval.

Lovins, J. (1968). Development of a Stemming Algorithm. *Mech. Transl. Comput. Linguistics, 11*, 22-31.

Luo, C., Sakai, T., Liu, Y., Dou, Z., Xiong, C., & Xu, J. (2017). Overview of the NTCIR-13 We Want Web Task. *Proceedings of the 13th NTCIR Conference on Evaluation of Information Access Technologies*, 394–401.

Macdonald, C., Lioma, C., & Ounis, I. (2007). Terrier takes on the non-English Web. *Improving non-English Web searching (iNEWS07). ACM SIGIR07 Workshop*, (s. 21–28).

Macdonald, C., Santos, R., & Ounis, I. (2013). The whens and hows of learning to rank for web search. *Information Retrieval, 16*(5), 584-628.

Macdonald, C., Santos, R., Ounis, I., & He, B. (2013). About learning models with multiple query-dependent features. *ACM Transactions on Information Systems, 31*(3), 11.

Majumder, P., Mitra, M., Parui, S., Kole, G., Mitra, P., & Datta, K. (2007). YASS: Yet another suffix stripper. *ACM Transactions on Information Systems, 25*(4), 18.

Mansoub, S. K., Ercan, G., & Çiçekli, İ. (2020). Selective personalization and group profiles for improved web search Personalization. *Turkish Journal of Electrical Engineering and Computer Science, 28*, 1631–1643.

Mao, J., Sakai, T., Luo, C., Xiao, P., Liu, Y., & Dou, Z. (2019). Overview of the NTCIR-14 We Want Web Task. *Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies*, (s. 455–467).

McCandless, M., Hatcher, E., & Gospodnetić, O. (2010). *Lucene in action* (2 b.). Manning Publications Co.

McNamee, P., & Mayfield, J. (2004). Character N-Gram Tokenization for European Language Text Retrieval. *Information Retrieval, 7*(1), 73-97.

Mishra, U., & Prakash, C. (2012). MAULIK: an effective stemmer for Hindi language. *International Journal on Computer Science and Engineering, 4*(5), 711.

Mooers, C. (1950). Information Retrieval Viewed as Temporal Signaling. *Proceedings of the International Congress of Mathematicians, Vol. 1.*, s. 572–573.

Mothe, J., & Ullah, M. (2021). Defining an Optimal Configuration Set for Selective Search Strategy - A Risk-Sensitive Approach. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (s. 1335-1345). Virtual Event, Queensland, Australia: Association for Computing Machinery.

Paik, J. H., Parui, S. K., Pal, D., & Robertson, S. E. (2013). Effective and robust query-based stemming. *ACM Transactions on Information Systems (TOIS), 31*, 18.

Paik, J., & Parui, S. (2011). A Fast Corpus-Based Stemmer. *ACM Transactions on Asian Language Information Processing, 10*(2), 8.

Paik, J., Mitra, M., Parui, S., & Järvelin, K. (2011). GRAS: An effective and efficient stemming algorithm for information retrieval. *ACM Transactions on Information Systems, 29*(4), 19.

Paik, J., Pal, D., & Parui, S. (2011). A novel corpus-based stemming algorithm using co-occurrence statistics. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, (s. 863-872).

Peng, F., Ahmed, N., Li, X., & Lu, Y. (2007). Context sensitive stemming for web search. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, (s. 639–646).

Peng, J., He, B., & Ounis, I. (2009). Predicting the Usefulness of Collection Enrichment for Enterprise Search. L. Azzopardi, G. Kazai, S. Robertson, S. Rüger, M. Shokouhi, D. Song, & E. Yilmaz (Dü.), *Advances in Information Retrieval Theory - Second International Conference on the Theory of Information Retrieval, ICTIR 2009 Cambridge, UK, September 10-12, 2009 Proceedings.* içinde *5766*, s. 366-370. Berlin: Springer Berlin Heidelberg.

Peng, J., Macdonald, C., & Ounis, I. (2010). Learning to Select a Ranking Function. C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, . . . K. van Rijsbergen (Dü.), *Advances in Information Retrieval - 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010.Proceedings.* içinde *5993*, s. 114-126. Berlin: Springer Berlin Heidelberg.

Peng, J., Macdonald, C., He, B., & Ounis, I. (2009). A Study of Selective Collection Enrichment for Enterprise Search. *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (s. 1999–2002). Hong: Association for Computing Machinery.

Ponte, J., & Croft, W. (1998). A language modeling approach to information retrieval. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, *51*, s. 275-281.

Porter, M. F. (1997). An Algorithm for Suffix Stripping. *Readings in Information Retrieval* (s. 313–316). içinde San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Robertson, S. (1977). The probability ranking principle in IR. *Journal of Documentation, 33*(4), 294-304.

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation, 60*, 503–520. doi:10.1108/00220410410560582

Robertson, S., & Jones, K. (1976). Relevance weighting of search terms. *Journal of the Association for Information Science and Technology, 27*(3), 129-146.

Robertson, S., & Walker, S. (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, (s. 232-241).

Robertson, S., & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval, 3*(4), 333-389.

Roy, A., Ghorai, T., Ghosh, K., & Ghosh, S. (2017). Combining Local and Global Word Embeddings for Microblog Stemming. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (s. 2267–2270). Singapore: Association for Computing Machinery. doi:10.1145/3132847.3133103

Saleh, S., & Pecina, P. (2019). Term Selection for Query Expansion in Medical Cross-Lingual Information Retrieval. L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, & D. Hiemstra (Dü.), *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I.* içinde *11437*, s. 507-522. Cham: Springer International Publishing.

Salton, G. (1968). *Automatic information organization and retrieval.* McGraw Hill.

Salton, G., & McGill, M. (1986). *Introduction to Modern Information Retrieval.* New York, NY, USA: McGraw-Hill, Inc.

Salton, G., Wong, A., & Yang, C. (1975). A vector space model for automatic indexing. *Communications of The ACM, 18*(11), 613-620.

Siedlaczek, M., Rodriguez, J., & Suel, T. (2019). Exploiting Global Impact Ordering for Higher Throughput in Selective Search. L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, & D. Hiemstra (Dü.), *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part II.* içinde *11438*, s. 12–19. Cham: Springer International Publishing.

Singh, J., & Gupta, V. (2016). Text Stemming: Approaches, Applications, and Challenges. *ACM Computing Surveys, 49*(3), 45.

Singh, J., & Gupta, V. (2019). A novel unsupervised corpus-based stemming technique using lexicon and corpus statistics. *Knowledge-Based Systems, 180,* 147–162.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28,* 11–21. doi:10.1108/eb026526

Stein, B., & Potthast, M. (2007). Putting Successor Variety Stemming to Work. *Advances in Data Analysis*, 367-374.

Tax, N., Bockting, S., & Hiemstra, D. (2015). A cross-benchmark comparison of 87 learning to rank methods. *Information Processing and Management, 51*(6), 757-772.

Tonellotto, N., Macdonald, C., & Ounis, I. (2013). Efficient and Effective Retrieval Using Selective Pruning. *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining* (s. 63–72). Rome: Association for Computing Machinery.

Tudhope, E. (1996). *Query based stemming.* University of Waterloo, Computer Science Department.

Voorhees, E. (2001). The Philosophy of Information Retrieval Evaluation. *CLEF '01 Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems, 2406*, s. 355-370.

Voorhees, E. (2007). TREC: Continuing information retrieval's tradition of experimentation. *Communications of The ACM, 50*(11), 51-54.

Voorhees, E. (2019). The Evolution of Cranfield. *Information Retrieval Evaluation in a Changing World*, 45-69.

Voorhees, E. M., Rajput, S., & Soboroff, I. (2016). Promoting Repeatability Through Open Runs. *Proceedings of the Seventh International Workshop on Evaluating Information Access*, (s. 17–20). Tokyo.

Wang, L., Bennett, P., & Collins-Thompson, K. (2012). Robust ranking models via risk-sensitive optimization. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, (s. 761-770).

Weiss, D. (2005). *Stempelator: A hybrid stemmer for the Polish language.* Technical Report.

Wood, V. (2013). *Improving query term expansion with machine learning.* Master's thesis, University of Otago.

Xu, J., & Croft, W. B. (1998). Corpus-Based Stemming Using Cooccurrence of Word Variants. *ACM Trans. Inf. Syst., 16*(1), 61–81.

Yilmaz, E., Verma, M., Mehrotra, R., Kanoulas, E., Carterette, B., & Craswell, N. (2015). Overview of the TREC 2015 Tasks Track. *In: Voorhees, EM and Ellis, A, (eds.) National Institute of Standards and Technology (NIST) (2015).*

Yom-Tov, E., Fine, S., Carmel, D., & Darlow, A. (2005). Learning to Estimate Query Difficulty: Including Applications to Missing Content Detection and Distributed Information Retrieval. *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (s. 512–519). Salvador: Association for Computing Machinery.

Zervanou, K., Iosif, E., & Potamianos, A. (2014). Word Semantic Similarity for Morphologically Rich Languages. *LREC*, (s. 1642–1648).

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems, 22*(2), 179-214.

Zhao, Y., Scholer, F., & Tsegay, Y. (2008). Effective Pre-retrieval Query Performance Prediction Using Similarity and Variability Evidence. In *Advances in Information Retrieval* (Vol. 4956, pp. 52-64). Springer Berlin Heidelberg.

http-1: https://trec.nist.gov (Erişim tarihi: 03.10.2022)

http-2: https://www.clef-campaign.org (Erişim tarihi: 03.10.2022)

http-3: http://research.nii.ac.jp/ntcir/index-en.html (Erişim tarihi: 03.10.2022)

http-4: https://trec.nist.gov/data/qrels_eng/index.html (Erişim tarihi: 03.10.2022)

http-5: http://trec.nist.gov/trec_eval (Erişim tarihi: 05.10.2022)

http-6: http://github.com/trec-web/trec-web-2014 (Erişim tarihi: 05.10.2022)

http-7: http://ir.cis.udel.edu/million/statAP_MQ_eval_v4.pl (Erişim tarihi: 06.10.2022)

http-8: https://snowballstem.org (Erişim tarihi: 06.10.2022)

http-9: https://lemurproject.org/clueweb09 (Erişim tarihi: 06.10.2022)

http-10: https://lemurproject.org/clueweb12 (Erişim tarihi: 06.10.2022)

http-11: http://ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm (Erişim tarihi: 07.10.2022)

http-12: https://catalog.ldc.upenn.edu/LDC93T3A (Erişim tarihi: 07.10.2022)

http-13: https://trec.nist.gov/pubs/trec3/overview.ps (Erişim tarihi: 07.10.2022)

http-14: https://jsoup.org (Erişim tarihi: 10.10.2022)

http-15: http://terrier.org (Erişim tarihi: 10.10.2022)

http-16: https://github.com/gokhanc90/matlabIRexperiments (Erişim tarihi: 15.10.2022)

http-17: https://bitbucket.org/gokhanc/lucene-clueweb-retrieval/src/master (Erişim tarihi: 15.10.2022)

http-18: https://lucene.apache.org (Erişim tarihi: 15.10.2022)

EKLER

EK-1. NoStem Performansı ile Farklı Gövdeleme Algoritmaları kullanılarak hesaplanmış Oracle Performanslarının Karşılaştırılması.

Tablo E.1.1. *NoStem ve gövdeleme algoritmalarının BM25 terim ağırlıklandırma modeli için MAP metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.0937	3.09E-05	WSJ	KStem	0.2197	9.40E-05
	Lovins	0.0919	8.56E-05		Lovins	0.2066	2.64E-03
	Porter	0.0944	1.88E-05		Porter	0.2298	4.19E-05
CW12B	KStem	0.0291	3.68E-02	MQ07	KStem	0.2091	<1.00E-20
	Lovins	0.0293	9.71E-03		Lovins	0.2145	<1.00E-20
	Porter	0.0299	3.80E-02		Porter	0.2197	<1.00E-20
NTCIR	KStem	0.2549	1.39E-02	MQ08	KStem	0.296	4.53E-11
	Lovins	0.2466	1.58E-08		Lovins	0.302	8.51E-10
	Porter	0.2539	1.30E-04		Porter	0.3094	6.36E-14
GOV2	KStem	0.2507	1.50E-05	MQ09	KStem	0.2115	4.20E-13
	Lovins	0.2444	5.72E-05		Lovins	0.2154	5.78E-15
	Porter	0.2617	3.08E-05		Porter	0.2201	1.12E-14

Tablo E.1.2. *NoStem ve gövdeleme algoritmalarının BM25 terim ağırlıklandırma modeli için nDCG@100 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1775	1.01E-05	WSJ	KStem	0.3621	4.48E-03
	Lovins	0.1764	2.12E-05		Lovins	0.3459	5.48E-04
	Porter	0.1794	1.92E-06		Porter	0.3749	1.46E-03
CW12B	KStem	0.0811	3.54E-04	MQ07	KStem	0.3184	<1.00E-20
	Lovins	0.0824	9.88E-04		Lovins	0.3234	<1.00E-20
	Porter	0.0835	3.86E-04		Porter	0.3299	<1.00E-20
NTCIR	KStem	0.3335	9.75E-05	MQ08	KStem	0.3528	1.77E-18
	Lovins	0.3271	3.29E-08		Lovins	0.3564	1.19E-17
	Porter	0.3356	7.19E-06		Porter	0.3641	<1.00E-20
GOV2	KStem	0.3931	5.92E-07	MQ09	KStem	0.3461	6.41E-15
	Lovins	0.3878	7.05E-07		Lovins	0.3513	5.53E-18
	Porter	0.4066	2.88E-07		Porter	0.3543	1.52E-17

Tablo E.1.3. *NoStem ve gövdeleme algoritmalarının LGD terim ağırlıklandırma modeli için MAP metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.0947	9.03E-05	WSJ	KStem	0.2500	1.86E-05
	Lovins	0.0951	5.35E-04		Lovins	0.2361	9.04E-05
	Porter	0.0982	8.22E-06		Porter	0.2621	9.00E-07
CW12B	KStem	0.0378	1.59E-02	MQ07	KStem	0.2257	<1.00E-20
	Lovins	0.0383	4.14E-03		Lovins	0.2311	<1.00E-20
	Porter	0.0401	1.65E-02		Porter	0.2371	<1.00E-20
NTCIR	KStem	0.2854	2.20E-02	MQ08	KStem	0.3191	1.92E-09
	Lovins	0.2730	7.78E-08		Lovins	0.3296	1.42E-11
	Porter	0.2816	2.78E-04		Porter	0.3366	2.38E-12
GOV2	KStem	0.2763	1.77E-05	MQ09	KStem	0.2193	4.73E-12
	Lovins	0.2696	1.70E-05		Lovins	0.2209	2.41E-16
	Porter	0.2883	1.20E-05		Porter	0.2291	2.27E-12

Tablo E.1.4. *NoStem ve gövdeleme algoritmalarının LGD terim ağırlıklandırma modeli için nDCG@100 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1800	1.70E-06	WSJ	KStem	0.4055	4.54E-06
	Lovins	0.1789	9.77E-09		Lovins	0.3911	5.49E-06
	Porter	0.1839	7.92E-08		Porter	0.4194	2.43E-06
CW12B	KStem	0.0992	1.87E-03	MQ07	KStem	0.3371	<1.00E-20
	Lovins	0.1009	7.42E-04		Lovins	0.3423	<1.00E-20
	Porter	0.1041	1.42E-03		Porter	0.3494	<1.00E-20
NTCIR	KStem	0.3492	1.31E-05	MQ08	KStem	0.3622	1.03E-16
	Lovins	0.3411	6.10E-08		Lovins	0.3679	1.00E-20
	Porter	0.3493	1.93E-05		Porter	0.3763	1.00E-20
GOV2	KStem	0.3981	6.44E-07	MQ09	KStem	0.3435	5.00E-16
	Lovins	0.3938	1.92E-06		Lovins	0.3455	<1.00E-20
	Porter	0.4119	1.10E-05		Porter	0.3553	4.98E-16

Tablo E.1.5. *NoStem ve gövdeleme algoritmalarının LGD terim ağırlıklandırma modeli için nDCG@20 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1605	1.47E-05	WSJ	KStem	0.4422	1.96E-07
	Lovins	0.1645	1.38E-07		Lovins	0.4239	2.32E-05
	Porter	0.1670	9.59E-08		Porter	0.4531	1.14E-07
CW12B	KStem	0.1149	3.30E-05	MQ07	KStem	0.2691	<1.00E-20
	Lovins	0.1171	4.09E-05		Lovins	0.2744	<1.00E-20
	Porter	0.1202	9.08E-06		Porter	0.2806	<1.00E-20
NTCIR	KStem	0.3197	1.69E-05	MQ08	KStem	0.2974	1.33E-15
	Lovins	0.3155	2.17E-08		Lovins	0.3034	1.13E-18
	Porter	0.3213	2.42E-06		Porter	0.3138	1.00E-18
GOV2	KStem	0.3941	1.74E-07	MQ09	KStem	0.2614	8.90E-17
	Lovins	0.3950	8.06E-07		Lovins	0.2614	1.19E-18
	Porter	0.4114	3.32E-07		Porter	0.2706	1.96E-16

Tablo E.1.6. *NoStem ve gövdeleme algoritmalarının DFIC terim ağırlıklandırma modeli için MAP metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1084	8.48E-05	WSJ	KStem	0.2387	2.33E-07
	Lovins	0.1079	6.71E-06		Lovins	0.2246	9.70E-04
	Porter	0.1106	1.74E-06		Porter	0.2512	3.95E-08
CW12B	KStem	0.0384	1.83E-04	MQ07	KStem	0.1810	<1.00E-20
	Lovins	0.0374	1.20E-03		Lovins	0.1814	<1.00E-20
	Porter	0.0399	3.66E-04		Porter	0.1862	<1.00E-20
NTCIR	KStem	0.3028	1.66E-02	MQ08	KStem	0.2009	2.43E-09
	Lovins	0.2901	6.72E-09		Lovins	0.2037	1.23E-09
	Porter	0.2989	9.22E-05		Porter	0.2041	4.25E-12
GOV2	KStem	0.2648	3.78E-05	MQ09	KStem	0.2142	1.46E-14
	Lovins	0.2582	4.61E-06		Lovins	0.2149	8.31E-18
	Porter	0.2714	9.78E-05		Porter	0.2216	2.53E-17

Tablo E.1.7. *NoStem ve gövdeleme algoritmalarının DFIC terim ağırlıklandırma modeli için nDCG@100 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1971	2.29E-06	WSJ	KStem	0.3999	2.13E-05
	Lovins	0.1960	1.32E-07		Lovins	0.3807	1.37E-04
	Porter	0.1992	2.45E-08		Porter	0.4140	2.94E-05
CW12B	KStem	0.0985	1.01E-04	MQ07	KStem	0.2851	<1.00E-20
	Lovins	0.0977	8.33E-05		Lovins	0.2859	<1.00E-20
	Porter	0.1013	8.89E-05		Porter	0.2919	<1.00E-20
NTCIR	KStem	0.3423	4.96E-05	MQ08	KStem	0.2775	1.13E-15
	Lovins	0.3341	2.45E-08		Lovins	0.2772	6.09E-16
	Porter	0.3420	9.13E-06		Porter	0.2827	5.10E-19
GOV2	KStem	0.3859	2.55E-06	MQ09	KStem	0.3284	5.42E-18
	Lovins	0.3802	9.01E-08		Lovins	0.3298	1.05E-15
	Porter	0.3920	5.24E-06		Porter	0.3380	1.70E-19

Tablo E.1.8. *NoStem ve gövdeleme algoritmalarının DFIC terim ağırlıklandırma modeli için nDCG@20 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1636	5.21E-05	WSJ	KStem	0.4375	1.26E-07
	Lovins	0.1657	6.93E-06		Lovins	0.4189	9.31E-05
	Porter	0.1666	1.13E-06		Porter	0.4526	1.65E-07
CW12B	KStem	0.1128	4.97E-05	MQ07	KStem	0.2181	<1.00E-20
	Lovins	0.1116	4.96E-05		Lovins	0.2199	<1.00E-20
	Porter	0.1155	3.40E-06		Porter	0.2233	<1.00E-20
NTCIR	KStem	0.3049	2.49E-06	MQ08	KStem	0.2135	2.75E-11
	Lovins	0.2974	1.39E-06		Lovins	0.2132	4.47E-11
	Porter	0.3040	3.56E-07		Porter	0.2166	4.70E-13
GOV2	KStem	0.3965	3.84E-07	MQ09	KStem	0.2341	1.81E-14
	Lovins	0.3967	4.34E-09		Lovins	0.2359	2.57E-14
	Porter	0.4037	2.14E-07		Porter	0.2421	7.86E-17

Tablo E.1.9. *NoStem ve gövdeleme algoritmalarının DFRee terim ağırlıklandırma modeli için MAP metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.0981	2.28E-05	WSJ	KStem	0.2492	2.86E-05
	Lovins	0.0966	1.31E-05		Lovins	0.2358	8.24E-04
	Porter	0.0986	8.89E-07		Porter	0.2631	5.15E-06
CW12B	KStem	0.0324	3.60E-04	MQ07	KStem	0.2303	<1.00E-20
	Lovins	0.0325	8.20E-03		Lovins	0.2343	<1.00E-20
	Porter	0.0332	2.86E-03		Porter	0.2402	<1.00E-20
NTCIR	KStem	0.2695	1.43E-02	MQ08	KStem	0.3173	2.56E-09
	Lovins	0.2590	7.12E-08		Lovins	0.3241	9.12E-09
	Porter	0.2661	4.64E-05		Porter	0.3292	2.80E-11
GOV2	KStem	0.2766	2.61E-05	MQ09	KStem	0.2422	4.67E-11
	Lovins	0.2702	4.84E-06		Lovins	0.2421	2.04E-16
	Porter	0.2868	2.36E-05		Porter	0.2507	2.51E-14

Tablo E.1.10. *NoStem ve gövdeleme algoritmalarının DFRee terim ağırlıklandırma modeli için nDCG@100 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1812	8.60E-07	WSJ	KStem	0.4072	1.49E-04
	Lovins	0.1802	3.25E-07		Lovins	0.3905	1.30E-04
	Porter	0.1840	1.27E-07		Porter	0.4226	7.08E-05
CW12B	KStem	0.0913	6.19E-04	MQ07	KStem	0.3391	<1.00E-20
	Lovins	0.0914	1.42E-03		Lovins	0.3427	<1.00E-20
	Porter	0.0930	4.50E-04		Porter	0.3492	<1.00E-20
NTCIR	KStem	0.3445	5.17E-06	MQ08	KStem	0.3661	2.65E-14
	Lovins	0.3382	9.78E-09		Lovins	0.3683	1.74E-16
	Porter	0.3451	1.41E-06		Porter	0.3756	2.10E-18
GOV2	KStem	0.3989	7.44E-08	MQ09	KStem	0.3702	3.85E-16
	Lovins	0.3953	1.95E-08		Lovins	0.3739	3.00E-20
	Porter	0.4119	7.31E-08		Porter	0.3810	1.90E-19

Tablo E.1.11. *NoStem ve gövdeleme algoritmalarının DFRee terim ağırlıklandırma modeli için nDCG@20 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1726	6.55E-06	WSJ	KStem	0.4406	6.20E-08
	Lovins	0.1734	3.77E-07		Lovins	0.4272	3.17E-05
	Porter	0.1748	3.22E-07		Porter	0.4549	4.07E-07
CW12B	KStem	0.1007	2.06E-05	MQ07	KStem	0.2684	<1.00E-20
	Lovins	0.1019	8.69E-05		Lovins	0.2724	<1.00E-20
	Porter	0.1028	6.21E-07		Porter	0.2778	<1.00E-20
NTCIR	KStem	0.3168	3.48E-06	MQ08	KStem	0.3008	3.47E-13
	Lovins	0.3120	1.75E-08		Lovins	0.3029	3.82E-15
	Porter	0.3185	1.50E-07		Porter	0.3122	2.35E-17
GOV2	KStem	0.3928	5.43E-07	MQ09	KStem	0.2931	7.67E-17
	Lovins	0.3932	1.22E-07		Lovins	0.2926	1.10E-18
	Porter	0.4074	6.88E-08		Porter	0.2997	9.00E-20

Tablo E.1.12. *NoStem ve gövdeleme algoritmalarının DLH13 terim ağırlıklandırma modeli için MAP metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.0788	5.27E-05	WSJ	KStem	0.2429	5.53E-06
	Lovins	0.0775	4.06E-05		Lovins	0.2288	7.47E-04
	Porter	0.0799	2.18E-06		Porter	0.2576	3.78E-06
CW12B	KStem	0.0251	2.07E-04	MQ07	KStem	0.2080	<1.00E-20
	Lovins	0.0253	9.54E-03		Lovins	0.2099	<1.00E-20
	Porter	0.0259	3.88E-04		Porter	0.2139	<1.00E-20
NTCIR	KStem	0.2444	3.56E-03	MQ08	KStem	0.2700	7.69E-11
	Lovins	0.2336	1.85E-08		Lovins	0.2694	2.46E-10
	Porter	0.2393	1.34E-05		Porter	0.2767	2.86E-12
GOV2	KStem	0.2444	3.27E-05	MQ09	KStem	0.2144	2.55E-11
	Lovins	0.2374	5.33E-06		Lovins	0.2151	9.54E-15
	Porter	0.2518	2.51E-05		Porter	0.2206	5.57E-14

Tablo E.1.13. *NoStem ve gövdeleme algoritmalarının DLH13 terim ağırlıklandırma modeli için nDCG@100 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1578	8.40E-06	WSJ	KStem	0.4000	3.31E-04
	Lovins	0.1581	4.42E-06		Lovins	0.3830	5.03E-05
	Porter	0.1605	2.67E-07		Porter	0.4153	1.33E-04
CW12B	KStem	0.0779	1.41E-05	MQ07	KStem	0.3130	<1.00E-20
	Lovins	0.0779	1.29E-04		Lovins	0.3134	<1.00E-20
	Porter	0.0792	2.57E-07		Porter	0.3196	<1.00E-20
NTCIR	KStem	0.3141	1.36E-06	MQ08	KStem	0.3372	4.13E-16
	Lovins	0.3080	7.66E-08		Lovins	0.3371	2.21E-18
	Porter	0.3130	2.32E-06		Porter	0.3442	4.00E-20
GOV2	KStem	0.3610	1.23E-06	MQ09	KStem	0.3352	6.72E-16
	Lovins	0.3570	2.69E-08		Lovins	0.3380	2.71E-18
	Porter	0.3727	4.73E-07		Porter	0.3433	2.00E-20

Tablo E.1.14. *NoStem ve gövdeleme algoritmalarının DLH13 terim ağırlıklandırma modeli için nDCG@20 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1405	1.34E-05	WSJ	KStem	0.4312	3.03E-07
	Lovins	0.1432	1.24E-05		Lovins	0.4139	2.23E-05
	Porter	0.1455	4.80E-07		Porter	0.4476	1.25E-06
CW12B	KStem	0.0824	1.22E-04	MQ07	KStem	0.2464	<1.00E-20
	Lovins	0.0831	7.73E-06		Lovins	0.2469	<1.00E-20
	Porter	0.0837	1.64E-06		Porter	0.2521	<1.00E-20
NTCIR	KStem	0.2849	7.78E-07	MQ08	KStem	0.2696	3.28E-14
	Lovins	0.2821	6.32E-06		Lovins	0.2677	3.75E-16
	Porter	0.2875	3.10E-07		Porter	0.2787	1.19E-18
GOV2	KStem	0.3476	1.75E-06	MQ09	KStem	0.2550	3.00E-15
	Lovins	0.3468	1.13E-06		Lovins	0.2564	1.14E-16
	Porter	0.3615	1.28E-06		Porter	0.2610	2.10E-19

Tablo E.1.15. *NoStem ve gövdeleme algoritmalarının DLM terim ağırlıklandırma modeli için MAP metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1050	9.70E-06	WSJ	KStem	0.2645	1.81E-04
	Lovins	0.1041	1.35E-05		Lovins	0.2487	2.81E-04
	Porter	0.1062	7.16E-08		Porter	0.2802	9.59E-06
CW12B	KStem	0.0487	8.38E-05	MQ07	KStem	0.2278	<1.00E-20
	Lovins	0.0490	1.05E-02		Lovins	0.2285	<1.00E-20
	Porter	0.0518	2.62E-05		Porter	0.2357	<1.00E-20
NTCIR	KStem	0.3234	1.23E-02	MQ08	KStem	0.2824	7.81E-10
	Lovins	0.3088	4.27E-09		Lovins	0.2915	6.10E-14
	Porter	0.3181	4.33E-05		Porter	0.2986	1.23E-13
GOV2	KStem	0.2940	9.91E-05	MQ09	KStem	0.2146	3.95E-14
	Lovins	0.2892	1.24E-05		Lovins	0.2144	4.04E-16
	Porter	0.3034	1.95E-04		Porter	0.2213	2.98E-16

Tablo E.1.16. *NoStem ve gövdeleme algoritmalarının DLM terim ağırlıklandırma modeli için nDCG@100 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1905	1.51E-06	WSJ	KStem	0.4237	3.02E-05
	Lovins	0.1898	5.57E-06		Lovins	0.4036	1.34E-05
	Porter	0.1922	1.36E-07		Porter	0.4389	5.77E-06
CW12B	KStem	0.1174	1.25E-04	MQ07	KStem	0.3382	<1.00E-20
	Lovins	0.1190	3.13E-03		Lovins	0.3407	<1.00E-20
	Porter	0.1222	2.40E-05		Porter	0.3471	<1.00E-20
NTCIR	KStem	0.3542	3.52E-04	MQ08	KStem	0.3429	3.25E-16
	Lovins	0.3444	4.41E-08		Lovins	0.3465	1.13E-16
	Porter	0.3530	6.62E-06		Porter	0.3544	8.00E-20
GOV2	KStem	0.4195	9.05E-06	MQ09	KStem	0.3194	2.15E-16
	Lovins	0.4160	9.92E-08		Lovins	0.3221	2.58E-14
	Porter	0.4274	5.81E-06		Porter	0.3284	4.20E-19

Tablo E.1.17. *NoStem ve gövdeleme algoritmalarının DLM terim ağırlıklandırma modeli için nDCG@20 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1569	1.25E-05	WSJ	KStem	0.4642	3.43E-06
	Lovins	0.1579	5.43E-07		Lovins	0.4447	1.14E-04
	Porter	0.1592	4.86E-07		Porter	0.4779	1.33E-06
CW12B	KStem	0.1365	1.08E-04	MQ07	KStem	0.2709	<1.00E-20
	Lovins	0.1383	1.03E-04		Lovins	0.2727	<1.00E-20
	Porter	0.1418	8.22E-07		Porter	0.2796	<1.00E-20
NTCIR	KStem	0.3127	6.71E-06	MQ08	KStem	0.2800	6.60E-14
	Lovins	0.3071	2.67E-05		Lovins	0.2842	2.01E-15
	Porter	0.3148	5.12E-07		Porter	0.2913	2.50E-16
GOV2	KStem	0.4280	1.19E-06	MQ09	KStem	0.2299	6.72E-15
	Lovins	0.4242	6.12E-09		Lovins	0.2330	1.79E-13
	Porter	0.4348	1.68E-07		Porter	0.2389	1.86E-17

Tablo E.1.18. *NoStem ve gövdeleme algoritmalarının DPH terim ağırlıklandırma modeli için MAP metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1081	8.46E-06	WSJ	KStem	0.2561	2.83E-05
	Lovins	0.1068	1.73E-06		Lovins	0.2396	3.00E-03
	Porter	0.1095	2.38E-06		Porter	0.2693	1.44E-06
CW12B	KStem	0.0366	1.34E-04	MQ07	KStem	0.2405	<1.00E-20
	Lovins	0.0367	3.88E-03		Lovins	0.2444	<1.00E-20
	Porter	0.0374	8.86E-04		Porter	0.2495	<1.00E-20
NTCIR	KStem	0.2792	2.82E-03	MQ08	KStem	0.3203	1.19E-11
	Lovins	0.2685	4.59E-07		Lovins	0.3243	7.82E-13
	Porter	0.2756	4.26E-06		Porter	0.3307	3.25E-16
GOV2	KStem	0.2884	1.12E-04	MQ09	KStem	0.2541	1.13E-10
	Lovins	0.2798	4.35E-06		Lovins	0.2557	5.02E-14
	Porter	0.2963	6.99E-05		Porter	0.2637	1.94E-14

Tablo E.1.19. *NoStem ve gövdeleme algoritmalarının DPH terim ağırlıklandırma modeli için nDCG@100 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1985	1.41E-06	WSJ	KStem	0.4184	1.54E-05
	Lovins	0.1973	1.89E-07		Lovins	0.3989	1.52E-04
	Porter	0.2020	1.11E-07		Porter	0.4318	1.28E-04
CW12B	KStem	0.0997	1.10E-04	MQ07	KStem	0.3523	<1.00E-20
	Lovins	0.1001	3.24E-04		Lovins	0.3562	<1.00E-20
	Porter	0.1015	1.56E-04		Porter	0.3623	<1.00E-20
NTCIR	KStem	0.3489	3.42E-05	MQ08	KStem	0.3697	3.30E-19
	Lovins	0.3394	4.32E-07		Lovins	0.3754	1.00E-20
	Porter	0.3481	7.85E-07		Porter	0.3800	<1.00E-20
GOV2	KStem	0.4219	1.58E-05	MQ09	KStem	0.3861	3.78E-16
	Lovins	0.4148	1.93E-07		Lovins	0.3903	1.70E-19
	Porter	0.4330	1.42E-06		Porter	0.3976	1.77E-17

Tablo E.1.20. *NoStem ve gövdeleme algoritmalarının DPH terim ağırlıklandırma modeli için nDCG@20 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1894	1.12E-06	WSJ	KStem	0.4555	6.59E-08
	Lovins	0.1920	8.19E-08		Lovins	0.4363	5.60E-04
	Porter	0.1946	1.96E-07		Porter	0.4669	8.54E-07
CW12B	KStem	0.1127	1.27E-05	MQ07	KStem	0.2829	<1.00E-20
	Lovins	0.1131	1.13E-06		Lovins	0.2870	<1.00E-20
	Porter	0.1145	2.09E-07		Porter	0.2917	<1.00E-20
NTCIR	KStem	0.3166	1.03E-06	MQ08	KStem	0.3020	2.39E-17
	Lovins	0.3090	4.52E-07		Lovins	0.3069	1.99E-17
	Porter	0.3195	1.36E-07		Porter	0.3127	3.00E-20
GOV2	KStem	0.4311	3.32E-06	MQ09	KStem	0.3042	4.16E-15
	Lovins	0.4238	9.21E-07		Lovins	0.3060	3.30E-19
	Porter	0.4398	1.57E-06		Porter	0.3134	1.47E-16

Tablo E.1.21. *NoStem ve gövdeleme algoritmalarının PL2 terim ağırlıklandırma modeli için MAP metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.0878	6.60E-05	WSJ	KStem	0.2070	3.67E-04
	Lovins	0.0880	2.69E-03		Lovins	0.1937	1.24E-02
	Porter	0.0895	1.69E-06		Porter	0.2194	5.54E-04
CW12B	KStem	0.0350	7.30E-03	MQ07	KStem	0.1783	<1.00E-20
	Lovins	0.0345	9.06E-03		Lovins	0.1813	<1.00E-20
	Porter	0.0362	3.14E-03		Porter	0.1856	<1.00E-20
NTCIR	KStem	0.2680	1.87E-02	MQ08	KStem	0.1992	1.63E-08
	Lovins	0.2563	1.80E-08		Lovins	0.2034	2.42E-10
	Porter	0.2646	4.45E-04		Porter	0.2055	8.49E-10
GOV2	KStem	0.2205	6.68E-05	MQ09	KStem	0.2036	5.00E-10
	Lovins	0.2164	1.91E-04		Lovins	0.2055	3.88E-15
	Porter	0.2294	4.47E-04		Porter	0.2121	2.38E-10

Tablo E.1.22. *NoStem ve gövdeleme algoritmalarının PL2 terim ağırlıklandırma modeli için nDCG@100 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1676	1.32E-06	WSJ	KStem	0.3524	7.87E-04
	Lovins	0.1668	2.30E-06		Lovins	0.3372	8.78E-04
	Porter	0.1703	5.30E-08		Porter	0.3667	2.57E-03
CW12B	KStem	0.0917	1.86E-04	MQ07	KStem	0.2690	<1.00E-20
	Lovins	0.0919	1.03E-03		Lovins	0.2713	<1.00E-20
	Porter	0.0947	1.98E-04		Porter	0.2773	<1.00E-20
NTCIR	KStem	0.3348	1.56E-05	MQ08	KStem	0.2531	3.54E-12
	Lovins	0.3272	2.74E-08		Lovins	0.2534	4.80E-16
	Porter	0.3350	7.62E-06		Porter	0.2579	2.61E-14
GOV2	KStem	0.3569	5.75E-06	MQ09	KStem	0.3327	6.17E-16
	Lovins	0.3501	5.83E-06		Lovins	0.3333	9.00E-20
	Porter	0.3681	5.04E-06		Porter	0.3399	6.35E-16

Tablo E.1.23. *NoStem ve gövdeleme algoritmalarının PL2 terim ağırlıklandırma modeli için nDCG@20 metriğinde Oracle performansı ve t-test sonuçlarının p değerleri.*

Sorgu Seti	Gövdeleme	Oracle	*p*-değeri	Sorgu Seti	Gövdeleme	Oracle	*p*-değeri
CW09B	KStem	0.1504	4.81E-05	WSJ	KStem	0.3707	9.79E-06
	Lovins	0.1530	1.02E-06		Lovins	0.3545	5.74E-04
	Porter	0.1533	4.51E-07		Porter	0.3852	4.19E-05
CW12B	KStem	0.1052	1.48E-04	MQ07	KStem	0.2150	<1.00E-20
	Lovins	0.1053	3.53E-05		Lovins	0.2179	<1.00E-20
	Porter	0.1083	1.49E-05		Porter	0.2229	<1.00E-20
NTCIR	KStem	0.3079	1.14E-05	MQ08	KStem	0.2011	1.63E-11
	Lovins	0.3043	8.14E-08		Lovins	0.2035	8.56E-14
	Porter	0.3096	3.17E-06		Porter	0.2092	4.17E-14
GOV2	KStem	0.3649	2.00E-06	MQ09	KStem	0.2559	8.73E-16
	Lovins	0.3589	2.21E-06		Lovins	0.2543	2.55E-18
	Porter	0.3756	9.90E-06		Porter	0.2615	5.68E-17

EK-2. Gövdeleme Algoritmalarının Performans Değerleri

Tablo E.2.1. *Gövdeleme algoritmalarının* MAP performans değerleri.

Terim Ağır. Modelleri	**Gövdeleme**	**CW09B**	**CW12B**	**NTCIR**	**GOV2**	**WSJ**	**MQ07**	**MQ08**	**MQ09**
BM25	NoStem	0.0857	0.0244	0.2217	0.2090	0.1799	0.1841	0.2668	0.1921
	KStem	0.0888	0.0272	0.2461	0.2409	0.2136	0.1917	0.2624	0.1983
	Lovins	0.0814	0.0247	0.2139	0.2275	0.1975	0.1847	0.2483	0.1872
	Porter	0.0867	0.0273	0.2351	0.2482	0.2232	0.1984	0.2633	0.1991
DFIC	NoStem	0.1013	0.0320	0.2610	0.2309	0.2040	0.1640	0.1862	0.1951
	KStem	0.1022	0.0371	0.2929	0.2503	0.2325	0.1666	0.1748	0.2002
	Lovins	0.0963	0.0327	0.2506	0.2328	0.2152	0.1510	0.1642	0.1881
	Porter	0.1027	0.0376	0.2766	0.2524	0.2434	0.1678	0.1712	0.2024
DFRee	NoStem	0.0898	0.0275	0.2377	0.2399	0.2123	0.2057	0.2771	0.2169
	KStem	0.0921	0.0308	0.2593	0.2614	0.2423	0.2104	0.2799	0.2261
	Lovins	0.0864	0.0286	0.2215	0.2468	0.2257	0.1985	0.2675	0.2114
	Porter	0.0905	0.0307	0.2420	0.2694	0.2547	0.2148	0.2840	0.2283
DLM	NoStem	0.0984	0.0395	0.2775	0.2610	0.2267	0.2047	0.2524	0.1969
	KStem	0.0995	0.0479	0.3110	0.2776	0.2556	0.2098	0.2538	0.1995
	Lovins	0.0922	0.0448	0.2614	0.2624	0.2364	0.1926	0.2456	0.1847
	Porter	0.0985	0.0503	0.2911	0.2826	0.2683	0.2140	0.2625	0.2008
DLH13	NoStem	0.0735	0.0214	0.2147	0.2109	0.2061	0.1870	0.2427	0.1941
	KStem	0.0743	0.0240	0.2346	0.2300	0.2371	0.1894	0.2340	0.1970
	Lovins	0.0685	0.0217	0.1999	0.2137	0.2192	0.1749	0.2113	0.1835
	Porter	0.0731	0.0240	0.2181	0.2341	0.2498	0.1896	0.2313	0.1974
DPH	NoStem	0.0993	0.0320	0.2463	0.2462	0.2160	0.2151	0.2826	0.2271
	KStem	0.1033	0.0347	0.2686	0.2747	0.2489	0.2224	0.2848	0.2363
	Lovins	0.0961	0.0322	0.2290	0.2574	0.2310	0.2083	0.2728	0.2230
	Porter	0.1011	0.0348	0.2503	0.2803	0.2616	0.2262	0.2868	0.2398
LGD	NoStem	0.0867	0.0313	0.2443	0.2373	0.2146	0.1992	0.2862	0.1953
	KStem	0.0905	0.0361	0.2766	0.2628	0.2413	0.2091	0.2804	0.2069
	Lovins	0.0865	0.0348	0.2390	0.2500	0.2228	0.1995	0.2752	0.1944
	Porter	0.0926	0.0380	0.2610	0.2736	0.2521	0.2164	0.2864	0.2112
PL2	NoStem	0.0805	0.0287	0.2291	0.1833	0.1691	0.1575	0.1778	0.1826
	KStem	0.0834	0.0335	0.2606	0.2125	0.2022	0.1644	0.1781	0.1908
	Lovins	0.0786	0.0308	0.2266	0.2009	0.1866	0.1549	0.1688	0.1795
	Porter	0.0832	0.0344	0.2490	0.2184	0.2147	0.1690	0.1797	0.1932

Tablo E.2.2. *Gövdeleme algoritmalarının nDCG@100 performans değerleri.*

Terim Ağır. Modelleri	Gövdeleme	CW09B	CW12B	NTCIR	GOV2	WSJ	MQ07	MQ08	MQ09
BM25	NoStem	0.1678	0.0720	0.3052	0.3484	0.3139	0.2871	0.3229	0.3236
	KStem	0.1677	0.0768	0.3186	0.3726	0.3512	0.3001	0.3271	0.3302
	Lovins	0.1566	0.0722	0.2839	0.3613	0.3321	0.2890	0.3104	0.3189
	Porter	0.1660	0.0780	0.3054	0.3852	0.3621	0.3073	0.3299	0.3321
DFIC	NoStem	0.1862	0.0881	0.3140	0.3552	0.3554	0.2624	0.2590	0.3074
	KStem	0.1864	0.0947	0.3266	0.3621	0.3896	0.2677	0.2535	0.3090
	Lovins	0.1750	0.0883	0.2891	0.3490	0.3664	0.2473	0.2360	0.2948
	Porter	0.1856	0.0958	0.3113	0.3651	0.4008	0.2690	0.2508	0.3149
DFRee	NoStem	0.1704	0.0812	0.3182	0.3648	0.3629	0.3100	0.3360	0.3445
	KStem	0.1709	0.0880	0.3296	0.3719	0.3965	0.3182	0.3409	0.3520
	Lovins	0.1641	0.0831	0.2930	0.3601	0.3762	0.3024	0.3241	0.3385
	Porter	0.1718	0.0876	0.3129	0.3837	0.4069	0.3219	0.3441	0.3570
DLM	NoStem	0.1800	0.1044	0.3247	0.3897	0.3793	0.3107	0.3149	0.3004
	KStem	0.1802	0.1145	0.3378	0.3954	0.4108	0.3184	0.3218	0.3007
	Lovins	0.1681	0.1106	0.2948	0.3861	0.3848	0.2992	0.3092	0.2863
	Porter	0.1782	0.1180	0.3191	0.4006	0.4204	0.3225	0.3282	0.3066
DLH13	NoStem	0.1499	0.0682	0.2893	0.3263	0.3552	0.2865	0.3098	0.3143
	KStem	0.1487	0.0752	0.2996	0.3363	0.3903	0.2922	0.3113	0.3180
	Lovins	0.1412	0.0699	0.2660	0.3217	0.3684	0.2722	0.2906	0.3036
	Porter	0.1487	0.0747	0.2828	0.3451	0.4003	0.2916	0.3088	0.3203
DPH	NoStem	0.1854	0.0900	0.3210	0.3796	0.3703	0.3223	0.3396	0.3592
	KStem	0.1895	0.0951	0.3334	0.3995	0.4052	0.3330	0.3450	0.3658
	Lovins	0.1775	0.0905	0.2909	0.3846	0.3851	0.3163	0.3336	0.3526
	Porter	0.1881	0.0954	0.3140	0.4086	0.4179	0.3377	0.3502	0.3738
LGD	NoStem	0.1676	0.0867	0.3166	0.3628	0.3654	0.3042	0.3328	0.3175
	KStem	0.1711	0.0964	0.3357	0.3751	0.3911	0.3196	0.3404	0.3309
	Lovins	0.1626	0.0937	0.3004	0.3693	0.3716	0.3070	0.3302	0.3174
	Porter	0.1736	0.1000	0.3223	0.3924	0.4020	0.3266	0.3452	0.3384
PL2	NoStem	0.1579	0.0793	0.3042	0.3124	0.3047	0.2422	0.2308	0.3070
	KStem	0.1578	0.0884	0.3219	0.3390	0.3422	0.2542	0.2364	0.3187
	Lovins	0.1493	0.0846	0.2893	0.3276	0.3254	0.2397	0.2173	0.3037
	Porter	0.1593	0.0909	0.3103	0.3489	0.3569	0.2598	0.2352	0.3223

Tablo E.2.3. *Gövdeleme algoritmalarının nDCG@20 performans değerleri.*

Terim Ağır. Modelleri	Gövdeleme	CW09B	CW12B	NTCIR	GOV2	WSJ	MQ07	MQ08	MQ09
BM25	NoStem	0.1606	0.0781	0.2927	0.3486	0.3281	0.2247	0.2614	0.2508
	KStem	0.1523	0.0858	0.2938	0.3697	0.3662	0.2307	0.2598	0.2543
	Lovins	0.1460	0.0807	0.2666	0.3670	0.3458	0.2221	0.2422	0.2433
	Porter	0.1497	0.0872	0.2823	0.3839	0.3701	0.2360	0.2622	0.2552
DFIC	NoStem	0.1545	0.0971	0.2800	0.3638	0.3887	0.1987	0.1969	0.2153
	KStem	0.1510	0.1085	0.2819	0.3618	0.4200	0.1999	0.1917	0.2156
	Lovins	0.1418	0.0985	0.2449	0.3568	0.3961	0.1831	0.1745	0.2051
	Porter	0.1487	0.1085	0.2648	0.3678	0.4309	0.1998	0.1879	0.2192
DFRee	NoStem	0.1596	0.0891	0.2964	0.3552	0.3960	0.2415	0.2696	0.2684
	KStem	0.1578	0.0966	0.2915	0.3552	0.4211	0.2455	0.2726	0.2722
	Lovins	0.1521	0.0916	0.2601	0.3506	0.4041	0.2327	0.2558	0.2572
	Porter	0.1562	0.0954	0.2777	0.3686	0.4270	0.2485	0.2777	0.2725
DLM	NoStem	0.1475	0.1175	0.2916	0.3958	0.4111	0.2457	0.2533	0.2129
	KStem	0.1455	0.1336	0.2894	0.3916	0.4456	0.2490	0.2582	0.2095
	Lovins	0.1351	0.1245	0.2531	0.3848	0.4219	0.2312	0.2449	0.1972
	Porter	0.1445	0.1365	0.2747	0.3979	0.4525	0.2532	0.2633	0.2142
DLH13	NoStem	0.1333	0.0720	0.2646	0.3105	0.3820	0.2217	0.2413	0.2343
	KStem	0.1294	0.0778	0.2648	0.3119	0.4123	0.2234	0.2422	0.2337
	Lovins	0.1258	0.0721	0.2386	0.3059	0.3913	0.2050	0.2184	0.2220
	Porter	0.1339	0.0768	0.2515	0.3214	0.4199	0.2209	0.2408	0.2353
DPH	NoStem	0.1714	0.0994	0.2933	0.3845	0.4021	0.2553	0.2711	0.2783
	KStem	0.1792	0.1077	0.2915	0.3954	0.4367	0.2600	0.2734	0.2838
	Lovins	0.1684	0.1008	0.2581	0.3862	0.4173	0.2461	0.2628	0.2695
	Porter	0.1765	0.1064	0.2783	0.4073	0.4419	0.2636	0.2781	0.2878
LGD	NoStem	0.1508	0.0993	0.2933	0.3526	0.3953	0.2386	0.2673	0.2366
	KStem	0.1488	0.1092	0.3006	0.3577	0.4183	0.2488	0.2715	0.2448
	Lovins	0.1432	0.1053	0.2688	0.3588	0.3953	0.2377	0.2603	0.2315
	Porter	0.1515	0.1124	0.2870	0.3803	0.4245	0.2549	0.2772	0.2497
PL2	NoStem	0.1417	0.0892	0.2841	0.3167	0.3189	0.1914	0.1795	0.2302
	KStem	0.1379	0.1006	0.2892	0.3331	0.3558	0.1971	0.1812	0.2386
	Lovins	0.1323	0.0942	0.2599	0.3237	0.3347	0.1853	0.1670	0.2224
	Porter	0.1378	0.1019	0.2779	0.3450	0.3689	0.2021	0.1835	0.2387

ÖZGEÇMİŞ

ORCID NO: 0000-0003-1426-0279

Ad Soyad : Gökhan GÖKSEL

Yabancı Dil : İngilizce

Eğitim

- Lise: Nazilli Anadolu Lisesi (2004-2008)
- Lisans: Anadolu Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği (2008-2013)
- Yüksek Lisans: Anadolu Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Yazılımı Bilim Dalı (2015-2016)

Bilimsel Faaliyetler

Bildiriler:

- Aydın, A., Göksel, G., Arslan, A., & Dinçer, B. T. (2020). ESTUCeng at the NTCIR-15 WWW-3 Task: Experimenting with new document quality features. *Proceedings of the 15th NTCIR Conference on Evaluation of Information Access Technologies,* (s. 253-257).
- Çıplak G., Aydın, A., & Arslan, A. (2019). The usage statistics of new HTML5 semantic elements in the ClueWeb12 dataset. *International Conference on Advanced Technologies, Computer Engineering and Science,* (s. 173-176)