KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

EDU.TR UZANTILI WEB SAYFASI İÇERİKLERİNDE ARAMA SONUÇLARI SIRALANMASI OPTİMİZASYONU

EMİNE ŞEYMA ALTINTAŞ

KOCAELİ 2021

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

EDU.TR UZANTILI WEB SAYFASI İÇERİKLERİNDE ARAMA SONUÇLARI SIRALANMASI OPTİMİZASYONU

EMİNE ŞEYMA ALTINTAŞ

Doç. Dr. Halil YİĞİT
Danışman, Kocaeli Üniversitesi …….
Doç. Dr. Ali ÇALHAN
Jüri Üyesi, Düzce Üniversitesi …….
Dr. Öğr. Üyesi Süleyman EKEN
Jüri Üyesi, Kocaeli Üniversitesi …….

Tezin Savunulduğu Tarih: 21.06.2021

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışması, geliştirilen prototip arama motoru uygulamasında aranan sonuçların sıralanmasını optimize etmek ve daha verimli hale getirmek amacıyla gerçekleştirilmiştir.

Yüksek lisans öğrenimim boyunca ve tez çalışma sürecimde benden maddi manevi desteğini esirgemeyen danışman hocam sayın Doç.Dr. Halil YİĞİT'e sonsuz teşekkür ve minnetlerimi sunarım.

Tez sürecimde her zaman yanımda olan, benden sevgisini ve desteğini bir an bile eksik etmeyen, bilgi ve birikimiyle çalışmalarıma ışık olan sevgili eşim Abdurrahman ALTINTAŞ'a sonsuz teşekkürlerimi sunarım.

Hayatım boyunca yanımda olan, başarılarımın arkasında en büyük katkıya sahip olan ve benden dualarını ve desteklerini esirgemeyen aileme teşekkürü borç bilirim.

Haziran-2021 Emine Şeyma ALTINTAŞ

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

CSNET	:Computer Science Network (Bilgisayar Bilim Ağı)
DNS	:Domain Name Service (Alan Adı Servisi)
FTP	:File Transfer Protocol (Dosya Transfer Protokolü)
HITS	:Hyperlink-Induced Topic Search (Köprü Kaynaklı Konu Arama)
HTML	:Hypertext Markup Language (Hiper Metin İşaret Dili)
HTTP	:Hyper-Text Transfer Protocol (Hiper-Metin Transfer Protokolü)
HTTPS	:Secure Hyper Text Transfer Protocol (Güvenli Metin Aktarma Protokolü)
ID	:Identification Number (Kullanıcı Numarası)
IIS	:Internet Information Services (İnternet Bilgi Servisleri)
IP	:Internet Protocol (İnternet Protokolü)
IR	:Information Retrieval (Alaka Düzeyi)
NoSQL	:Not only SQL (İlişkisel Olmayan Veritabanı)
NSFNET	:National Science Foundation Network (Ulusal Bilim Vakfı Ağı)
OPIC	:On-Line Page Importance Computation
SALSA	:Stochastic Approach for Link-Structure Analysis (Bağlantı Yapısı Analizi)
SEO	:Search Engine Optimization (Arama Motoru Optimizasyonu)
SÖD	:Site Önem Derecesi
SQL	:Structured Query Language (Yapısal Sorgulama Dili)
TCP	:Transmission Control Protocol (Veri İletim Kontrol Protokolü)
TÜİK	:Türkiye İstatistik Kurumu
URL	:Uniform Resource Locator (Tek Düzen Kaynak Bulucu)
WEKA	:Waikato Environment for Knowledge Analysis
WWW	:World Wide Web (Dünya Çapına Ağ)
YANDEX	:Yet Another Index

EDU.TR UZANTILI WEB SAYFASI İÇERİKLERİNDE ARAMA SONUÇLARININ SIRALANMASI OPTİMİZASYONU

ÖZET

Arama motorları bilgi bulmak ve bilgiye en hızlı ve en kolay şekilde ulaşmak için kullanılan en önemli İnternet hizmetlerinden biridir. Arama motorları tarama, indeksleme ve sıralama fonksiyonları üzerinde çalışır. Bilginin aranmasında sıralama önem kazanmaktadır. İnternet'te web sayfalarını sıralamak için bağlantı temelli algoritmalar kullanılmaktadır. Bu tez çalışmasında, arama motorunun işleyişi ve sıralama teknikleri ele alınmıştır. Bu kapsamda, prototip bir arama motoru uygulaması hazırlanmıştır. Prototip uygulama ile yaklaşık dokuz yüz bin "edu.tr" alan adına sahip web sayfası indirilmiştir. İndirilen "edu.tr" web sayfalarında yaklaşık üç milyon benzersiz kelime tespit edilmiştir. Yapılan arama sonuçlarının sıralanması Google tarafından da kullanılan bağlantı temelli sıralama algoritmalarından PageRank ile gerçekleştirilmiştir. Çalışmada yeni bir sıralama algoritması önerilmiş olup, sonuçlar karşılaştırılmıştır. Önerilen yaklaşımın, PageRank algoritmasına kıyasla sayfa önem derecesi birikmesi problemini önemli ölçüde azalttığı sonucuna ulaşılmıştır.

Anahtar Kelimeler: Arama Motoru, Bilgi Arama, Sıralama, Web.

OPTIMIZATION OF ORDERING SEARCH RESULT IN EDU.TR WITH EXTENSION WEB PAGE CONTENTS

ABSTRACT

Search engines are one of the most important Internet services used to find information and to access information in the fastest and easiest way. Search engines work on crawling, indexing and ranking functions. Ranking becomes important when searching for information. Link-based algorithms are used to rank web pages on the Internet. In this thesis, search engine functions and ranking techniques are discussed. A prototype search engine application has been created in this context. With this prototype application, approximately nine hundred thousand web pages with the "edu.tr" domain name were downloaded. Approximately three million unique words were identified on the downloaded "edu.tr" web pages. The ranking of the search results was made with PageRank, which is one of the link-based ranking algorithms also used by Google. A new ranking algorithm was proposed in the study and the results were compared. It is concluded that the proposed approach significantly reduces the page rank accumulation problem compared to the PageRank algorithm.

Keywords: Search Engine, Search For Information, Ranking, Web.

GİRİŞ

Günümüzde İnternet kullanımının artması ile birlikte bilgiye ulaşma hızı da artmıştır. Dünya nüfusunun toplamda %65,6'lık kısmı İnternet kullanmaktadır [1]. İnternet kullanıcılarının, bilgiye en hızlı ve kolay ulaşma yöntemi ise arama motorları olmaktadır [2]. Her İnternet kullanıcısının günde en az bir defa dahi olsa arama motorunu kullandığı göz önüne alınırsa bilgi kirliliğinin oldukça yüksek olduğu veri ortamında, aranan bilgiye doğru ve hızlı şekilde ulaşması önemli hale gelmiştir.

Dünya arama motoru pazarının yaklaşık %90'ına sahip olan Google [3], webi verimli bir şekilde taramak ve dizine eklemek ve mevcut sistemlerden çok daha tatmin edici arama sonuçları üretmek için tasarlanmıştır. Google, geniş veri yığınının depolanması ve işlenmesi için birbirine entegre bir yapı üzerine oluşturulmuştur [4]. Bu yapı birçok arama motoruna örnek teşkil etmiştir. Ülkemizde aktif hizmet veren yerli bir arama motorunun olmamasından dolayı bazı verilerin değerlendirilmesi noktasında dışa bağımlı hale gelinmiştir. Bundan dolayı, yerli bir arama motoru geliştirilmesi ve yaygın olarak kullanılması ülkemiz adına önemlidir.

Arama motoru, dünyada büyük verilerin toparlandığı, işlendiği, sonuçlar üretildiği birçok alanda farklı çıktıları olan ve sınırlı sayıda örneği bulunan ticari veya akademik sistemlerdir [5]. Arama motorunu bir yazılımdan farklı kılan büyük anonim verinin toplanması, indekslenmesi ve arama işlevi esnasında ulaşılmasının sağlanmasıdır. Verinin toplanması, indekslenmesi ve arama işlevlerinin tamamı klasik anlamda bir yazılımın tek başına yapabileceği bir iş olmasından ziyade birçok yazılımın eşzamanlı çalıştırılması ile mümkün olabilecek geniş kapsamlı bir süreçtir.

Arama motorları konusunda yapılan bazı çalışmalar aşağıda verilmektedir.

Arama motorları mimarisi, yapısı, sıralaması gibi arama motorlarını inceleyen ve geliştirilmesi üzerine dünyada ve ülkemizde yapılmış birçok akademik çalışma bulunmaktadır. Bu çalışmalardan biri temel olarak kabul edilen ve sonraki çalışmalara kaynak gösterilen 1998 yılında Sergery Brin ve Lawrance Page tarafından yayınlanan,

arama motoru yapısı ve uygulaması anlamında yeni bir bakış açısı getiren çalışmalarıdır [4]. Google'ın kuruluş çalışması olarak da nitelendirilen bu çalışma, öncekilerden farklı olarak; geniş veri yığını depolanması, işlenmesi için birbirine entegre bir yapı oluşturmuştur.

PageRank olarak adlandırılan ve sonraları Google içerisinde de kullanılan web sayfalarının link (bağlantı) modeline dayalı bir puan ve sıralama algoritması duyurulmuştur [6]. PageRank site içeriğinden bağımsız olarak ilgili siteye verilen bağlantılara dayalı bir puanlama sistemidir [7].

Arama motorları ihtiyaç duydukları veriyi elde etmeleri için web üzerinden sayfaları kayıt altına almalıdır. Web sayfalarını tarayarak kayıt altına alan uygulamalar "Web Crawler" olarak adlandırılmaktadır [8]. Boldi ve diğerleri yaptıkları çalışmada, Java programlama dilinde "UbiCrawler" olarak adlandırdıkları web tarama uygulamalarını sunmuşlardır. Sık değişen web sayfaları için ve bazı web sayfalarından alınan hataları düzeltmek için bir çözüm önermişlerdir [9].

Karlık yüksek lisans tezinde, tarayıcı uygulaması olan tarama kuyruğunda bulunan web sayfalarını indiren bir web crawler uygulamasını anlatmıştır [10]. Daha sonra, arama motoru indeksleme mimarisini anlatmış ve ara yüze sahip bir web uygulaması geliştirmiştir. Bu çalışmada, arama sıralama sonuçlarında rank değeri hesaplamasına link (bağlantı temelli derecelendirme) hesabı dâhil edilmemiştir.

Razbonyalı yüksek lisans tez çalışmasında, yatay ve dikey arama motorları arasındaki farkları detaylıca incelemiştir. Çalışma sırasında geliştirilen arama motoru yapısına da geniş olarak yer vermiştir. Çalışmada Weka hazır kütüphanesinden yararlanmıştır [11].

İnternet'te bilgiyi ararken kullanıcıların gerçek ihtiyaçlarına uygun bilgileri bulmak için eleme işlemini kendilerinin yapması gerektiği düşünüldüğünde arama sonuçlarında gelen sayfaların otomatik özetlemesi önemlidir. Pembe yapmış olduğu akademik çalışmasında, web aramaları için özgün bir otomatik özetleme çalışmasına yer verilmiştir [12].

Baker'in 2017 yılında yayınlanmış olan doktora tez çalışmasında, web tarama robotu ve web sıralama algoritması gerçekleştirilmiştir. Geliştirilen sıralama algoritması PageRank ve HITS algoritmaları ile kesinlik ve duyarlılık ölçütleri kullanılarak karşılaştırma yapılmıştır [13].

Son yıllar da arama motorları üzerine yapılan akademik çalışmalarda ise daha çok web siteleri için arama motoru optimizasyonu üzerinde durularak SEO yöntemleri üzerinde çalışmalar gerçekleştirilmiştir [14,15]. Vuran yüksek lisans tez çalışmasında, SEO yöntemleri derlenerek, arama motoru deney ortamı oluşturulmuştur [16].

Tez çalışmasının amacı, arama motorunun çalışma prensibini ortaya koymak ve prototip bir arama motoru geliştirerek sıralama algoritmalarını incelemektir.

Günümüzde, eskiye oranla bilgiye ulaşma kolaylığının artması, İnternet vasıtasıyla bilginin dolaşımının hızlanması, bilgiye hızlı ulaşım gereksinimini gün be gün artırmaktadır. Arama motorları sayesinde, milyarlarca bilginin olduğu İnternet ortamında bilgiye ulaşmak kolay ve hızlı hale gelmiştir. Ülkemizde 2020 yılında yapılan arama motoru kullanım istatistiklerine bakıldığında %85 gibi yüksek bir oranla ilk sırada Google yer almaktadır. Sıralama Yandex %12, Yahoo %1,3 olarak devam etmektedir [17]. Sıralamada kullanımda olan tamamen yerli bir arama motoru olmaması maddi olarak dışa bağımlı olmanın yanı sıra bazı bilgileri değerlendirme konusunda bizleri dışa bağımlı hale getirmektedir. Bu sebepler değerlendirildiğinde yerli bir arama motoru geliştirilmesi önemlidir.

Google örneğinde olduğu gibi arama motoru sadece arama ve elde edilen sonuçlar olmayıp reklam, iletişim ve medya gibi faaliyet alanlarını da ilgilendirmekte ve ekonomik bir değer ifade etmektedir.

Arama motorlarının başarısını etkileyen etkenlerden biri ise hiç şüphesiz aranılan bilgiye en hızlı ve kısa yoldan ulaşmaktır. Bu kavram da sıralama olarak karşımıza çıkmaktadır. Tez kapsamında, Türkiye üniversitelerinin "edu.tr" uzantılı web sayfalarının taranması, depolanması, indekslenmesi ve arama sonuçlarını bağlantı temelli algoritma kullanarak sıralanması işlemlerini gerçekleştiren uygulama geliştirilmiştir. Arama sonuçlarının sıralanması sırasında Google tarafından geliştirilen PageRank algoritmasının gerçeklemesi yapılmıştır. Uygulamada sayfa

önem derecesi birikmesi problemini önemli ölçüde azaltan yeni bir sıralama algoritması SÖD önerilmiş olup PageRank formülü ile karşılaştırma yapılmıştır.

Tez çalışması giriş bölümünde çalışmanın genel kapsamı, gerekliliği konusu ve literatür taraması yapılmıştır. Birinci bölümden başlayarak İnternet ve gelişimi, ikinci bölümde ise arama motorları hakkında bilgiler ve çeşitlerinden bahsedilmiştir. Üçüncü bölümde arama motoru mimarisinden bahsedilmiş olup dördüncü bölümde sıralama algoritmaları incelenmiş ve karşılaştırmalar yapılmıştır. Beşinci bölümde ise tez çalışması kapsamında geliştirilen prototip arama motoru mimarisi ve çalışma prensibi açıklanmıştır. Altıncı ve son bölümde ise sıralama algoritmalarının sonuçları değerlendirilmiştir.

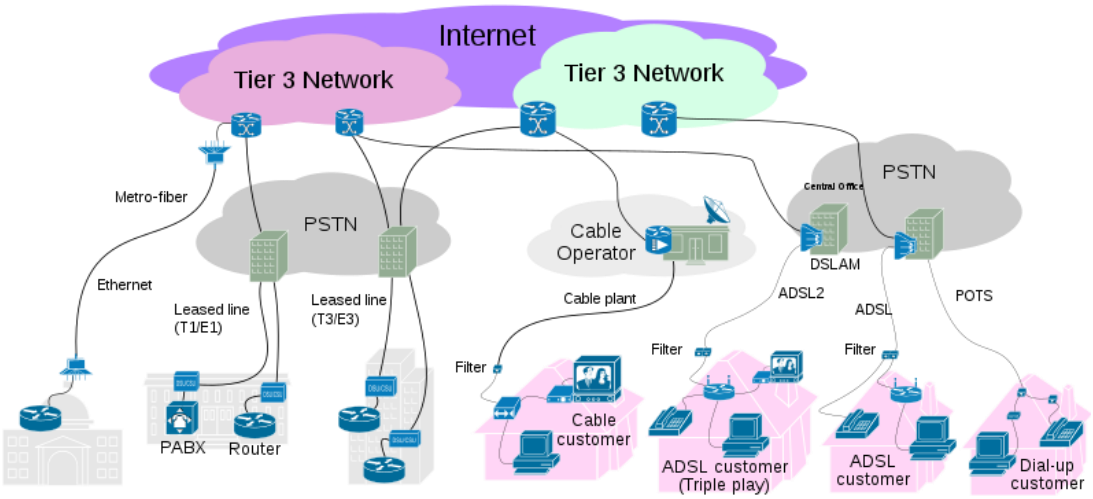
1. TÜRKİYE'DE İNTERNET VE GELİŞİMİ

1.1. İnternet Nedir? Nasıl Çalışır?

Türk Dil Kurumu Sözlük anlamına göre İnternet "tüm dünyada bilgisayar ağlarını ve kurumsal bilgisayar tesislerini birbirine bağlayan elektronik bir iletişim ağı, genel ağ" şeklinde tanımlanmıştır. Temelde İnternet donanımı fark etmeksizin aralarında iletişim sağlayan elektronik bir ağdır.

Kurulduğu yıllarda savunma sistemlerini birbirine bağlayan kablolu veya kablosuz bir bağlantı modeliyken günümüzde küresel çapta bir ağ halini almıştır. İnternet herkesi ve her yeri birbirine bağlar. Temelde uç noktaların birbirine bağlanması ve kullanıcıların da bu uç noktalara bağlanması olarak açıklanabilir. İnternet TCP/IP adı verilen protokol üzerine inşa edilmiştir. Bağlantı ve veri akışı TCP/IP protokolü ile birlikte HTTP, FTP, DNS, MAIL, gibi uygulama protokolleri ile sağlanır.

Kullanıcılar İnternet sayesinde farklı konum ve cihazlarda bulunmalarına rağmen TCP/IP protokolü gibi standart protokolleri kullanarak aralarında bilgi akışı sağlayabilirler. Şekil 1.1 İnternet bağlantısı erişim modelini göstermektedir.



Şekil 1.1. İnternet Erişim Modeli [18]

1.2. İnternetin Tarihsel Gelişimi

İnternetin tarihi 1950'li yıllara dayanır. O yıllarda savunma sistemlerinin geliştirilmesi sürecinde ortaya çıkmış ABD, İngiltere ve Fransa'da çeşitli laboratuvarlarda geliştirilmiştir [19].

1960'lı yılların başında ARPANET isminde IP protokolünü kullanan ilk ağ çalışması duyurulmuştur. ARPANET üzerinden ilk mesaj Los Angeles'taki Kaliforniya Üniversitesi laboratuvarından Stanford araştırma enstitüsünde bulunan bir bilgisayara gönderildi [20].

1970'li yıllarda ARPANET, Npl, Cyclades, Merit, Tymnet ve Telenet gibi birçok haberleşme protokolü paket anahtarlamalı ağ teknolojisi ile geliştirildi. Paket anahtarlama teknolojisi internetin gelişiminde önemli bir aşamadır [21].

ARPANET farklı ağların büyük ağlara bağlanması konusunda olanak sağlayarak internetin gelişimini sağlamıştır. 1981 yılında CSNET isimli ağ sağlayıcısının Amerikan menşeili National Science Foundaditon (NSF) tarafından desteklenmesiyle ARPANET'e erişim genişlemiştir [21].

1982 yılına gelindiğinde ARPANET için standart ağ protokolü olarak TCP/IP duyuruldu. 1980'lerin sonuna doğru ise ARPANET ve NSFNET projesi ile gelişen İnternet altyapısında ticari internet servis sağlayıcılar kurulmaya başlandı [22].

1990'da ARPANET, 1995'de ise NSFNET ömrünü tamamlamışlardır. Böylece günümüz İnternet dünyası için veri erişiminin önündeki engeller ortadan kalktığından 90'lardan itibaren eposta, anlık mesajlaşmalar, video konferans ve sosyal ağ iletişim araçlarıyla WWW hızlıca gelişmiştir [22].

Günümüzde internetsiz güne başlanılmamakta, ekonomiye ise internet yön vermektedir. İnternetin böylesine vazgeçilmez olmasının nedeni sağladığı kolaylık ve hızdır. Alışveriş, iletişim ve hatta sosyalleşme artık internet üzerinden yapılmaktadır.

1.3. Tarayıcı, URL ve Web Kavramları

İnternete ulaşım aracı olan 1990 yılından itibaren hayatımızda bir kavram olan tarayıcılar, web sayfalarında yer alan verilere ulaşma, indirme ve ya yüklemeye

yarayan yazılımlardır. Teknik olarak ifade edecek olursak HTML den http ye kadar bütün internet protokollerini destekleyen, ağ sunucularındaki web sayfalarının açılmasını sağlayan yazılımlardır. Günümüzde birçok tarayıcı olmasına karşın en çok tercih edilenlerden bir kaçı Google Chrome, Opera, Safari, Microsoft Internet Explorer, Firefox'tur. Kullanıcılar tercih ettikleri tarayıcının güvenli olması, hızlı olması gibi seçenekleri dikkate alırlar.

URL terimi, ulaşılmak istenen web sayfasının adres linkini ifade eder. Bir URL alan adresinin her tarayıcıda düzgün olarak çalışabilmesi için en fazla 2083 karakter olması gereklidir. Önceleri bir web sayfasına ulaşmak için URL adresini bilmemiz gerekliyken günümüz İnternet dünyasında ulaşılmak istenen bilgi her neyse arama motoruna yazılarak ulaşılmaktadır. URL ile arama motorlarının bu denli yakın olması SEO kavramı gibi arama motoruna yönelik yaklaşımları ortaya çıkarmıştır.

URL protokol, alan adı, klasör, dosya ve parametrelerden oluşur. Protokol HTTP veya HTTPs olacağı gibi başka protokollerde olabilir. Web sitesi için örnek URL "http://www.ornekurl.com/giris.html" olacağı gibi medya içeriği için örnek URL "rtmp://www.canliyayin.com/tvcanli" olabilir.

Tablo 1.1. URL için tipik alan adı uzantıları

Uzantı	**Açıklaması**
.edu	(education) Eğitim kurumları (üniversite vb.) tarafından kullanılır.
.gov	(goverment) Devlet kurumları tarafından kullanılır.
.com	(company) Şirketler ve özel kurumlar tarafından kullanılır.
.mil	(military) Askeri web sitelerinde kullanılır.
.org	(organization) Dernek, vakıf vb. kar amacı gütmeyen kuruluşlar tarafından kullanılır.
.net	(network) İnternet ve ağ hizmet sağlayıcılar tarafından kullanılır.
.k12	ABD, Kanada, Türkiye ve Avustralya'daki ilk ve orta dereceli eğitim kurumlarına tahsis edilen alan adıdır.

Web, İngilizcedeki aynı ifadeyi taşıyan web kelimesinden geldiği düşünülmektedir. Kumaş dokusuna benzetilerek bilgisayar ağını temsil ettiği söylenebilir. Ancak bu ifade bugünkü webi açıklayamamaktadır. Web, İnternet üzerindeki servislerden

biridir. İnternet üzerinde metin, grafik ve medya gibi içeriğin uzaktaki bilgisayarlarca açılmasını sağlayan servistir denilebilir.

Web, web sayfasını içerir. Web sayfasında bulunan içeriğin ulaşım kanalı webdir. Web sayfaları içerisinde başka web sayfalarına ait link olarak adlandırılan çeşitli bağlantılara sahiptir. Bu bağlantılar aracılığıyla kullanıcının web de gezinmesi sağlanır. Arama motorlarının da temel kaynağı web sayfası ve bağlantı linkidir.

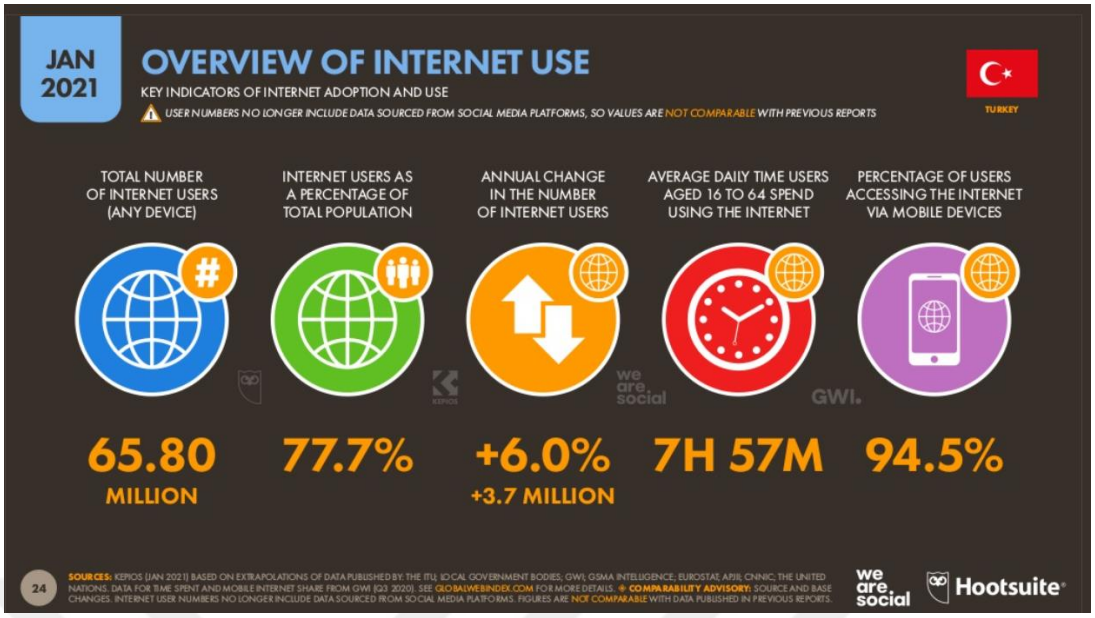
Web sayfaları tarayıcılar ile gezinilir. Örneğin uzak doğudaki bir ülkeye ulaşım 12 saat veya daha fazla sürerken, uzak doğuda bulunan bir web sayfasının ziyaret edilmesi birkaç saniye almaktadır. Muazzam bir iletişim ve bilgi kaynağı web ile parmaklarımızın ucundadır.

Tarayıcı, URL vasıtasıyla web de bulunan bir içeriğe erişim sağlar. Bu web sayfası olabileceği gibi başkaca bilgi ve iletişim servisleri olabilir. Ancak günümüz İnternet kullanım alışkanlıkları tarayıcı ile doğrudan arama motorunun kullanımı şekline evirilmiştir.

1.4. Türkiye'de İnternet Kullanımı

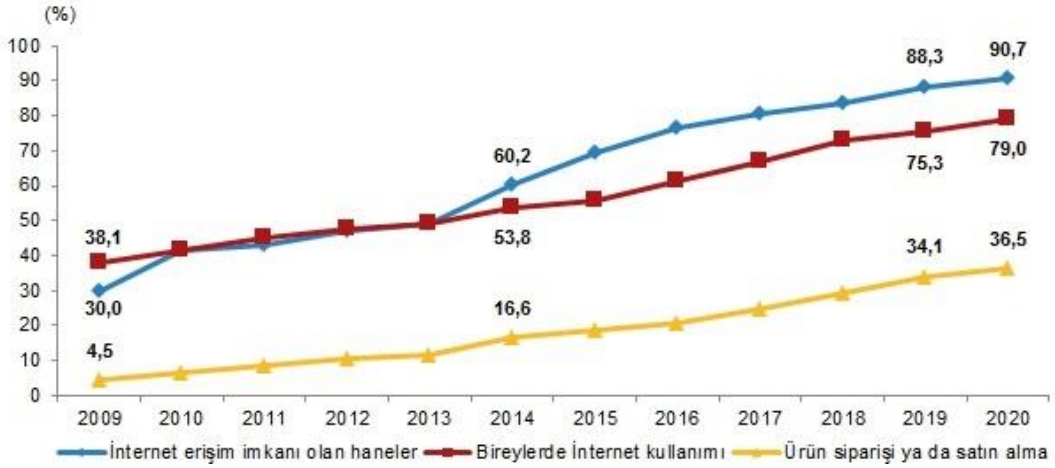
Türkiye'de İnternet kullanımı ile ilgili her sene düzenli çalışmalar yapılmaktadır. Ülkemizden istatistiklerle İnternetin hayatımızda kapladığı yer açıklanmaya çalışılacaktır. Yapılan çalışmalardan iki tanesi incelenerek Şekil 1.2 ve Şekil 1.3'te açıklanmıştır.

Şekil 1.2 de gösterilen 2021 Türkiye dijital raporuna göre Ocak 2021 de 84,69 milyon nüfusa sahip ülkenin %77,7'lik kısmı yani 65,80 milyonu İnternet kullanıcısıdır. Aynı raporun 2020 Ocak ayı raporuna göre İnternet kullanıcı sayısı 3,7 milyon artmıştır. Belirtilen raporda 16 ile 64 yaş arası kullanıcıların İnternette ne kadar zaman harcadıklarını da belirterek ortalama yaklaşık sekiz saat olduğu görülmektedir. Önceki yıllara oranla İnternet kullanım sayısının ve ortalama kullanım sayısının artmasına pandemi sürecinde uzaktan çalışma ve ya uzaktan eğitimin de katkısı olduğu aşikârdır.



Şekil 1.2. Türkiye'de İnternet kullanımına genel bakış [23]

2020 TÜİK verilerine göre 15-64 yaş arası toplam nüfusun %67,7'lik kısmını oluşturmaktadır [24]. Ülkemizde toplam nüfusun %77,7'lik kısmının internet kullanması [25], ülkemizde internetin oldukça yaygın ve etkili şekilde kullanıldığını göstermektedir.



Şekil 1.3. 2020 TÜİK Hane Halkı Bilişim Teknolojileri Kullanım Araştırması [25]

TÜİK tarafından hazırlanmış olan ve her sene ağustos ayında yayınlanan "Hane Halkı Bilişim Teknolojileri Kullanım Araştırması"na göre 2020 yılında Türkiye de internet kullanım oranı bir önceki seneye göre %3,7 artarak %79,0 olmuştur. Türkiye de evden İnternete ulaşım imkânı ise %90,7'dir. Evden İnternete ulaşma imkânı birçok ülkeye

göre geri dururken önceki yıla oranla artış mevcuttur. Yine aynı araştırmada İnternet üzerinden ürün sipariş verme ve ya alma oranı %36,5 olduğu yazmaktadır.

2. ARAMA MOTORLARI

2.1. Arama Motoru Nedir?

Günümüz dünyasında İnternetin vazgeçilmez bir kaynak olmasından dolayı arama motorları hayatın merkezine yerleşmişlerdir. Aranan bilgiyle web içeriğini eşleştiren ve kullanıcıya bulmak istediği sonuçları üreten geniş kapsamlı yazılım ve donanım birleşiminden oluşan büyük çaplı sistemleri arama motoru olarak adlandırabiliriz.

Binlerce aramaya birkaç saniye içerisinde cevap verebilen bu sistemlerde aranan bilginin bulunmaması nadiren yaşanır. Arama motorundan beklenen aranan bilgiyle en alakalı sonuçların bulunmasıdır. Arama motoru dünyasında aranılanın bulunması önemlidir ve tercih sebebidir.

Çalışma prensibi aranan veriyi arama anında bulmak yerine önceden verileri derleyerek kaydetmeye, indekslemeye ve çeşitli matematiksel hesaplara dayanır. Günümüzde tahminen altı milyar web sayfası olduğu tahmin edilmektedir [26].

2.2. Arama Motoru Tarihsel Gelişimi

Bilinen ilk arama motoru olan Archie 1990 yılında Alan Emtage isminde bir üniversite öğrencisi tarafından kurulduğu bilinmektedir. Archie kelimesi İngilizce "archive" kelimesinden esinlenilmiş olduğu düşünülmektedir. Archie günümüz arama motorlarından farklı olarak dosya arama üzerine inşa edilmişti. Şimdilerde oldukça basit görünen bu yapı o dönemlerde devrim niteliğindeydi [27].

1993 yılına gelindiğinde Massachusetts Teknoloji Enstitüsünden Matthew Gray, indeks oluşturmak için "Wandex" isminde bir indeks yapısı üretmek için ilk internet botunu hazırladı [28].

1994 yılında ise Jumpstation adında üç aşamalı olduğu bilinen arama motoru kuruldu. Bu arama motorunun en büyük dezavantajı bir sıralama algoritması olmamasıdır [29].

1994 yılında meydana gelen bir başka gelişme ise full text search (tam metin arama) kavramı ile çalışan Web Crawler adındaki arama motorunun geliştirilmesidir. “Web Crawler” ayrıca tarayıcı tabanlı ilk ticari arama motoru olarak da bilinir [28].

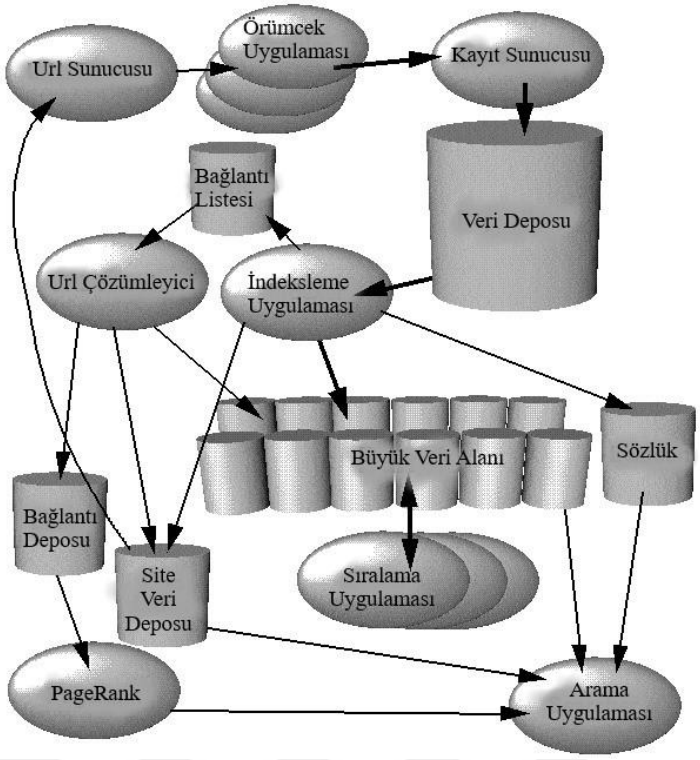
Aynı yıl Carnegie Mellon Üniversitesi’nden Dr. Michael Mauldini tarafından geliştirilen Lycos arama motorunun satışa çıkarılması ticari bir başka girişimdir. Devamında Excite, Altavista, Megellan ve Infoseek adında pek çok arama motoru kullanıma sunuldu [29].

1995 yılında günümüzde hala hizmet veren Yahoo isimli arama motoru David Filo ve Jeryy Yang tarafından kuruldu [28]. Yahoo arama motorunun insanların aradıkları bilgiye ulaşmalarında başarılı olması kurulduğu yıllarda popüler olmasına neden oldu.

1996 yılında ise günümüzde hatırı sayılır şekilde kullanımda olan Yandex arama motoru kuruldu. Yandex arama motoru tam metin arama ve indekslemeye dayalı günümüz modern arama motorlarından biridir.

Günümüz vazgeçilmez arama motoru olan ve dünya üzerinde en çok kullanılan arama motoru olan Google Stanford üniversitesinde iki öğrenci olan Sergey Brin ve Lerry Page tarafından 1998 yılında bir doktora çalışması olarak hazırlanmış ve ikilinin ticari girişimi neticesinde kurulmuştur. Google arama motorunun bu denli popüler olmasının nedeni arama sonuçlarının kalitesidir. Google arama motoru kurulduğunda tam metin arama özelliğini taşırken günümüze gelindiğinde resim arama, akademik arama, video arama gibi pek çok özelliği barındırmaktadır. Şüphesiz Google’ın bu kadar başarılı olmasının nedeni indeksleme yapısı ve PageRank algoritmasıdır.

Google günümüzde arama ve sıralama algoritmasını paylaşmamakla beraber yüzden fazla algoritma ile desteklediği tahmin edilmektedir. Şekil 2.1’de Google’ın 1998 yılında yayınlanan arama motoru mimarisinin Türkçe’ye çevrilmiş hali gösterilmektedir. Şekil 2.1, bir arama motorunun devasa yapısı ve bileşenlerinin genel görünümünün nasıl olabileceği hakkında fikir vermektedir.



Şekil 2.1. Google Arama Motoru Mimarisi (Türkçe 'ye çevrilmiştir.) [4]

2000 yılında Robin Li ve Eric XU tarafından Çin'de geliştirilen ve sadece Çince hizmet veren Baidu arama motoru kurulmuştur. Çin devletinin Baidu kullanımıyla beraber Google arama motorunun Çin üzerindeki etkinliği azalmıştır.

2009 yılına gelindiğinde ise ilk olarak Kumo adıyla piyasaya çıkan, Microsoft tarafından geliştirilen daha sonra ise ismi Bing olarak değiştirilen arama motoru kuruldu. Microsoft Bing ile "alışkanlıkları değiştirme" iddiası ile yola çıktıysa da halen Google arama motoru küresel çapta baskın arama motorudur.

2.3. Arama Motoru Türleri

Arama motorlarının insanların sınırsız arama isteklerine yanıt verebilmesi için internette bulunan tüm verilere erişmesi gereklidir. Veri erişimi teorik olarak farklı yollarla yapılabilir. Arama motorları örümcek tabanlı, dizin tabanlı ve hibrit olarak sınıflandırılabilir [11]. Günümüzün modern arama motorlarını belirli bir sınıfa indirgemek zordur. Karmaşık yapıları ve doğru sonuca ulaşma arzusu, arama motorlarını çeşitli yapıların bileşimi olarak ön plana çıkarmaktadır.

2.3.1. Örümcek tabanlı arama motorları

Örümcek tabanlı arama motorları, web sayfalarını araştırmak ve kategorilere ayırmak için otomatik yazılım programları kullanır. Arama motorları tarafından web sayfalarına erişmek için kullanılan programlar "örümcek", "tarayıcı", "robot" veya "bot" olarak adlandırılır. Tarayıcı uygulaması örümceğe benzer bir şekilde tüm İnternet ağını dolaşır.

Bir örümcek İnternet'te web sayfasını bulur, onu indirir ve web sayfasında sunulan bilgileri analiz eder. Web sayfası daha sonra arama motorunun veri tabanına kaydedilir. Daha sonra, bir kullanıcı bir arama yaptığında, arama motoru, bir bağlantı sonuçları listesi sunmak için kullanıcının aradığı anahtar kelimeler için web sayfalarının veri tabanını kontrol eder. Sonuçlar (gitmek için önerilen bağlantıların listesi), kullanıcının çevrimiçi bulmak istediklerine "en yakın" ("botlar" tarafından tanımlandığı gibi) olan sayfalarda listelenir. Tarayıcı tabanlı arama motorları sürekli olarak İnternet'te yeni web sayfaları arar ve bu yeni veya değiştirilmiş sayfalarla bilgi veri tabanlarını günceller. Başlıca örümcek tabanlı arama motorları Google ve Yahoo ile birlikte diğer diğer arama motorlarının çoğu örümcek tabanlıdır [29].

2.3.2. Dizin tabanlı arama motorları

İnternet'in ilk yıllarında kullanılan ve genelde insan faktörü ile oluşturulup denetlenen web içeriği için anahtar kelimelerle erişilen arama motoru türüdür. İnsan faktörü dizin içeriğinin oluşturulmasında, güncellenmesinde, genişletilmesinde yani tüm aşamalarında gereklidir.

Bu tür arama motorları, web sayfası listelerini ilgili web sayfası yöneticileri tarafından yapılan gönderimlerden alır. Gönderim, sitenin adresini, başlığını ve kısa bir açıklamasını içerir. Daha sonra gönderiler editörler tarafından incelenir. Bir dizin, yalnızca kendisine gönderilen sayfa açıklamalarındaki sonuçları arar. Bu bir avantajdır çünkü sayfalar manüel olarak gönderildiğinden içeriğin kalitesi, örümcek tabanlı bir arama motoru tarafından alınan sonuçlara göre daha iyi ve daha uygun olacaktır. Ancak dezavantajı, önceden gönderilmiş bir web sayfasında yapılan herhangi bir değişikliğin, tekrar gönderilene kadar güncellenmemesidir. Ayrıca, sıralama yapıldıktan sonra sayfaların sıralaması değiştirilemez. Dizin tabanlı arama motorları

kategori bazlı içerik gösterimi sunmaktadır. Büyük birçok arama motoru arama sonuçlarını daha anlamlı hale getirmek için dizin verisi sağlayan sistemleri kullanmaktadır [29].

Dizin tabanlı arama motorları kategori bazlı içerik gösterimi de sunmaktadır. Aramanın dışında görsel olarak kullanıcıya da hitap ettiği söylenebilir. İnsan faktörü nedeniyle içeriği sınırlı, güncelliği zayıf fakat insan faktörü nedeniyle kalitesi yüksektir.

2.3.3. Hibrit arama motoru

Hibrit arama motoru esasen hem örümcek tabanlı hem de dizin tabanlı arama motorlarının özelliklerini içerir. Günümüz İnternet dünyasında daha çok haber ve alışveriş siteleri için sonuçlar üretildiğinde karşımıza çıkmaktadır. Örneğin bir blog aradığımızda karşımıza alakalı sonuçlar çıkarken, fiyat veya ürün arandığında hem alakalı sonuçlar hem de kategorik dizinden fiyat veya karşılaştırma bilgileri görüntülenmektedir. Günümüzde bazı arama motorları, etkili sonuçlar sağlamak için her iki özelliği de kullanmaktadır [29].

3. ARAMA MOTORU MİMARİSİ

3.1. Verilerin Derlenmesi

Web içeriği birçok türden meydana gelen devasa bir kütüphanedir. Aranan bir veriye ulaşmak için verilerin belirli bir düzende kaydedilmesi ve bulunabilmesi gereklidir. Veri kütüphanesini düzenlemek ve verileri saklamak için öncelikle verilerin webden indirilerek depolanması gerekmektedir. İnternetteki

Tarayıcı uygulaması, tarama kuyruğunda bulunan web sayfalarını indirme işlevini yerine getirmektedir. Tarama kuyruğundan bulunan her sayfa için öncelikle sayfanın başka bir tarayıcı ile indirilip indirilmediğini kontrol etmek için sunucu sistemde bulunan tarama veri dizisi kontrol edilir. Tarama işlemi daha önce yapıldı ise tekrar yapılmasına gerek yoktur. Tarama işlemi daha önce yapılmadı ise adres kuyruktan çıkartılarak web sayfası indirilir ve URL veri dizisine kayıt esnasında alınan ID değerli ile kaydedilir.

Tarama işlemi için 50 ms'lik bir bekleme süresi sabit olarak belirlenmiş olup bir sonraki taramaya bu süre sonunda geçilir. Tarama sırasında karşı sistemlerin engelleme risklerine karşı günlük alan adı başına 200 adet web sayfası indirilme sınırı belirlenmiştir.

Her tarayıcı sayısal bir kimlik değerli ile çalışmaktadır. Tarama işlemi sırasında ve sonrasında sitenin hangi tarayıcı tarafından indirildiği sunucu sistemde kayıtlı ve izlenebilir durumdadır.

Tarama sırasında çeşitli hatalar ile karşılaşılmaktadır. Bu hatalar sayfanın yayından kaldırılması, şifre ile korunması veya erişim kısıtlamaları olabilir. Her alan adı için sıkça hata ile karşılaşılması durumunda aynı gün için hata ile karşılaşılan web sayfasından, ilgili alan adında bulunan sayfalar için indirme işlemi geçici olarak durdurulur. Alınan hatalar için tarayıcı uygulamasındaki bu tip ince düzenlemelerin performansı arttırdığı görülmüştür.

Tarayıcı uygulaması gelen URL'yi kontrol ederek, URL'nin HTML kodu dışında dosya uzantısına sahip (jpg, mov, mp3) olması durumunda indirme işlemini yapmamaktadır. İlgili URL'de tarama kuyruğundan silinir.

Tarayıcı uygulaması web sitelerini ziyaret ettiğinde http protokolü istek mesajında kendini tanıtıcı bir bilgi vermesi gerekmektedir. Genelde arama motorları kendi isimleri ile birlikte Bot (Robot) kavramını kullanmaktadır.

Tablo 3.1 derleme işlemi yapılacak web sayfası adresi için Web derleme kuyruk yapısını göstermektedir.

Tablo 3.1. Web derleme kuyruğu

Veri Adı	**Veri Tipi**
URL	String

Birim zamanda yapılan bir indirme işlemi doğrusal olarak artan veri hacmi sağlar. İnternetteki verinin büyüklüğü yani web sitelerinin sayısı dikkate alındığında birim zamanda çok fazla indirme işlemi gereklidir. Bu yüzden, eşzamanlı olarak çalışan tarayıcılar ile veri depolama işlemi yapılmalıdır. Örnek veri kümesi "edu.tr" adresli web sayfaları ile sınırlı olduğundan az sayıda tarayıcı çalıştırılarak indirme işlemi yapılmıştır. Tarayıcılar indirme işlemini depolama sunucularına yaparlar. Tarama işlemi neticesinde web sitesine bir kimlik numarası (ID) atanır ve HTML kodlarını içeriğinde dokunulmadan kaydedilir. Tablo 3.2 indirilen Web sayfalarının veri tabanına kayıt formatını göstermektedir.

Tablo 3.2. Web sayfası veri tabanı

Veri Adı	**Veri Tipi**	**Açıklama**
ID	Integer	Kayıt no'su
URL	String	Kayıt edilen web sayfası adresi
Durum	Byte	Durum bilgisi (Ayrıştırma İşlemi Bekliyor/Ayrıştırıldı)
ZiyaretTarihi	DateTime	Web sayfasının içeriğinin kaydedildiği/güncellendiği tarih

İndirilen her web içeriği indeksleme işlemi için durum alanı ile işaretlenir. Tarayıcı olarak hazır bir bilgi sistemi kullanılmamış olup web istemcisi şeklinde bir uygulama hazırlanmıştır.

3.2. Verilerin Depolanması

Depolama sunucuları dağıtık yapıda çalışmaktadır. Depolama sunucularında yapılan kayıt işlemini tutan bir adet katalog sunucusu da depolama bilgi sisteminde bulunmaktadır. Katalog sunucusu her bir dosyanın hangi sunucuda depolandığını tutar. Depolama sunucusunda hazır bir SQL ya da NoSQL bilgi sistemi kullanılmamış olup, veriler dosya şeklinde yerel diskte saklanmıştır. Dosyaların katalog bilgisi de servis olarak hazırlanan bir uygulama tarafından sağlanmaktadır.

3.2.1. Hafızadaki verinin dizilerinin kaydedilmesi

Veriler ana bellekte (RAM) tutulduğu için belirli aralıklarla kaydedilmeleri gerekmektedir. Kaydedilen veri dizileri program çalıştırılırken yüklenir. Uygulama başladığında her veri dizisi için yükleme işlemi yapılır. Yükleme işleminde bayt cinsinden kaydedilmiş veri seri hale getirilerek kendi obje (nesne) yapısına çevrilir.

Sözlük, web sayfası ve önem derecesi gibi veri tipleri obje listesi halinde hafızada tutulurken web sayfası verileri, ayrıştırma ve indeksleme verileri diskte tutulur. Diskte tutulan veriler katalog sunucusu üzerinden referanslanarak dosya sunucularına yüklenir.

3.3. Verilerin Ayrıştırılması

Ayrıştırma işlemi HTML kodları ile birlikte bulunan metnin alınması işlemidir. Temizlenen HTML kodları neticesinde salt metin ifadeler derlenir. Ayrıştırma işlemi tarama veri dizisinden henüz üzerinde ayrıştırma işlemi uygulanmamış web sayfaları için yapılır. Şöyle ki ayrıştırma işlemine tabi tutulacak URL ID bilgisi alındıktan sonra diskten ham halde HTML kodlarını barındıran web sayfası “ID.txt” içeriği okunur. Alınan text içeriğin html kodlarından ayrılması için çeşitli kütüphaneler mevcuttur. Bunlardan en bilineni “HtmlAgilityPack” kütüphanesi olup tez çalışmamız kapsamında kullanılmıştır. Alınan text içerik “HtmlAgilityPack” kütüphanesi

yardımıyla ayrıştırılır ve başlık (title), linkler, salt metin (text) içeriği olarak alınır. Alınan başlık (title) ve link (anchor) bilgileri URL ve Link veri dizilerine kaydedilir.

Ayrıştırılan salt metin (text) ise kelime bazlı olarak "Kelime" veri dizisinden her bir kelime için 32 bit sayısal Id değeri alınarak sayısal bir diziye çevrilir. Kelime veri dizisinde olmayan kelimeler de eklenerek Id değeri alınır ve diziye dâhil edilir. Bu işlem bir veri tabanında bulunan "Sözlük" tablosundan kelime kaydı çağırmak gibidir. Her bir kelime için bir Id değeri vardır ve bu değerler alınır. Her kelime için Tablo 3.3'te yapısı verilen sözlük kontrol edilir, yoksa eklenir ve bir kelime kimliği (ID) oluşturulur.

Tablo 3.3. Sözlük

Veri Adı	**Veri Tipi**	**Açıklama**
ID	Integer	Kayıt no'su
Kelime	String	Web sayfasında geçen sözcük
Klasör	Integer	Kelimeye ait kayıt klasörü. Kelimeler belirli klasörlere kaydedilir.

Tablo 3.4 aramalarda kullanılan örnek kelime tablosunu göstermektedir.

Tablo 3.4. Örnek kelime tablosu

Kelime	**ID**
Kocaeli	405421
Üniversitesi	652140
Bilişim	112520
Sistemleri	600421
Mühendisliği	10321

Ayrıştırma öncesi;

<html>

<body>

Kocaeli Üniversitesi

Bilişim Sistemleri Mühendisliği

</body>

</html>

Ayrıştırma sonrası;

Site İçeriği: Kocaeli Üniversitesi Bilişim Sistemleri Mühendisliği

String dizisi: [“Kocaeli”,“Üniversitesi”,“ Bilişim”,“Sistemleri”,“Mühendisliği”]

Integer dizisi: [405421, 652140, 112520, 600421, 10321]

Ayrıştırma sonrasında her bir kelime için ID değerleri ve dizi içerisinde konum bilgileri yer almaktadır. İndis bilgisi olarak örneğin 0. indis “Kocaeli” kelimesi iken 3. indis “Sistemleri” kelimeleridir. İndis bilgisi daha sonra Arama uygulaması içerisinde çoklu kelime aramalarında mesafe hesabı yapılması için kullanılacaktır.

Ayrıştırılan ve sayısal diziye (integer array) çevrilen salt metin içeriği diske kaydedilecektir. Metin ifadesinde geçen kelimeler konum indisleri (kelimenin metin içerisinde geçtiği yer) ile birlikte her biri farklı sunucularda bulunan veri klasörlerine kaydedilir. Tablo 3.5 veri klasörü yapılanmasını göstermektedir.

Tablo 3.5. Klasör yapılanması

Kelime ID Aralığı	**Klasör No**	**Sunucu No**	**Veri Dosyası**	**Boyutu**
(Kelime Adedi / 254)	0-254	0-2	Data[0-n].dat	100KB

Bu çalışmada 254 adet klasör ve 3 farklı sunucu üzerine kayıt işlemi gerçekleştirilmiştir. Klasörlerin ve farklı sunucuların olması ayrıştırma ve indeksleme işlemlerinin eşzamanlı yapılması açısından veri kaynağını bölmemize olanak vermektedir. Her bir Data.dat dosyası 100KB’dan oluşur ve kayıt yapıldığında boyutu 100KB’a ulaştığında yeni dosya açılarak işleme devam edilir. Küçük veri dosyaları kullanmamızın nedeni hata oluşması durumunda hatanın tespitini kolaylaştırmak ve olası veri kaybını azaltmaktır.

Ayrıştırma işlemi sonucunda veri sabit bir kodlama ile kaydedilir. Bu kodlama veri tekrarını olabildiğince azaltmak üzere tasarlanmıştır. Veri kodlaması Data[0-n].dat dosyasında her bir URL için tekrarlanır. Tablo 3.6 indeksleme öncesi veri kodlaması yapısını göstermektedir.

Tablo 3.6. Veri kodlaması (İndeks öncesi – Ayrıştırılmış veri)

Veri Adı	**Veri Tipi**	**Açıklama**
URL ID	4 Byte (Integer)	Web sayfası kayıt no'su
İçerik Boyutu	4 Byte (Integer)	Web sayfasının içerik boyutu
Kelime ID	4 Byte (Integer)	Kelime no'su. Kelime adedince tekrar eder
Kelime Konum İndisi	2 Byte (Int16)	İndis: Kelimenin web sayfasındaki konumu.
Tür	1 Byte	Tipi: Kelimenin Başlık, Link ve Metin gibi çeşitli tip bilgisi. Kelime konumuna göre tekrar eder

Kelime türleri bir HTML sayfasında geçebilecek olan farklı türleri ifade etmektedir. Bunlar; alan adı, URL adresi, başlık (title), link (anchor), düz metin şeklinde olabilir. Ayrıştırma işlemi eşzamanlı olarak çalışan uygulamamız tarafından yapılmaktadır. Veri derlenmesi ve işlenmesi süreçleri verinin büyüklüğü dikkate alındığında olabildiğince yatay mimari üzerinde olmalıdır. Uygulamalar geliştirilirken bu prensip üzerinde durulmuştur.

3.4. Verilerin İndekslenmesi

İndeksleme işlemi depolama sunucularına kaydedilen her biri farklı klasörde bulunan kelime gruplarının işlenmesi ve her kelime için bir URL listesi oluşturulması esasına göre çalışmaktadır.

İndeksleyici (Indexer) uygulaması, temel olarak metin ve link ayrıştırma işlemleri ile elde edilen veri dosyalarından aynı kodlama ile okuma yaparak veriyi hafızada dizi haline getirir.

İndeksleyici uygulamasının veri dosya okuma alanı 0-254 arası tanımlanmış veri klasörleridir. Her bir klasör okunarak hafızaya yüklenir, işlemi yapılır ve bir sonrakine geçilir. Her bir klasörde belirli bir kelime grubunu bulunmaktadır ve hafızada kelime bazlı bir veri dizi yapısı oluşturulur. Bu işlem her klasör için belirli bir zaman almaktadır. Her indeksleyici uygulaması işlem sırasında müsait olan klasörde çalışır böylece birden fazla indeksleyici uygulaması aynı anda çalışabilir.

İndeksleme işlemi başlarken, sürerken ve bitirildiğinde sunucu uygulamasındaki veri dizilerinde yapılan işlemle ilgili bilgi verilir. İndekslenen klasör ve kelime bilgisi sunucu uygulamasına yollanarak birden fazla indeksleyici uygulamasının aynı kaynağa erişmesi önlenmiş olur.

Uygulama eşzamanlı olarak tüm sunucularda ve klasör bazlı olarak işlemlerini gerçekleştirir. Yapılan işlem neticesinde her kelime için oluşturulan URL listesi sabit bir kodlama ile kaydedilir. Bu kodlama daha sonra arama işlemi sırasında okunacaktır.

Ayrıştırma ve indeksleme işleminin performans açısından en zayıf tarafı bilgisayar dâhili depolama ünitesinin kullanılmasıdır. Bu konu hakkında Sonuç ve Değerlendirme bölümünde detaylı bahsedilmektedir.

Veri ayrıştırma işleminde yapılan kayıt toplu bir şekilde iken indeksleme işlemi sonucunda her bir kelime için bir adet indeks dosyası oluşturulur. Tablo 3.7 indeks dosyası yapısını göstermektedir.

Tablo 3.7. Veri kodlaması (İndeks)

Veri Adı	**Veri tipi**	**Açıklama**
URL ID	4 Byte Integer	Web sayfası kayıt no'su
Kelime Adedi	1 Byte	Sayfada geçen kelime adeti
Kelime Konum Bilgisi	2 Byte (Int16)	Konum: Kelimenin web sayfasındaki konumu Tür: Kelimenin Başlık, Link ve Metin gibi çeşitli tip bilgisi. Kelime adedince tekrar

3.5. Arama İşlemleri

Arama uygulaması (Searcher), arama motorunun arama ve sunum katmanını oluşturmaktadır. Arama, birleştirme ve sıralama işlevlerini yerine getirmektedir.

Arama uygulaması indekslenen veri üzerinde okuma ve sıralama işlemleri yapmaktadır. Sonuçların gösterilmesi esnasında URL'lere ilişkin bilgiler URL ID ile web sitesi veri tabanından alınmaktadır.

Arama işlemi bir kelime veya birden fazla kelime şeklinde ikiye ayrılmaktadır. Tek kelimelik aramalarda ilgili kelimenin Kelime ID'si Sözlük veri tabanından alınır, indekslenen verisi disk ortamından okunur ve kelimenin site içerisinde bulunduğu öneme göre alaka düzeyi derecesi oluşturulur.

Alaka düzeyi önem derecesi kelime tipi ve önem derecesi ile sonuç listesindeki her site için hesaplanır. Alaka düzeyi daha sonra sunum katmanında Pagerank veya SÖD ile birleştirilecektir.

Tablo 3.8. Kelime tipi önem derecesi tablosu

Kelime Tipi	**Önem Derecesi (1- Az Önemli 5- Çok Önemli)**
Alan adı	5
URL Adresi	4
Başlık (title)	3
Link	2
Metin içeriği	1

Birden fazla kelime arama işlemi ise biraz daha karmaşıktır. Çünkü birden fazla kelime ile yapılan aramalarda, kelimelerin her birinin Kelime ID'si alınır ve her kelime için indekslenen veri yüklenir. Yüklenen sonuçlar birbiri ile karşılaştırılıp kesişen kayıtlar alınır ve kesişen kayıtlar için kelimeler arası mesafeler ve kelime tiplerine göre alaka düzeyi derecesi hesaplanır. Alaka düzeyi derecesi IR skoru olarak da literatürde yer almaktadır.

Örneğin “Kocaeli Üniversitesi Bilişim sistemleri Mühendisliği” araması yaptığınızda, bu beş kelimenin geçtiği herhangi web sayfalarını getirip kullanıcıya sunarsanız sonuçlar kullanıcının aradığı şekilde olmayabilir.

Bu nedenle, arama sonucu olarak indekslenen veriler alındığında kelimelerin birbirine olan mesafelerinin puanlanmasına dayalı bir algoritma çalıştırmaktadır. Sonuçlar kelimelerin birbirine olan mesafeyi 0-15 arası 5 farklı konum puanlamasına göre sıralanmaktadır. Böylece alakalı sonuçların üstte çıkması beklenir.

Tablo 3.9. Kelimeler arası mesafe önem tablosu

İki Kelime Arası Mesafe	**Önem Derecesi (1-Az, 5-Çok)**
1	5
2-3	4
4-5	3
6-9	2
10 ve fazlası	1

Arama ara yüzünde kullanıcı dilerse sayfanın önbelleğe kaydedilmiş halini de görebilmektedir. Böylece sayfanın indirildiği zamanki haline erişilebilmektedir.

Arama işlemi neticesinde hesaplanan alaka düzeyi derecesi verileri ile her site için önceden hesaplanmış Pagerank veya SÖD derece verileri birleştirilerek son bir sıralama yapılır ve kullanıcıya sonuçlar sunulur.

Arama yapıldığında tüm arama sonuçları için önbelleğe yükleme işlemi yapılmaktadır. Arama yüküne göre belirli bir zaman olarak ayarlayabileceğimiz bu özellik ile sunucu üzerine her defasında yük oluşturmaktansa daha önceki sonuçlar gösterilmektedir. Şuan için önbellek süresi 2 saat şeklindedir. Bu süre uygulamanın kullanım yüküne göre değiştirilebilir.

Arama algoritması için bir diğer önemli sayılabilecek özellik ise kullanıcı davranışlarının arama sonucuna yansıtılmasıdır. Kullanıcı davranışları sıralamanın veya içeriğin ne kadar doğru olduğunu gösteren yegâne bilgidir. Kullanıcı aradığı

sonuca ulaşmış ise aynı aramayı tekrarlamayabilir veya gelen sonuç listesinden doğru olan sonuca tıklayacaktır.

4. WEB ARAMA SONUÇLARI SIRALAMA ALGORİTMALARI

Arama motorları kullanıcıların yaptığı aramaya göre binlerce sonuç döndürebilmektedir. Vakit sınırı olan kullanıcının tüm sonuçları incelemesi neredeyse imkânsızdır. Yapılan araştırmalara göre, kullanıcıların büyük bir kısmının arama sonuçlarının ilk sayfasını okuyup diğer sayfaları okumadığını göstermektedir.

Arama motorunu kullanan kullanıcıların ilgileri sadece yaptıkları aramanın sayfalarına değil, başka sayfalar da ilgilerini çekmektedir. Arama motorları sıralama algoritmaları mantığına göre sayfalara puanlar vererek değerlendirmek, kullanıcının aradığı ile ilgili sayfalar ve arananla alakalı olabilecek sayfaları listelemektedir [33].

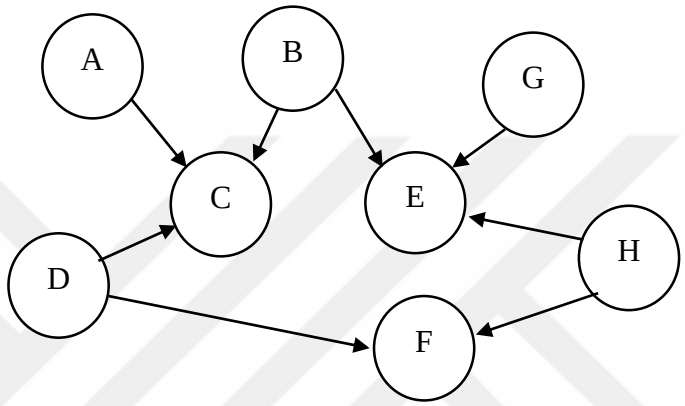
Arama motorlarının değerini artıran önemli bir amacı da, web sayfalarının sonuç sıralamasında hangi sırada yer alacağını belirlemektir. Aranan veriye göre web sayfalarını sıralamak amacıyla PageRank ve HITS başta olmak üzere çeşitli algoritmalar geliştirilmiştir.

Web sayfalarının önemli olanlarının arama motorlarında üst sırada yer alması, popülerliğinin değerlendirilmesi için bağlantı temelli olarak PageRank, HITS ve SALSA gibi çeşitli algoritmaların yanı sıra çalışmamızda SÖD adında yeni bir sıralama algoritma önerilmiştir. Tez çalışması kapsamında bağlantı temelli algoritmalar incelenecektir

4.1. HITS Algoritması

Bağlantı temelli bir algoritma olan HITS algoritması, 1998 yılında Jon Kleinberg tarafından ortaya atılmıştır. Bağlantı dahil metin araması olarak da Türkçe ’ye çevirebileceğimiz HITS algoritması temel de birbirine referans veren metinlerin puanlanması amacıyla ortaya çıkmıştır. Kısaca sayfalar arasındaki bağlantı, arama sonucuna etki edecektir. PageRank algoritması mantığından daha basit kabul edilen HITS algoritması, PageRank algoritmasının ilham kaynağı olarak da anlamlandırılabilir [31].

Bahsi geçen algoritma web sayfalarını derecelendirirken sayfa içeriğine ve de bağlantılarını değerlendirerek sıralar. Algoritma, sayfaları iki farklı küme altında sınıflandırır. Bu kümeler "hub" ve "otorite" olarak adlandırılabilir. HITS algoritması web sayfalarındaki dışarıya bağlantıları ve içeriye bağlantıları kullanarak sayfalarını sıralamaktadır. Bir web sayfası birçok bağlantı ile işaretleniyorsa bu web sayfası otorite sayfa olarak tanımlanmaktadır. Eğer bir web sayfası birçok bağlantıyı işaretliyorsa bu web sayfası iyi bir hub sayfa olarak tanımlanmaktadır.



Şekil 4.1. Otorite ve Hub sayfaları

Şekil 4.1 Hub ve otorite sayfalarını göstermektedir. A,D,B,G,H sayfaları kendinden bağlantı çıkan yani Hub sayfalar, C,E,F ise kendine bağlantı veren yani otorite sayfalar olarak tanımlanmaktadır.

En büyük avantajlarından biri olan arama konusuna göre web sayfalarını sıralayan HITS algoritmasının dezavantajları da mevcuttur. Algoritmanın hesaplanma süresi oldukça fazla olması sebebi ile zaman sıkıntısı mevcuttur. Aynı zamanda PageRank'ta da dezavantaj olan karşılıklı link verme gibi sadece puan artırmaya sebep olan spam linklere çözümü yoktur. Spam linkler sebebiyle bazı sayfalara birikme meydana gelecek ve ne ararsanız arayın birikme olan sayfalar karşınıza çıkacaktır. Başka bir dezavantaj olarak algoritmada konu kayması mevcuttur. Yani her hangi bir web sitesi kendi konusu ile alakalı olmayan herhangi bir web sitesine bağlantı verebilir.

Bu dezavantajları gidermek için çeşitli çalışmalar yapılmıştır. Otorite ve hub hesaplamasını optimize etmek için SALSA algoritması önerilmiştir [32]. SALSA algoritması Pagerank ve HITS Algoritmasından esinlenmiştir.

4.2. PageRank Algoritması

İlk arama motorları temel olarak konu dizinleri ve metinde geçen kelimeleri dikkate alarak arama yapmaktaydı. Aranan konudan bağımsızda olsa metinde aranan kelime ne kadar çok geçiyorsa o sitenin önem derecesini artırmaktaydı. Bu yüzden arama motorlarının aranan veriden ziyade herhangi başka sonuçlar getirerek arama motorlarının yanıltılması oldukça kolaydı.

PageRank algoritması, arama motoru dünyasında ve hatta İnternette bir devrim olarak Google kurulurken oluşturulmuş bir yaklaşımdır. 1998 yılında Sergery Brin ve Page Lawrance tarafından Stanford Üniversitesinde yayınladıkları arama motorları anatomisi hakkındaki çalışmalarında PageRank ifadesi ortaya atılmıştır.

Google'ı 1998 yılında yapılan bu çalışma ile önceki arama motorlarından ayıran ve başarısına etken olan büyük bir faktör PageRank algoritmasıdır [4]. PageRank algoritmasına, rasgele bir İnternet kullanıcısının, herhangi bir siteden başlayarak o sitedeki linkleri takip ederek İnternette gezinmesi temel mantık olmuştur. Kullanıcı eğer ki A sitesinden bir linke tıklayıp B sitesini ziyaret ettiyse, B sitesi A sitesine link verdiği için önem derecesi artmıştır.

PageRank mantığına örnek verilirse; nasıl ki akademik bir çalışma yaptığımızda başka yapılan akademik çalışmalara atıf yapıyorsak, atıf yaptığımız kaynağın alanında önemli bir kaynak ve çalışma olduğunu belirtmiş oluyoruz demektir. Aynı şekilde başka çalışmalar da bizim çalışmalarımızı kaynak olarak gösteriyorsa bu durumda bizim çalışmamızın değerini artırmaktadır. Tabi ki başka bir önemli noktada bizi kimin kaynak gösterdiğidir. Yani onun önem derecesi ne kadardır. O ölçüde önem derecesini artıracaktır. Web siteleri açısından bakıldığında, herhangi 100 sitenin bir siteye link vermesinden daha çok PageRank değeri yüksek olan önemli, bilinen bir sitenin link vermesi daha önemlidir.

Kısacası PageRank internette web sayfalarının birbirine olan bağlantı haritasını kullanarak hesaplanır. Google'da yer alan PageRank değeri hesaplanırken üç durumu değerlendirir. Bunlar;

- Tüm sayfaların ayrı ayrı PageRank değeri,

- Sayfaya gelen bağlantıların miktarı ve kalitesi,
- Her sayfadan giden bağlantıların sayısı

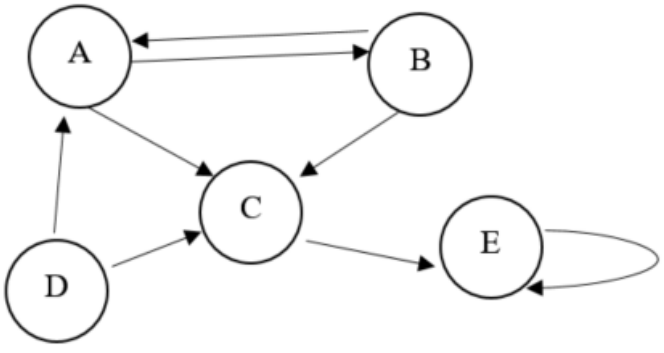
Bir A web sitesi için PageRank değeri (4.1) eşitliğinde verilen formül ile hesaplanır.

$$PR(\mathrm{A}) = (1-d) + d\left(\frac{PR(\mathrm{B})}{L(\mathrm{B})} + \frac{PR(C)}{L(\mathrm{C})} + \frac{PR(\mathrm{D})}{L(\mathrm{D})} + \cdots\right) \quad (4.1)$$

PR(A) A sitesinin PageRank değeri, *d* parametresi 0 ile 1 arasında değişen sönümleme faktörü, *PR*(B) B sitesinin PageRank değeri, *L*(B) B sitesinin bağlantı çıkışı sayısını ifade etmektedir. *PR*(C), *PR*(D), *L*(C) ve *L*(D) ifadeleri de aynı şekilde PageRank değeri ve bağlantı çıkış sayılarını vermektedir. Sönümleme faktörü 0.85 değerine ayarlandığı bilinmektedir. Basit yinelemeli bir algoritma ile hesaplanabilen PageRank değerlerinin toplamı 1'e eşit olur. (4.1) eşitliğine göre A sitesinin PageRank değeri kendisine link veren B, C, D, .. sitelerinin PageRank değerlerine bağlıdır [4]. Google 2016'da Araç Çubuğu PageRank desteğini kaldırmış olmasına rağmen Google tarafından geliştirildiği ve şu anda çok farklı bir sürümün kullandığı tahmin edilmektedir [33].

$$PR(\mathrm{A}) = \left(\frac{1-d}{N}\right) + d\left(\frac{PR(\mathrm{B})}{L(\mathrm{B})} + \frac{PR(C)}{L(\mathrm{C})} + \frac{PR(\mathrm{D})}{L(\mathrm{D})} + \cdots\right) \quad (4.2)$$

(4.2) eşitliğinde verilen PageRank formülü (4.1) eşitliğinde verilen formüle benzer terimlerden oluşmaktadır. (4.2) eşitliğinin tek farkı $\frac{1-d}{N}$ ifadesidir. N terimi PageRank hesaplanan toplam web sitesi sayısını ifade etmektedir. (4.1) eşitliği Google kurucuları S.Brin ve L.Page tarafından duyurulan formül olsa da (4.2) eşitliğinin PageRank hesaplamasında kullanılan asıl formül olduğu Google çalışanları tarafından ifade edilmektedir [34].



Şekil 4.2. Pagerank örneği – bağlantı durum şeması

Şekil 4.2 belli bir İnternet sayfası için bağlantı haritası olarak gösterilebilir. Sayfalar A,B,C,D,E ile gösterilen daireler, sayfalar arası bağlantılar ise ok işareti ile gösterilen kenarlardır. Yukarıdaki şekilde A ve C sayfaları için D geriye bağlantılıdır. Aynı şekilde C sayfası için A, B ve D sayfaları için geriye bağlantılıdır. Rasgele sayfalar arasında gezinen bir kullanıcı B sayfasında ise, A ve C sayfalarını seçme olasılığı 1/2 diğer sayfaları seçme olasılığı ve B sayfasında kalma olasılığı 0'a eşittir. E sayfası sadece kendine link vermiştir. Bu şekilde sadece kendinden başka sayfaya bağlantı vermeyen sayfalar örümcek tuzağı (veya tarayıcı tuzağı) olarak adlandırılmaktadır.

Şekil 4.2'de verilen bağlantı durum şemasının matrisi (4.3) eşitliği ile gösterilebilir.

$$M=\begin{bmatrix} 0 & 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (4.3)$$

M_{ij}= 1/k olarak kabul edilirse, i sayfasından j sayfasına yönlendirilen bağlantıyı ifade etmektedir. k parametresi her i sayfasından çıkan dış bağlantıların toplam sayısıdır. Bu bağlantılardan birisi "j" sayfaya verilir. Matristeki her bir satır i sayfasından çıkan dış linkleri, her bir sütun i sayfasına gelen iç linkleri gösterir. Örneğin, matriste M_{AB}=1/2 olduğunu görmekteyiz. A sayfasından 2 çıkış olduğunu ve bu çıkışlardan birisinin B sayfasına gittiğini anlayabiliriz. Eğer ki M_{ij}= 0 ise i sayfasından j sayfasına bağlantı verilmemiş demektir.

M matrisinin her bir sütununda oluşan değerlerin toplamı 1'e eşit olmalıdır. Sütundaki her hangi bir değer PageRank değerini vermektedir.

4.2.1. PageRank algoritmasındaki problemler

PageRank algoritmasının hesaplanması ve uygulanmasında başlıca iki problem mevcuttur. Bu problemlerden ilki sıra sızıntısı (rank leakage), ikincisi ise sıra kapanı (rank sink) olarak adlandırılmaktadır.

Sıra sızıntısı problemi, başka sayfalara bağlantı vermeyen web sayfalarının aldıkları PageRank değerini diğer sayfalara aktarmadıkları durumda kendileri üzerinde PageRank değeri birikmesi olarak ifade edilir. Bu durumun çözümü olarak PageRank

algoritmasının uygulanması sırasında sıra sızıntısı problemine neden olan sayfaların, hesaplama kümesinden çıkarılarak hesaplamanın sağlıklı bir şekilde yapılması veya ilgili sayfalardan hesaplama kümesindeki tüm sayfalara sanal birer bağlantı eklenmesi önerilmiştir [35].

Sıra kapanı problemi ise kendileri arasında bağlantı sağlayarak kapalı gruplar oluşturan sayfaların, diğer sayfalardan bağlantı aldıkları halde dışarıya bağlantı sağlamamaları nedeniyle sistemdeki PageRank değerinin kendi grupları üzerinde birikmesine yol açmalarıdır. Bu problemin algılanması ve çözümü için PageRank formülünün değiştirilerek güçlendirilmesi gerekmektedir [35].

4.2.2. PageRank algoritmasının dezavantajları

1998 yılında PageRank algoritmasının sunulmasıyla beraber sürekli geliştirilerek, ek koruma algoritmaları ile sağlamlaştırılmıştır. Bazı İnternet sayfaları arama sonuçlarında PageRank değerlerini fazla çıkartıp üst sıralarda yer almak için, kendi sayfalarına çok sayıda bağlantı veren sayfalar oluşturabilir, başka sayfalar ile anlaşarak karşılıklı bağlantı vererek algoritmayı manipüle edebilirler. Bu ve benzeri şekil de algoritmayı yanıltacak şekilde kötüye kullanımın yaygınlaşması sebebiyle 2016 yılından itibaren Google sayfaların PageRank değerlerini herkese açık olarak yayınlamaktan vazgeçmiştir [33].

Bazı PageRank algoritması kötüye kullanımı şunlardır

- Karşılıklı bağlantılar alma.
- Komut dosyaları ile ya da ödül yöntemleriyle bağlantı elde etme.
- Yakın çevre desteğiyle bağlantılar sağlama.
- Ücretsiz veya ücretli sayfa linkleri sağlayan web sayfaları.

Google PageRank değeri üzerindeki kötü kullanımları belirlemek ve ortadan kaldırmak amacı ile çok fazla farklı algoritmayı ve çözümü de devreye almıştır. Uygulanan çözümler PageRank ile birlikte sıralamanın belirleyici unsurlarıdır. Tüm bu manipülasyonlara rağmen arama motoru optimizasyonu ile ilgilenen uzmanların PageRank algoritmasının öneminin ve etkinliğinin hala çok değerli olduğu hakkında fikir değişikliği yoktur.

Bağlantı temelli algoritmaların temel mantığı olan bağlantı verilen siteye önem derecesi aktarımı bazı sitelerde derece birikmesine yol açmaktadır. Örneğin önem derecesi yüksek bir site başka bir siteye bağlantı vermişse önem derecesini bağlantı sayısı kadar aktardığından bu sayı bir ise önem derecesi olduğu gibi aktarılmaktadır.

4.3. Önerilen Site Önem Derecesi Algoritması

Site Önem Derecesi Algoritması, bağlantı temelli bir sıralama algoritmasıdır. PageRank algoritmasına benzer bir şekilde bağlantı sağlanan web sitesinin bağlantıları üzerinden hesaplanır. PageRank'ta bulunan sönümleme faktörü site önem derecesi algoritmasında bulunmamaktadır. Basitçe; bir A web sayfasının SÖD değeri, kendisine bağlantı sağlayan diğer web sayfalarının bağlantı çıkış sayılarının bir fazlalarına bölümünün toplamı olarak ifade edilir.

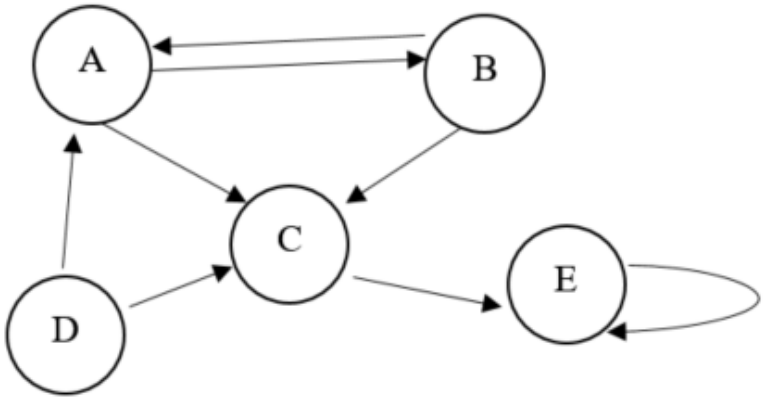
Bir A web sitesi için Site Önem Derecesi değeri (4.4) eşitliğinde verilen formül ile hesaplanır.

$$S\ddot{O}D(\mathrm{A}) = \frac{1}{N} + \left(\frac{S\ddot{O}D(\mathrm{B})}{B\acute{C}S(\mathrm{B})+1} + \frac{S\ddot{O}D(\mathrm{C})}{B\acute{C}S(\mathrm{C})+1} + \frac{S\ddot{O}D(\mathrm{D})}{B\acute{C}S(\mathrm{D})+1} + \cdots\right) \tag{4.4}$$

SÖD(A) A sitesinin önem derecesi, *N* toplam site sayısı, *SÖD*(B) B sitesinin önem derecesi, *BÇS*(B) B sitesinin bağlantı çıkış sayısını ifade etmektedir. *SÖD*(C), *SÖD*(D), *BÇS*(C) ve *BÇS*(D) ifadeleri de aynı şekilde site önem derecesi ve bağlantı çıkış sayılarını ifade etmektedir. Site önem dereceleri yinelemeli olarak hesaplanmaktadır. Her bir site için bu siteye link veren sitelerin önem dereceleri ile bağlantı çıkış sayıları bir arttırılarak bölünür. Site önem derecesinin bir kez hesaplandıktan sonra bağlantı haritası değişimlerinde güncellenmesi gerekmektedir.

Bağlantı çıkış sayısının bir artırılmasının nedeni, önem derecesi atanmış bir sitenin sadece bir bağlantı çıkışı olduğunda önem derecesi kendisinden diğerine tam katsayıyla aktarmasının bazı sitelerde önem derecesi birikimine yol açmasını engellemektir. Değerli bir sitenin bağlantı paylaşımı yoluyla önem derecesinin tamamını transfer etmesi yerine bağlantı çıkışına göre bir katının kendisine kalması daha gerçekçi olmaktadır.

$\frac{1}{N}$ terimini kullanmamızın nedeni, site önem derecesi hesaplamasına başlarken hesaplama yapılacak sayfaların önem derecelerinin sıfıra eşit olması ve böylece her sayfa için bir başlangıç değeri sağlanmasıdır.



Şekil 4.3. Site Önem Derecesi bağlantı haritası

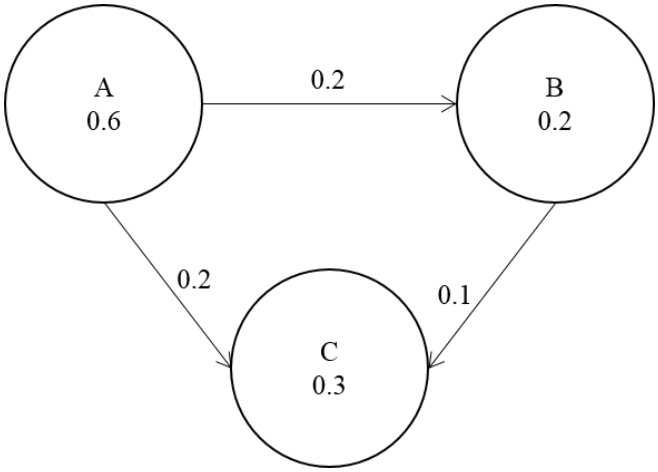
Şekil 4.3 bağlantı haritasını göstermektedir. Önerilen yaklaşımda belirli sayıda bağlantı verilmişken her bağlantı sayısına bir eklenmiştir. Örneğin; C sitesinin SÖD değeri hesaplanırken bağlantı veren siteler 1/(k+1) olarak önem derecelerini aktarmışlardır. Böylece, PageRank'ta meydana gelen rank birikmesi sorununu aza indirilmiş olur.

Şekil 4.3'de verilen bağlantı haritasının SÖD yaklaşımı matrisi (4.5) eşitliği ile verilir.

$$M=\begin{bmatrix} 0 & 1/3 & 0 & 1/3 & 0 \\ 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 \end{bmatrix} \quad (4.5)$$

M_{ij}= 1/(k+1) olarak değerlendirirsek, "j" sayfasından çıkan "k" adet bağlantı vardır ve bunlardan birisi "i" sayfaya verilmiş demektir. Örneğin (4.5) deki matriste M_{ab}=1/3 olduğunu görmekteyiz. B sayfasından 2 çıkış olduğunu ve sayıya 1 eklenerek 3'e bölündüğünü ve bu çıkışlardan birisinin A sayfasına gittiğini anlayabiliriz. Eğer ki M_{ij}= 0 ise j sayfasından i sayfasına bağlantı verilmemiş demektir.

Başka bir örnekte ise;

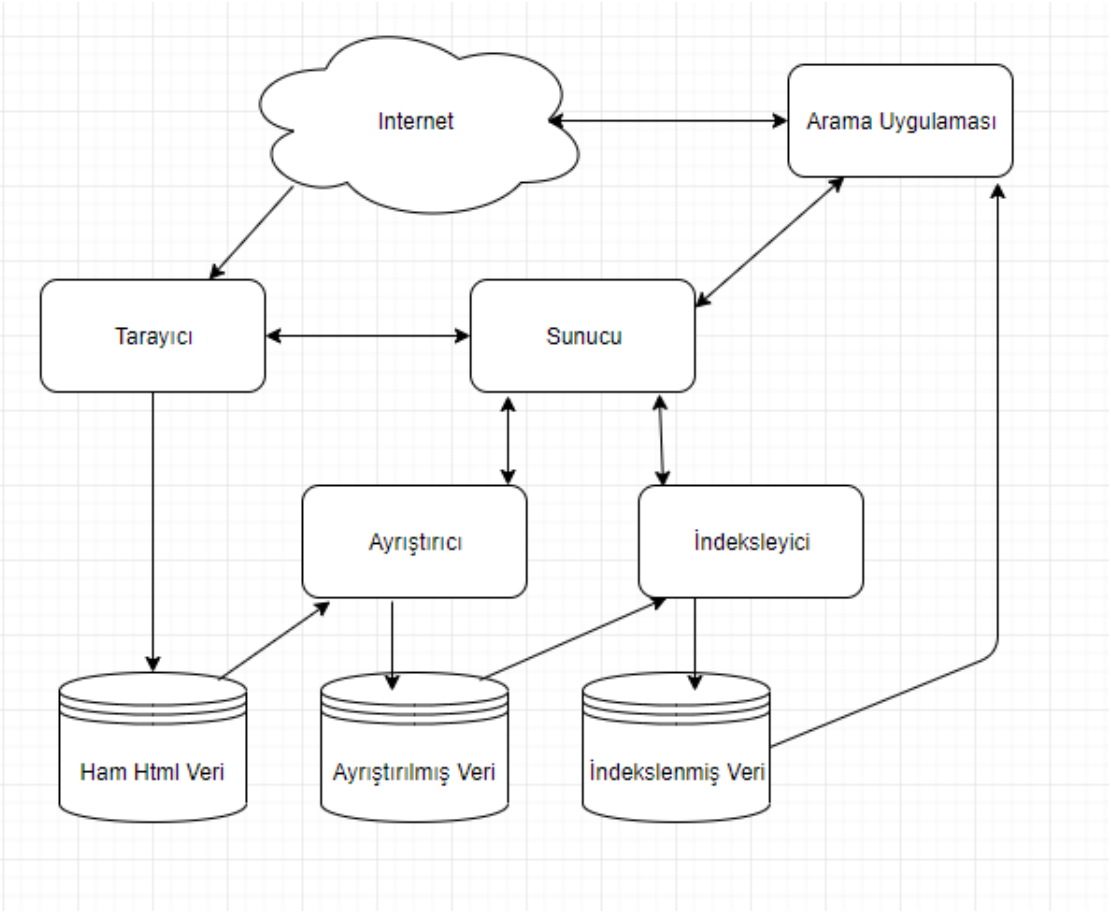


Şekil 4.4. Örnek bağlantı şeması (Site Önem Derecesi)

Şekil 4.4'de verilen bağlantı şemasında A sitesinin önem derecesi 0,6 iken B ve C sitesine bağlantı verdiğinde PageRank algoritmasına göre 0,6/2=0,3 değeri verilirken, SÖD algoritması yaklaşımı ile 0,6/3=0,2 puan aktarım gerçekleşmektedir. Aynı şekilde B sitesi de C sitesine bağlantı sağlamaktadır. B sitesinin verdiği bağlantı neticesinde önem derecesinin tamamı yerine 0,2/2=0,1 puan aktarım gerçekleşmektedir.

5. ARAMA MOTORU UYGULAMASI

5.1. Uygulama Mimarisi



Şekil 5.1. Mimari genel görünümü

Şekil 5.1 arama motoru mimarisini göstermektedir. Arama motoru mimarisi tarama, ayrıştırma, indeksleme ve arama olmak üzere 4 ana işlevden oluşur.

Veri deposu henüz boş iken tarama kuyruğuna herhangi bir web sayfası eklenir. Bu web sayfası başka sayfalara bağlantı sağlamalıdır. Tarama buradan başlayarak tüm bağlantılar ve onların bağlantılarına ulaşacaktır.

Tarayıcı uygulaması kuyruktan bir web sayfası alır. Bu sayfayı indirerek HTML veri olarak veri deposuna yükler. Artık bu web sayfası taranmıştır.

Ayrıştırıcı uygulaması indirilen web sayfasını okuyarak html kodlarını ayırıp salt metini veri dizisine çevirerek veri deposuna kayıt eder. Bu sırada sayfadaki bağlantıları tarama kuyruğuna ekler.

İndeksleyici uygulaması ayrıştırılarak veri dizisi haline getirilmiş veriyi kelime bazlı olarak sınıflandırarak kaydeder. Arama uygulaması indekslenmiş veriden sonuçlar üretir.

5.2. Uygulamanın Çalıştırılması ve İşleyişi

Geliştirilen uygulama .net core ile visual studio ortamında katmanlı mimari şeklinde hazırlanmış, IIS üzerinden local olarak çalıştırılmaktadır. Uygulama bir TCP server olmak üzere üç tane de Linux işletim sistemine sahip bilgisayar üzerinde çalıştırılarak "edu.tr" uzantılı web siteleri taranmıştır. Yaklaşık 900 bin sayfa kayıt altına alınmış olup yaklaşık üç milyon benzersiz kelime indekslenmiştir.

İlk olarak tarayıcı uygulaması çalıştırılmalı ve internette gezinme başlatılmalıdır. Arama motoru birçok web sayfasından veri yükleyerek kullanılabilir duruma geldiğinde dahi tarayıcı uygulaması çalıştırılarak yeni web sayfalarının indirilmesi ve ayrıştırma, indeksleme süreçlerine dâhil edilmesi gereklidir.

Tez kapsamında tarayıcı uygulaması henüz indirilmemiş web sayfalarını indirmeye devam edecektir. Eski sayfaların güncellenmesi konusunda uygulama bir konfigürasyon dosyası ile ayarlanabilir parametrik hale getirilebilir. Tarayıcı uygulaması arama motorunun veri girdisi işlevini devamlı suretle yürütmelidir.

Tarayıcı tarafından indirilen veriler ham halde kaydedilir. Arama sonuçlarına yansıması için ham veri üzerinde hazırlık işlemi yapılmalıdır. Ham veri html ve metin ile birlikte karışık halde bulunur. Ham veri web sayfasının içeriğinin bir kopyasıdır ve saklanır. Saklanan kopya veri üzerinde yapılan işlemlerin ardından gerektiğinde iyileştirme kapsamında tekrar aynı işlemlere tabi tutulabilir. Ayrıca kayıtlı bulunan kopya arama sonuçları sunumunda önbellek hali olarak kullanıcıya sunulmaktadır.

Verinin arama için hazırlanması ayrıştırma ve indeksleme işlemleri ile sağlanır. Ayrıştırma işlemi her yeni eklenen ham veri için yapılmalıdır. Ayrıştırma sonucunda indeksleme işlemine girdi sağlayacak salt metin verisi elde edilir. Elde edilen salt

metin verisi hafızaya yüklenme, işlenme yükü nedeniyle kategorize edilmiş halde ikilik düzende kodlanarak saklanır.

Ayrıştırma işlemi sonrası veri indeks işlemine hazır hale gelir. Ayrıştırma uygulaması çalıştığı sırada henüz işleme tabi tutulmamış ham veriyi sunucu üzerinden sorgulayarak bu veri üzerinde çalışır. Sunucu uygulaması çalıştığı sürece çalışabilmektedir.

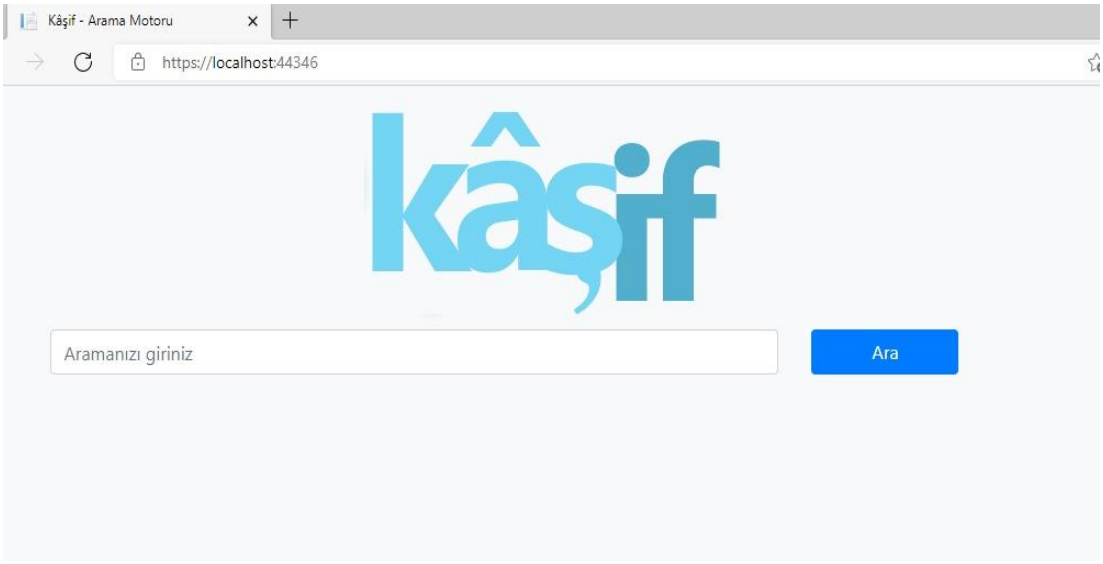
Ayrıştırma işlemine tabi tutulup elde edilen veri indeksleme işlemi için girdi niteliğindedir. İkilik düzende düzenli halde kodlanmıştır. İndeksleyici uygulama ayrıştırma sonrası elde edilen veri üzerinde çalışır. Bu veriyi hafızaya yüklenir ve çeşitli algoritmalar ile kategorize edilir. İndeksleme işlemi basitçe hangi kelimenin hangi web sayfalarında geçtiğinin ortaya çıkarılmasıdır. Bu yapılırken sitedeki konum ve tip bilgisi de bu çıktıya dahil edilir.

Konum ve tip bilgisi arama sonuçlarının sıralanması ve doğru sonuçların üretilmesi adına önemlidir. İndeksleme işlemi neticesinde veri ikilik kodlanarak farklı bir ortama yazılır.

Tarayıcı uygulaması sürekli çalışırken ayrıştırma ve indeksleme işlemi yeni ham veri geldikçe tekrarlanır. Tarayıcı, ayrıştırıcı ve indeksleyici uygulamalar kapalı olsa bile arama uygulaması önceden indekslenmiş verinin sunumunu gerçekleştirebilir. Çünkü indeks verisi mantıksal olarak farklı bir alanda tutulmaktadır. Erişmek için arama uygulamasının disk hiyerarşisinde o alanı görmesi yeterlidir.

İndekslenmiş veri değiştirilmeden arama motoru çalışmaya devam edebilir ancak bu suretle veri eski kalacaktır ve güncelliğini yitirecektir. Bu yüzden güncellenme ve yeni verilerin alınıp eklenmesi önemli bir hal almaktadır.

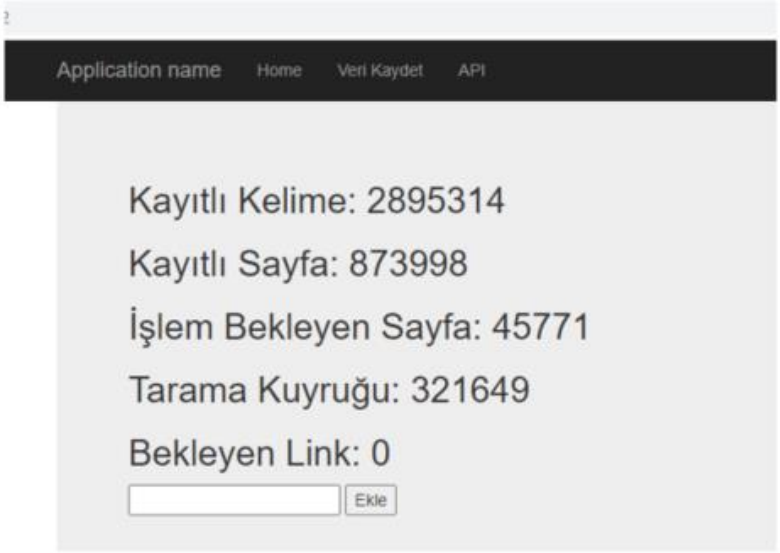
Veri indirme, ayrıştırma ve indeksleme işlemlerinin sonucunda elde edilen arama alt yapısı arama uygulaması ile kullanılır. Arama uygulaması girilen arama terimini indekslenmiş veri havuzunda arayarak sonuçların sıralanması ve gösterilmesini sağlar. Şekil 5.2’de geliştirilen arama uygulaması ekranı verilmiştir.



Şekil 5.2. Kâşif Arama Motoru

5.3. Arama Sonuçlarının Kıyaslanması

Uygulama veri kümesi "edu.tr" Web sitelerini kapsamaktadır. Tez çalışmasında, Şekil 5.3'de gösterildiği gibi 873998 Web sayfası indirme, ayrıştırma ve indeksleme işlemine tabi tutulmuştur. Arama motorunun çalışmaya devam etmesi durumunda bu sayı artacaktır.



Şekil 5.3. Veri işleme durumunu gösterir ekran

Tablo 5.1 Google tarafından kullanılan PageRank ve önerilen Site Önem Derecesi algoritmalarının karşılaştırması vermektedir.

Tablo 5.1. PageRank ve Site Önem Derecesi algoritmaları istatistik

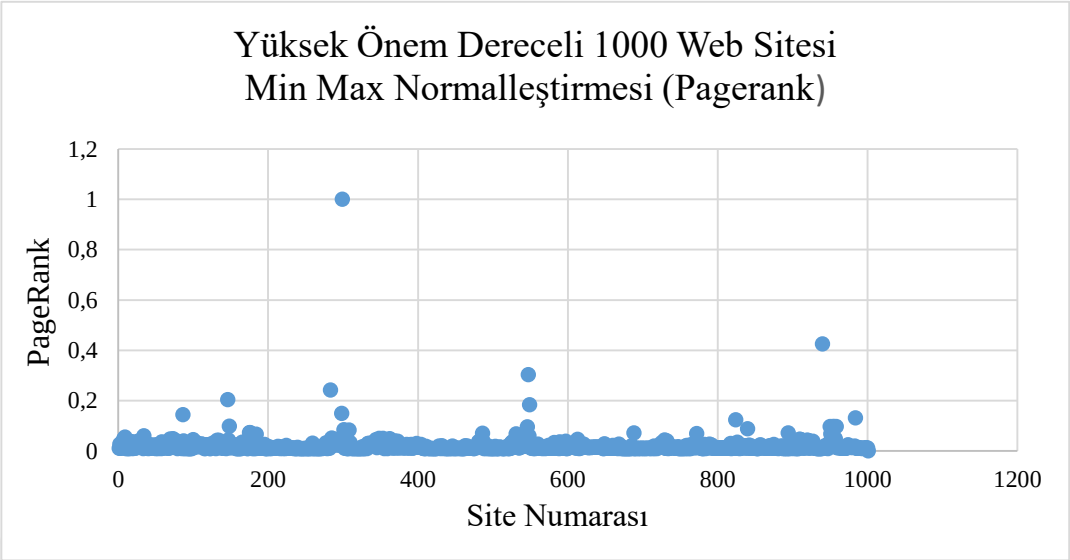
	SÖD yaklaşımı	**PageRank**
Toplam Değer	3,9579	386379,6394
Link Sayısı	29575025	29575025
En Yüksek Değer	0,027877763	3033,39993
En Düşük Değer	0,0000011442	0,15

Tablo 5.1'de verilen Hesaplama Süresi 29575025 link ve 873998 sayfa için PageRank ve SÖD algoritmalarının çalışma süresini, Toplam Değer 873998 sayfaya ait önem derecelerinin toplamını, Link Sayısı 873998 sayfanın birbirine sağlamış oldukları bağlantı sayısını, En Yüksek Değer 873998 sayfanın önem derecelerinden en yüksek önem derecesini, En Düşük Değer 873998 sayfanın önem derecelerinden en düşük önem derecesini ifade etmektedir.

Şekil 5.4. Site Önem Derecesi hesaplama uygulaması

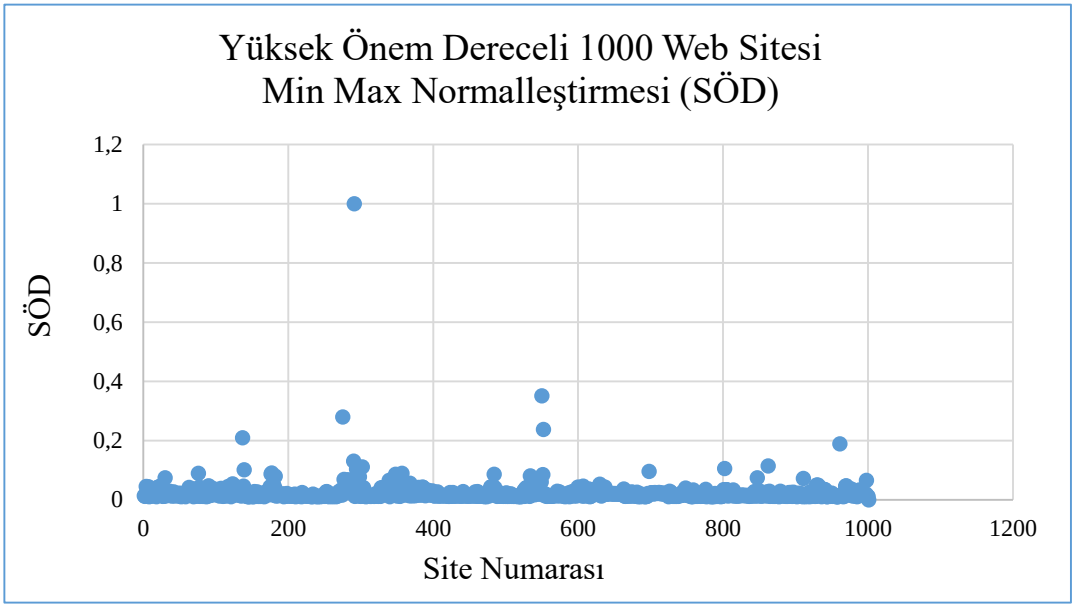
Şekil 5.4 ekran görüntüsü verilen uygulama PageRank ve SÖD hesaplama algoritmalarını gerçeklemek için geliştirilen uygulamayı göstermektedir.

Şekil 5.5 PageRank algoritması sonuçlarını göstermektedir. En yüksek normalleşme değerleri sırasıyla 1 ve 0,42 olarak elde edilmiştir. Büyük çoğunlukla normalleşme değerleri 0,2'nin altında gerçekleşmiştir. Önerilen SÖD yaklaşımında 2. sıradaki web sayfasının değeri 0,35 iken PageRank algoritmasında 0,42'dir.



Şekil 5.5. PageRank Min-Max Normalleşmesi

Şekil 5.6 önerilen SÖD eşitliği ile gerçekleştirilen uygulama sonuçlarını göstermektedir. Yüksek önem derecesi alan 1000 Web sitesinin önem derecelerinin min max normalleşmesi ile dağılımı incelenmiştir. En yüksek normalleşme değerleri sırasıyla 1 ve 0,35 olarak elde edilmiştir. Büyük çoğunlukla normalleşme değerleri 0,2 değerinin altında gerçekleşmiştir.

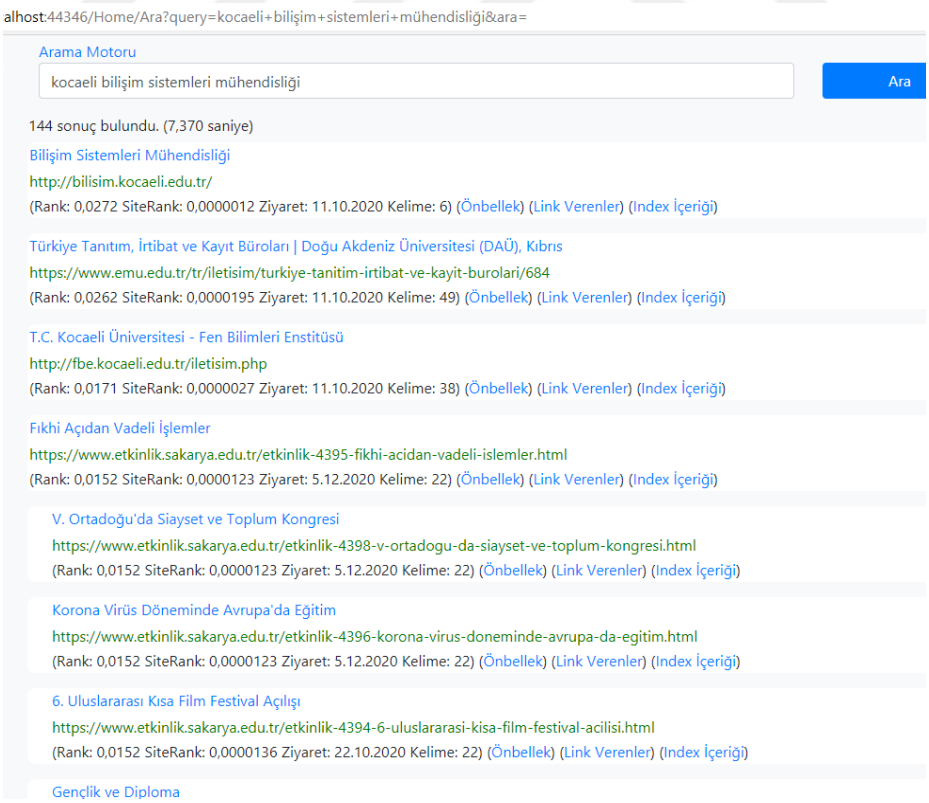


Şekil 5.6. SÖD Min-Max Normalleşmesi

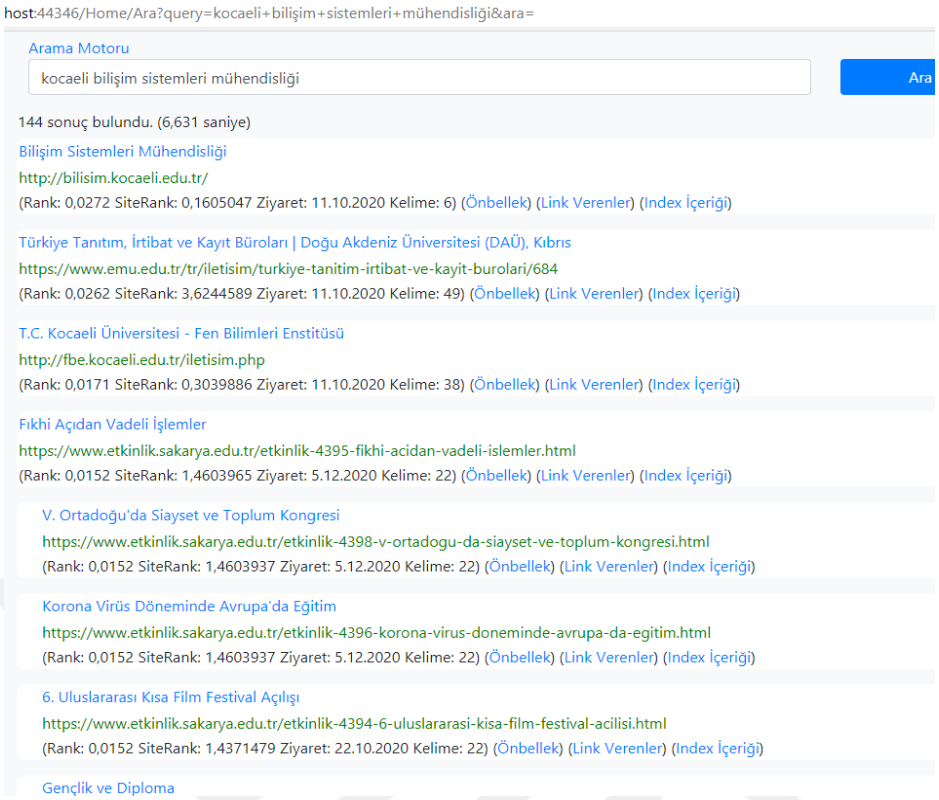
Geliştirdiğimiz prototip arama motoru uygulamasında örnek aramalar PageRank ve SÖD algoritmaları ile ayrı ayrı gerçekleştirerek alınan sonuçlara aşağıda yer verilmektedir.

Şekil 5.7 ve 5.8 sırasıyla PageRank ve önerilen SÖD için "Kocaeli bilişim sistemleri mühendisliği" arama sonuçlarını göstermektedir. Yapılan arama işleminde her iki algoritmanın benzer sıralama sonuçları verdiği görülmüştür.

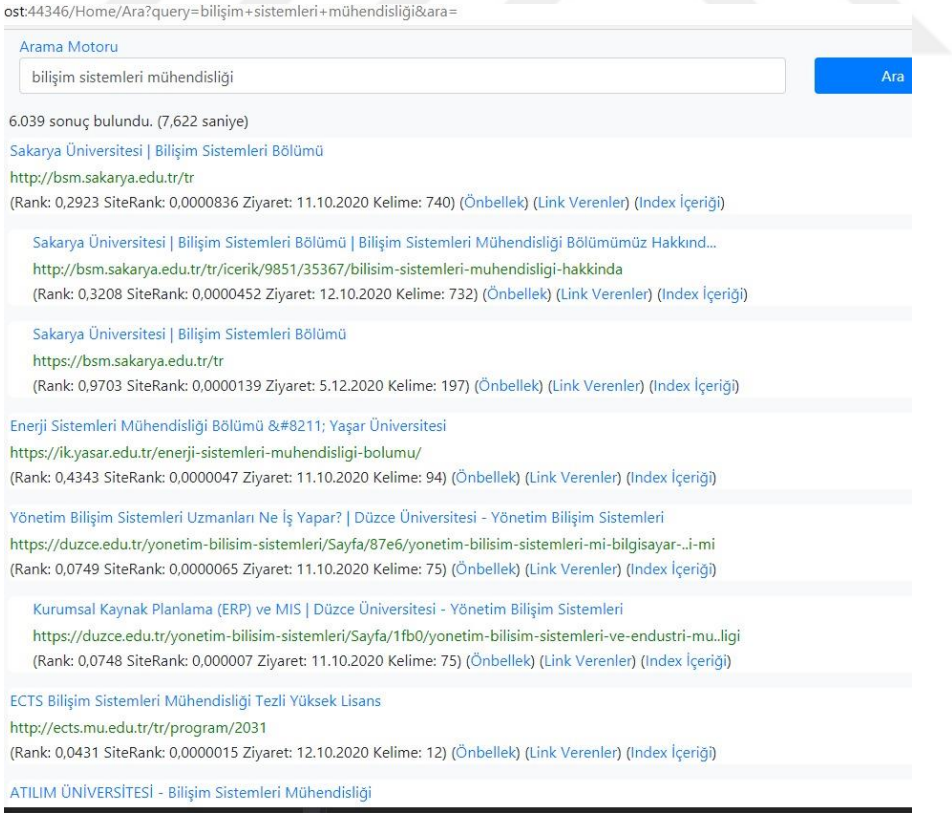
Şekil 5.9 ve 5.10 sırasıyla PageRank ve önerilen SÖD için "bilişim sistemleri mühendisliği" arama sonuçlarını göstermektedir. Yapılan arama işleminde her iki algoritmanın birbirine çok yakın ancak farklı arama sonuçları verdiği görülmüştür. Şekil 5.9'da PageRank algoritması ile alınan sonuçlarda "Enerji Sistemleri Mühendisliği" başlıklı sonuç dikkat çekmektedir. Aynı sayfanın şekil 5.10'da verilen SÖD algoritması sonuçlarında bulunmadığı görülmektedir. "bilişim sistemleri mühendisliği" aramasında SÖD algoritmasındaki sonuçların daha tutarlı olduğu anlaşılmaktadır.



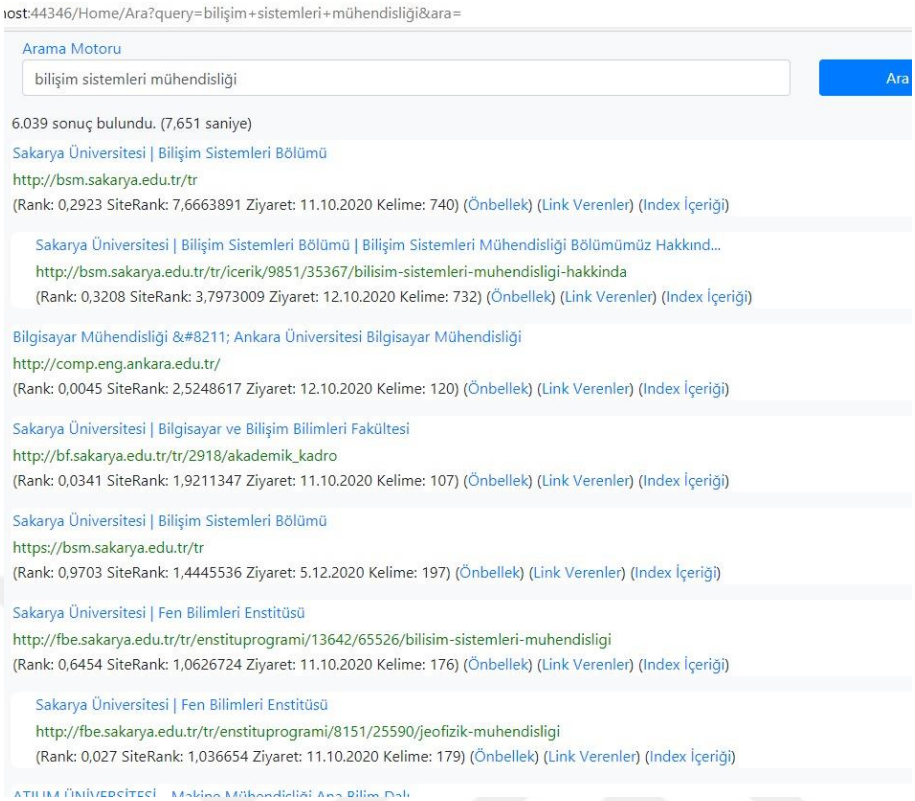
Şekil 5.7. "Kocaeli bilişim sistemleri mühendisliği" arama sonuçları (PageRank)



Şekil 5.8. "Kocaeli bilişim sistemleri mühendisliği" arama sonuçları (SÖD)



Şekil 5.9. "Bilişim sistemleri mühendisliği" arama sonuçları (PageRank)



Şekil 5.10. “Bilişim sistemleri mühendisliği” arama sonuçları (SÖD)

6. SONUÇ

Uygulamaların çalıştırılması sırasında arama motoru işlevleri geliştirilirken birçok zorluk ile karşılaşılmıştır. Başlıca zorluk işlemlerin çalışma sürelerinin uzunluğudur. Çalışma süresinin kısaltılması için eş zamanlı çalışabilecek işlevler ayrıştırılmıştır. Bu durum planlama ve kodlama zamanının artmasına neden olmuştur. Bir diğer zorluk ise disk yazma/okuma işlemlerinin zaman almasıdır. Bu problemlere en iyi çözüm olarak işlemleri ve veri depolarını fiziki olarak birbirinden ayırıp yatay bir mimari kurmak olmuştur. Ayrıca "edu.tr" adresli web sayfalarının indirme işlemleri sırasında İnternet servis sağlayıcıların güvenlik engellemeleri ile karşılaşıldığından indirme işlemlerinin yavaşladığı görülmüştür.

Tez çalışmasında geliştirilen prototip arama motoru uygulaması ile PageRank algoritması ve SÖD algoritmasının arama sonuçlarına etkisi incelenmiştir. Yapılan uygulama sonuçlarından elde edilen verilerden bağlantı temelli algoritmalar kullanımının arama sonuçlarının sıralanmasında etkin olduğu gözlemlenmiştir. SÖD algoritması bağlantı sayısı az bulunan sitelerin önem derecesi aktarımını genel dağılım açısından daha adil hale getirip sıralamaya olumlu katkı sağlamaktadır. Önerilen yaklaşım, PageRank algoritmasına kıyasla sayfa önem derecesi birikmesi problemini önemli ölçüde azaltmaktadır. Bu şekilde, aranan kelime ile ilgili alaka düzeyi yüksek sonuçların gösterilmesi sağlanmış olmaktadır.

Prototip arama motoru uygulaması, yapısında kullanıcı alışkanlığı geri dönüş bilgileri yer almamaktadır. Kullanıcı bilgileri arama sonuçlarının iyileştirilmesinde oldukça önemlidir. Gelecek çalışmalarda bu bilgilere de yer verilmesi arama sonuçlarının doğru sıralanmasını sağlayacaktır.

KAYNAKLAR

[1] World Internet Users and 2021 Population Stats, https://www.internetworldstats.com/stats.htm, (Ziyaret Tarihi: 11 Mayıs 2021).

[2] Yang G., Yang X., Hao W., Design and implementation of a professional search engine, *7th International Conference on Modelling, Identification and Control (ICMIC)*, 2015, 1-6, DOI: 10.1109/ICMIC.2015.7409452.

[3] Search Engine Market Share Worldwide (Apr 2020-Apr 2021), https://gs.statcounter.com/search-engine-market-share, (Ziyaret Tarihi: 11 Mayıs 2021).

[4] Brin S., ve Page L., The Anatomy of a Large-Scale Hypertextual Web Search Engine, *Computer Networks and ISDN Systems*, 1998, **30**(1-7), 107-117.

[5] Aravindhan R., Shanmugalakshmi R., Comparative analysis of Web 3.0 search engines: A survey report, *International Conference on Advanced Computing and Communication Systems*, 2013, 1-6, DOI: 10.1109/ICACCS.2013.6938715.

[6] Brin S., Page L., Motwami R., Winogard T., The PageRank Aitation Ranking: Bringing Order to The Web, *Stanford University*, SIDL-WP-1999-0120, 1-17, 1999.

[7] Işık M., Arama Motorları Mimarisi, Web Sayfalarının İçerik Skoru ve Google Pagerank Formülünün İncelenmesi, Yüksek Lisans Tezi, Kadir Has Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2013, 333150.

[8] Kausar, Md Abu, V. S. Dhaka, and Sanjeev Kumar Singh. "Web crawler: a review." International Journal of Computer Applications, 2013, **63**(2).

[9] Boldi, Paolo, et al. "Ubicrawler: A scalable fully distributed web crawler." Software: Practice and Experience, 2004, **34**(8), 711-726.

[10] Karlık M., Arama Motoru Mimarisi ve Uygulaması, Yüksek Lisans Tezi, Konya Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Konya, 2018, 531394.

[11] Razbonyalı C., Dikey Arama Motorlarının İncelenmesi ve Dikey Arama Motoru Uygulaması, Yüksek Lisans Tez, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Edirne, 2011, 300178.

[12] Pembe F.C., Güngör T., Structure-Preserving and Query-Biased Document Summarisation for Web Searching, *Online Information Review*, 2009, **33**(4), 696-719.

[13] Baker M.R.B., Web Tarama Robotu ve Sonuç Sıralama Algoritmasının Geliştirilmesi, Doktora Tez, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, 2017, 479824.

[14] Veglis A., Giomelakis D., *Search Engine Optimization*, Printed Edition of the Special Issue Published in Future Internet, MDPI, Basel, Switzerland, 2021.

[15] Yalçın N., Köse U., What is Search Engine Optimization: SEO?, *Procedia Soc. Behav. Sci.*, 2010, 9, 487–493.

[16] Vuran E.G., Arama Motoru Optimizasyonu, Yüksek Lisans Tez, Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, 2019, 607036.

[17] Türkiye'deki Arama Motoru Kullanım Oranları, https://novasta.com.tr/turkiyede-ki-arama-motoru-kullanim-oranlari/, (Ziyaret Tarihi: 11 Mayıs 2021).

[18] İnternet Erişim Modeli, https://ca.wikipedia.org/wiki/Fitxer:Internet_Connectivity_Access_layer.svg, (Ziyaret Tarihi: 11 Mayıs 2021).

[19] İnternetin Tarihi, https://tr.wikipedia.org/wiki/internetin_tarihi , (Ziyaret Tarihi: 07 Mayıs 2021)

[20] Shapiro, Stuart., "Turing's Legacy: A History of Computing at the National Physical Laboratory, 1945-1995." (2000), 172-174.

[21] McQuillan, J., Richer, I., & Rosen, E., The new routing algorithm for the ARPANET. *IEEE transactions on communications*, 1980, **28**(5), 711-719.

[22] Catlett, C. E., The NFSNET: Beginnings of a National Research Internet. *Academic Computing*, 1989, **3**(5).

[23] Kemp, S., Internet users in Turkey, https://datareportal.com/reports/digital-2021-turkey, (Ziyaret Tarihi: 13 Mayıs 2021).

[24] Adrese Dayalı Nüfus Kayıt Sistemi Sonuçları, https://data.tuik.gov.tr/Bulten/Index?p=Adrese-Dayali-Nufus-Kayit-Sistemi-Sonuclari-2020-37210, (Ziyaret Tarihi: 13 Mayıs 2021).

[25] Hanehalkı Bilişim Teknolojileri (BT) Kullanım Araştırması, https://data.tuik.gov.tr/Bulten/Index?p=Hanehalki-Bilisim-Teknolojileri-(BT)-Kullanim-Arastirmasi-2020-33679, (Ziyaret Tarihi: 13 Mayıs 2021).

[26] Völske, M., Bevendorff, J., Kiesel, J., Stein, B., Fröbe, M., Hagen, M., & Potthast, M. (2021). Web Archive Analytics. *INFORMATIK 2020*."Internet History - Search Engines" (from Search Engine Watch), University Leiden,

Netherlands, September 2001, web: LeidenU-Archie. 13 Nisan 2009 tarihinde Wayback Machine sitesinde arşivlendi.

[27] Seymour, T., Frantsvog, D., & Kumar, S., History of search engines. *International Journal of Management & Information Systems (IJMIS)*, 2011, ***15***(4), 47-58.

[28] Web Arama Motoru, https://tr.wikipedia.org/wiki/web_arama_motoru , (Ziyaret Tarihi: 07 Mayıs 2021)

[29] Can Razbonyalı, Dikey Arama Motorları İncelemesi ve Bir Dikey Arama Motoru Uygulaması, 2011, 5-7

[30] Chien O.K., , Poo Kuan Hoong P.K., Ho C.C., A comparative study of HITS vs PageRank algorithms for Twitter users analysis, *International Conference on Computational Science and Technology (ICCST)*, 2014, 1-6, DOI: 10.1109/ICCST.2014.7045007.

[31] Borodin A., Roberts G.O., Rosenthal J.S., Tsaparas P., Link Analysis Ranking: Algorithms, Theory, and Experiments, ACM Transactions on Internet Technology, 2005, **5**(1), 231–297.

[32] Lempel R., Moran S., SALSA: The Stochastic Approach for Link-Structure Analysis, ACM Transactions on Information Systems, 2001, **19** (2), 131–160.

[33] Schwartz B., Google Has Confirmed It Is Removing Toolbar PageRank, https://searchengineland.com/google-hasconfirmed-they-are-removing-toolbar-pagerank244230, (Ziyaret Tarihi: 15 Mayıs 2021).

[34] PageRank, https://en.wikipedia.org/wiki/PageRank, Ziyaret Tarihi: 12 Mayıs 2021

[35] Gürdağ, B., & Özturan, C., Web Arama motorları için bağlantı temelli bir sıralama algoritmasının gerçeklenmesi. *TBD Kurultayı*, 2002.

KİŞİSEL YAYINLAR VE ESERLER

[1] **Altıntaş E Ş**, Yiğit H, Altıntaş A, Arama Motorları İçin Yeni Bir Web Sayfası Sıralama Algoritması Yaklaşımı, *6. Uluslararası Marmara Fen Bilimleri Kongresi* (IMASCON 2021) Kocaeli, 21-22 Mayıs 2021.

ÖZGEÇMİŞ

İlk ve orta öğretimini Çankırı'da tamamladı. 2009 yılında Nevzat Ayaz Anadolu Öğretmen Lisesi'nden mezun oldu. 2009 yılında kazanmış olduğu Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümünü bir yıl İngilizce hazırlık okuyarak 2014 yılında bitirdi. 2016 yılında Kocaeli üniversitesi bilişim sistemleri mühendisliği ana bilim dalında yüksek lisans eğitimine başladı. 2017 yılında Zeytinburnu belediyesine memur olarak atandıktan bir yıl sonra Ankara Mamak ilçe belediyesine nakil olarak, çalışma hayatına Mamak Belediyesi Bilgi İşlem Müdürlüğü'nde yazılım geliştirici olarak devam etmektedir.