



**ANLAMSAL EŞLEŞTİRME TEMELLİ OTOMATİK META ARAMA
MOTORU ve UYGULAMASI**

Yaşar GÖZÜDELİ

**DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

OCAK 2020

Yaşar GÖZÜDELİ tarafından hazırlanan “ANLAMSAL EŞLEŞTİRME TEMELLİ OTOMATİK META ARAMA MOTORU ve UYGULAMASI” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. M. Ali AKCAYOL

.....

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Başkan: Prof. Dr. Ebru AKÇAPINAR SEZER

.....

Bilgisayar Mühendisliği Ana Bilim Dalı, Hacettepe Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç. Dr. H. Şakir BİLGE

Elektrik Elektronik Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

.....

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç. Dr. Hacer KARACAN

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

.....

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Dr. Öğr. Üyesi Gönenç ERCAN

Bilgisayar Mühendisliği Ana Bilim Dalı, Hacettepe Üniversitesi

.....

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Tez Savunma Tarihi: 06/01/2020

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

.....

Prof. Dr. Sena YAŞYERLİ

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

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

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

Yaşar GÖZÜDELİ

06/01/2020

ANLAMSAL EŞLEŞTİRME TEMELLİ OTOMATİK META ARAMA MOTORU ve UYGULAMASI

(Doktora Tezi)

Yaşar GÖZÜDELİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Ocak 2020

ÖZET

Meta Arama Motorları günümüzde bilgi çıkarımında başarıyı artıran araçlardır. Özellikle Web’de gösterilen verilerin çok çeşitlenmesi nedeni ile veri çıkarımı ve semantik anlamlandırma oldukça büyük öneme sahiptir. Web arama motorları ve benzeri Web veri kaynakları günümüzde oldukça karmaşık bileşenlerden oluşur. Bu yapıların ürettiği verileri kullanmak, bileşenlerini ve çalışma ilkelerini yakından tanımakla mümkündür. Bu çalışmada özgün bir meta arama motoru sistemi tasarlanmış ve geliştirilmiştir. Sistem, son kullanıcı sorguları ve Web bilgi çıkarımı ile elde ettiği domain bilgilerini tekrardan sorguların ayrıştırılması ve derecelendirmesi için kullanmaktadır. Önerilen sistem sorgu domain’i kestirme, domain’e özgü arama motorlarını ve veri kaynaklarını otomatik olarak ekleme, arama sonuç bölgelerini web sayfalarından otomatik olarak tespit etme ve çıkarımı işlemini gerçekleştirmektedir. Geliştirilen meta arama motoru ile birden fazla veri tabanından elde edilen verilerin bir tek liste halinde kullanıcıya gösterimi sağlanmaktadır.

Bilim Kodu	: 92404
Anahtar Kelimeler	: Web bilgi çıkarımı, Web sonuç sıralama, Arama motorları, Meta-arama motorları, Arama sonuç kayıtları, Anlamsal sorgu analizi, k en yakın komşuluk,
Sayfa Adedi	: 105
Danışman	: Prof. Dr. M. Ali AKCAYOL

SEMANTIC MATCH BASED AUTOMATIC META SEARCH ENGINE AND
APPLICATION

(Ph. D. Thesis)

Yaşar GÖZÜDELİ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

January 2020

ABSTRACT

Meta search engines are among the tools increase the success of information extraction today. Data extraction and semantic interpretation are of great importance, especially due to the wide variety of data sources on the Web. Web search engines and Web data sources nowadays consist of highly complex components, as well. It is possible to consume the data produced by these structures by getting to know their components and working principles. In this study, a new meta search engine system is asserted from the search result extraction to the search result ranking. The system is used for re-classification and ranking of the domain information obtained by end-user queries and Web information extraction. The proposed system performs the query domain shortcuts, automatically adding domain-specific search engines and data sources, automatically detecting and extracting search result regions from web pages. With the developed meta search engine, the data obtained from more than one database is displayed to the user as a single list.

Science Code : 92404

Key Words : Web information retrieval, Web result ranking, Search engine, Meta-search engine, Search result record, Semantic query analyze, k-nearest neighborhood

Page Number : 105

Supervisor : Prof. Dr. M. Ali AKCAYOL

TEŞEKKÜR

Bu tez Sanayi ve Ticaret Bakanlığı San-Tez Programı çerçevesinde 0431.STZ.2013-2 numarası ile desteklenmiş ve Gazi Üniversitesi - Huawei ortak çalışması olarak gerçekleştirilmiştir. Destekleyen tüm kurum ve kuruluşlara teşekkürlerimi sunarım. Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren saygıdeğer hocalarıma ve özellikle de danışmanım Prof. Dr. M. Ali AKCAYOL'a, destek olan aileme ve özellikle de eşime ve çocuklarıma teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	x
ŞEKİLLERİN LİSTESİ	xi
SİMGELER VE KISALTMALAR.....	xiii
1. GİRİŞ.....	1
2. WEB ARAMA MOTORLARI	13
2.1. Veri Tabanı Boyutu	20
2.2. Güncellik	20
2.3. İşlem Kapasitesi	20
2.4. Teknoloji	21
2.5. Meta-Arama Motorlarına Duyulan İhtiyaç	21
3. META ARAMA MOTORLARI.....	27
3.1. Meta Arama Motorlarının Mimarisi.....	27
3.1.1. Helios mimarisi	28
3.1.2. TadPole mimarisi.....	29
3.1.3. TreeMap mimarisi	29
3.2. Meta Arama Motorlarının Süreç ve Bileşenleri	30
3.2.1. Web arama arayüz çıkarımı.....	30
3.2.2. Şema öznitelik/eleman eşlemesi.....	32
3.2.3. Global öznitelik analizi.....	34
3.2.4. Arama motoru bağlantısı	36
3.2.5. Arama sonuç çıkarımı.....	37

3.2.6. Arama sonuç gösterimi.....	38
3.2.7. Web bilgi çıkarımı ve makine öğrenmesi.....	40
4. OTOMATİK ŞEMA EŞLEME.....	47
4.1. Otomatik Sonuç Çıkarımı Algoritması Geliştirilmesi.....	47
4.1.1. Deneysel sonuçlar.....	52
4.2. Otomatik Arama Sonuç Çıkarımı Algoritması Geliştirilmesi.....	54
5. TASARLANAN META-ARAMA MOTORU.....	63
5.1. Geliştirilen Otomatik Meta-Arama Motoru Modeli.....	63
5.1.1. Çalışmanın literatür geçmişi.....	64
5.1.2. Web arama sorgu ayrıştırması	64
5.1.3. Web sonuç çıkarımı	65
5.1.4. Otomatik ASK çıkarımı ve birleştirme.....	66
5.1.5. Meta arama sonuç sıralaması ve rank.....	67
5.2. Model ve Bileşenler	68
5.2.1. İVT ile web arama sorgusunun anlamlandırılması ve sınıflandırılması.....	69
5.2.2. İVT'lerden kalıpların hesaplanması	70
5.2.3. Kalıp öbeklerinden yeni ivt'lerin keşfedilmesi	71
5.2.4. Tıklanan web adresi bilgisinin öğrenilen loglara dâhil edilmesi.....	72
5.2.5. Arama sorguları için domain öğrenme aşamaları.....	73
5.2.6. Web sonuç çıkarımı.....	74
5.2.7. Web sonuç birleştirme	76
5.2.8. ASK ağacı derinlik azaltma algoritmasının sisteme eklenmesi	76
5.2.9. Şema eşleme sistem bileşeni	77
5.2.10. Web sonuçlarının sıralanması.....	78
5.2.11. Sonuçların sıralanmasına ilişkin başarımın ölçülmesi.....	80
5.2.12. Önerilen sıralama algoritması: SemiRank.....	81

Sayfa

5.3. Kullanılan Mimariler ve Sistem Özellikleri	82
6. DENEYSEL SONUÇLAR.....	85
6.1. Arama Sorgularının Sınıflandırılmasına İlişkin Başarımlar.....	85
6.2. Web Sonuç Bölgesinin Bulunması.....	87
6.2.1. ASK bölgesinin bulunmasına ilişkin başarımlar	87
6.2.2. ASK öge çıkarım ve eşleştirmeye ilişkin ölçüm sonuçları.....	88
6.3. Sonuçların sıralanmasına ilişkin başarımın ölçümlenmesi	89
6.4. SemiRank yaklaşımının diğer yöntemlerle kıyaslaması	90
6.4.1. Sorguların çalışma zamanlarının kıyaslanması	91
7. SONUÇLAR VE DEĞERLENDİRMELER	93
7.1. Sonuçlar.....	93
7.2. Gelecekte Yapılabilecek Çalışmalar	93
KAYNAKLAR	95
ÖZGEÇMİŞ	103
DİZİN.....	105

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 1.1. Meta arama motorlarından kullanıcı beklentileri	10
Çizelge 2.1. Belli başlı düşey arama motorları	13
Çizelge 2.2. Google, Yahoo ve Bing arama motorlarının özellikleri	22
Çizelge 2.3. Google, Yahoo, Live ve Ask arama sonuçlarının çakışma oranları	24
Çizelge 3.1. Şema çıkarım algoritmasının schema.org ve foaf ile kıyaslanması.....	42
Çizelge 3.2. Çıkarımı yapılan varlıklar için doğruluk kıyaslaması	44
Çizelge 4.1. Veri Kümesi için düzenli ve düzensiz ASK'ların oranları	53
Çizelge 4.2. Farklı arama motorları için Duyarlık ve Doğruluk değerleri.....	54
Çizelge 5.1. Örnek bir arama için oluşturulmuş aday düğüm.....	75
Çizelge 6.1. Örnek Sorgular ve tahmin edilen domainler.....	86
Çizelge 6.2. Farklı arama motorlarına ilişkin precision ve recall değerleri.....	87
Çizelge 6.3. TSAP algoritması ile ranking sonucun değerlendirilmesi	89
Çizelge 6.4. Deneylere ilişkin standart sapma ve precision değerleri	90
Çizelge 6.5. Örnek bir TSAP değerlendirme formu	90
Çizelge 6.6. TSAP algoritması ile ranking sonucun değerlendirilmesi	91
Çizelge 6.7. İlk 100 sonuç için çalışma süreleri kıyaslanması	91
Çizelge 6.8. ilk 10 sonuç için deneylere ait precision ve standart sapma	92

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Arama motoru bileşenleri	16
Şekil 2.2. En iyi arama motorlarına yönelik araştırma sonuçları.....	19
Şekil 3.1. Helios mimarisi.....	28
Şekil 3.2. TadPole meta arama mimarisi	29
Şekil 3.3. TreeMap mimarisine sahip meta arama motoru	30
Şekil 3.4. Kendi kendine ontolojik öğrenme yapan bir sistemin bileşenleri	41
Şekil 3.5. Otomatik şema çıkarım sistem mimarisi.	41
Şekil 3.6. İteratif bir Bilgi Çıkarım aracının gerçekleştirme mimarisi	43
Şekil 3.7. Semantik analiz ve sorgu işleme ile meta-arama motoru mimarisi	45
Şekil 4.1. Düzensiz ASK'lar içeren örnek bir Google arama sonucu.....	49
Şekil 4.2. Bir düzenli ASK için basitleştirilmiş ağaca ait gösterimler.....	51
Şekil 4.3. ASK DOM ağacından bir kısmı ve bu kısmın basitleştirilmesi.	52
Şekil 4.4. Farklı arama motorlarında düzenli ve düzensiz ASK'lar.....	55
Şekil 4.5. Bir ASK kaydının HTML-DOM ağacı ile gösterimi.....	57
Şekil 4.6. Bir HTML-DOM ağacının seviyesinin indirgenmiş hali.....	57
Şekil 4.7. ASK'lar için Frekans Ağırlıklı Öge ağacı	58
Şekil 4.8. Sonuçların birleştirilmiş hali.....	60
Şekil 5.1. Otomatik meta arama motoru modeli	68
Şekil 5.2. Sorgulardaki kelime sayılarına göre frekans yüzdelikleri	70
Şekil 5.3. İVT'ler aracılığıyla arama loglarının öğrenilmesi	72
Şekil 5.4. Super Mario İVT'si için tıklanan web adresleri.....	73
Şekil 5.5. Web Çıkarım genel akış algoritması.....	76
Şekil 5.6. Ağaç derinliğinin azaltılması.....	77
Şekil 5.7. Örnek bir Frekanslandırılmış ASK öge ağacı.....	78
Şekil 5.8. Domain benzerliği yok iken arama sonuç sayfası	79

Şekil**Sayfa**

Şekil 5.9. Domain bilgisinin aramaya dâhil edilmesi	80
Şekil 6.1. Örnek bir meta-arama sonucu.....	86
Şekil 6.2. Eşleştirme başarımında ASK öge sayısına bağlı F-Skor grafiği	88
Şekil 6.3. Bir arama sonuç sayfasında sonuçların gösterildiği bölge alan oranları	89
Şekil 6.4. Örnek bir sorgu için precision değerleri	91

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
AOL	America Online
API	Application Programming Interface
ASK	Arama Sonuçları Kaydı
CSS	Cascading Style Sheets
DDİ	Doğal Dil İşleme
DOM	Document Object Model (Doküman Obje Modeli)
DWDE	Deep Web Data Extraction (Derin Web Veri Çıkarımı)
DWDE-IR	Deep Web Data Extraction-Information Retrieval
GBA	Görsel Blok Ağaçları
GE	Global Elements (Global Elemanlar)
HTML	Hypertext Markup Language (Hipertext İşaretleme Dili)
HTTP	Hypertext Transfer Protocol
IMDB	İnternet Movie Database (Uluslararası Film Veritabanı)
İÖV	İçerik özellik vektörü
IR	Information Retrieval (Bilgi Çıkarım)
İVT	İsimlendirilmiş Varlık Tanımlama
JSON	JavaScript Object Notation
LE	Local Elements (Yerel Elemanlar)
LSA	Latent Semantik Analizi
R	Retrieval (Getirme, Çıkarım)
RAN	Representative Attribute Name (Temsili Özellik Adı)
REST API	Representational State Transfer Application Protocol Interface
RI	Random Indexing (Rasgele İndeksleme)
SEO	Search Engine Otimization (Arama Motoru Optimizasyonu)
SPARQL	SPARQL Protocol and RDF Query Language
SRR	Search Result Record (Arama Sonuçları Kaydı)
SVM	Support Vector Machine (İstatistikte bir metot)

Kısaltmalar**Açıklamalar****TF/IDF**

Bir dokuman için belirgin terimleri seçen algoritma (Term Frequency– Inverse Document Frequency)

TSAP

TREC- Style Average

UPA

Uygulama Programlama Arayüzü

URL

Uniform Resource Locator (Tekil Kaynak Bulucu)

YSA

Yapay Sinir Ağları

XML

Extensible Markup Language (Genişletilebilir Biçim Dili)

1. GİRİŞ

1990'lı yılların başında, Web gelişip insanların gezinemeyeceği büyüklüğe ulaştığında, arama işlemi için araçlara ihtiyaç duyulmaya başlanmıştır (Berners-Lee, Cailliau, Groff ve Pollermann, 1992). İlk önceleri bir telefon rehberini andıran Web katalogları editörler tarafından kategorilere ayrılarak bir araya getiriliyor, böylece insanların istedikleri Web adresine bir tık kolaylığında ulaşmaları sağlanmaktaydı.

Sonraki yıllarda İnternet'e erişen kullanıcı sayısının da artması ile birlikte Web içerikleri hızla artmış ve Web dünyadaki herkese açık, heterojen, en büyük veri ve bilgi kaynaklarından biri haline gelmiştir. Buna bağlı olarak da İnternet'e erişen kullanıcı sayısı ve daha etkin arama teknik ve araçlarına olan gereksinim devasa boyutlara ulaşmıştır. 2000 yılında ilk defa indekslenen sayfa sayısında 1 milyar sınırı aşılrken, 2010 yılında, Web arama motorlarından önde gelenlerinden Google'ın indekslediği sayfa sayısı 30 milyarı'ı geçmiştir. 2019 yılı Kasım ayı itibari ile 1,7 milyardan fazla kayıtlı domain sayısı ve 4,4 milyardan fazla İnternet erişimi olan kullanıcı bulunmaktadır ("İnternet: İnternet Live Stats - İnternet Usage & Social Media Statistics", y.y.). Web'de hangi içeriğin hangi adreste olduğunu bilmek için Web içeriklerini otomatik olarak kategorize edecek yöntemler ve araçlara ihtiyaç duyulmuştur. Yapılan bir araştırmaya göre Web gezinimlerinin %80'i bir arama motoru ile başlamaktadır (Brown, 2013). Web Arama motorları bu ihtiyacı gidermek için tasarlanmıştır ve kısa sürede Web içeriklerini toparlayıp kopyalarını getiren *örümcek* (crawler); getirilen verileri indeksleyen bir *indeksleyici* (indexer), kullanıcıdan gelen sorguları işleyip cevaplandıran bir *sorgu işleyici* (query processor) olmak üzere üç bölümden oluşan araçlardır (Risvik, Aasheim ve Lidal, 2003).

Bilgi Çıkarımı (Information Retrieval), bilgi parçacıklarının sunumu, saklanması, organize edilmesi ve erişilmesi ile ilgilenmektedir. Bir bilgi çıkarım aracı, kullanıcı isteğine uygun verilerin (dokümanların) içeriklerini bir şekilde yorumlayarak ilgiliklerine göre uygun bir sıralama ile sorgu cevabı şeklinde göstermesi gerekir (Tallack, 1997). Arama motorları ve meta arama motorları birer bilgi çıkarım araçlarıdır.

Son kullanıcılar açısından arama motorları günümüzde milyarlarca sayfa arasından istenen sayfaların bulunup getirilmesini sağlayan araçlardır. Genellikle sıradan bir arama işleminde

yüzlerce hatta binlerce Web sayfası sonuçta yer alabilir. Sonuçların arama ile ilgililik durumlarına göre sıralanması özel bir sıralama işlemidir (Ranking) ve arama motorunun başarımı açısından oldukça önemlidir. Arama motorlarının ilk yıllarında bir arama için sorgu işleme ve sonuç sıralama algoritmaları oldukça basit yapıya sahipti. Takip eden yıllarda arama motorlarında Web sitelerinin birbirleri ile olan bağlantıları, kelimelerin anlamsal ifadeleri gibi etmenleri de dâhil ederek daha karmaşık yöntemler kullanılmaya başlanılmıştır. Başlarda sadece İnternet'i kullanan azınlığı ilgilendiren bu algoritmalar, zaman içerisinde hayatın bir parçası olmuş ve özellikle işletmeler için Web sayfalarının hangi sırada geldiği önemli hale gelmiştir¹ (Linner, 2008). Bu durum, Web sitesinin rank değerini yükseltmek isteyen Web geliştiriciler arasında *arama motoru optimizasyonu* (search engine optimization) olarak anılan bir dize tekniklerin geliştirilmesine yol açmıştır ve Web sayfalarının tasarımına yön vermeye başlamıştır (Gözüdeli, 2009).

1990'lı yılların sonlarında başlayan çalışmalar, özellikle arama motorlarının Web'in büyük bir kısmını indeksleyemediğinden hareketle Web'in bu kısmına derin web adı verilmiştir (Bergman, 2001).

Web meta arama motorları, kendilerine ait bir indeksleri olmadığı halde Web arama motorlarından sonuçları arama anında bulup getirerek kullanıcıya gösteren bilgi çıkarım araçlarıdır. Meta arama motorlarından ilki Washington Üniversitesi tarafından geliştirilen MetaCrawler'dır (Selberg ve Etzioni, 1995, 1997). MetaCrawler arama motoru birden fazla arama motorunu seçerek sonuçlarını almakta ve birleştirme işleminde ise tekrarlı olanları silme gibi basit işlevleri yerine getirmektedir.

Daha sonra geliştirilen Mamma Search, Carleton Üniversitesi tarafından geliştirilmiş olup; rSort adında sıralama algoritmasını sonuçlar üzerinde uygulamıştır. Tekrarlı sonuçlar her bir arama motoru için oylama olarak düşünülerek rank değerleri hesaplanmıştır (Selberg ve Etzioni, 1997).

¹ Bkz. Corciya'lı ayakkabı satıcısı Neil Moncrief'in hikâyesi (the Search, John Battelle)

Meta arama motorlarının tarihi gelişimi

SavvySearch meta arama motoru, sorguya göre ilgili olan arama motorlarının seçiminde meta-indeks adında bir yaklaşım kullanmıştır. SavvySearch daha önceki arama sonuçlarına tıklanma sıklığını arama motorlarının seçiminde kullanmıştır (Adele E. ve Dreilinger, 1997; Dreilinger, 1996).

ProFusion, Kansas Üniversitesi tarafından geliştirilmiştir. ProFusion, daha önceki bilgilere dayanarak arama motorlarının güvenilirliklerini belirlemekte ve gelen sonuçların manuel veya otomatik olarak kategorize edilmesini sağlamaktadır (Gauch, Wang ve Gomez, 1996). Gelen sonuçlardan aynı olanları çıkarma, sorunlu linkleri çıkarma, gelen sonuçları birleştirme ve yeniden rank değeri hesaplama işlemleri yapmaktadır. Metaseek, Kolombiya Üniversitesi tarafından geliştirilmiştir ve içerik tabanlı meta arama motoru olup resim arama işlemi yapabilmektedir (Benitez, Beigi ve Chang, 1998). Anvish, Louisiana Devlet Üniversitesi tarafından geliştirilmiştir ve yapay sinir ağları tabanlı doküman sınıflandırma algoritması kullanmaktadır (Dorn ve Naz, 2008; Kak, 1999). Web sayfalarının sınıflandırılması için başlık ve kısa özet bilgilerini kullanmaktadır. Her arama motorundan gelen ilk iki sonucun ilgili olduğunu değerlendirerek, başlık ile özet içerisindeki anahtar kelimelere göre dokümanların ilgililik düzeylerini belirlemektedir.

KOMET meta arama motoru Karlsruhe Üniversitesi tarafından geliştirilmiştir. Kendine özgü geliştirilen mantık tabanlı tanımlama dili kullanarak sonuçların birleştirilmesini gerçekleştirmektedir. PrologCrawler, Pisa Üniversitesi tarafından geliştirilmiş olup; Prolog mantıksal programlama dilini sonuçları birleştirmek için kullanmıştır. CleverSearch, bir meta arama motorudur ve MetaCrawler, MetaFind, Mamma, Surfy, Dogpile, SavvySearch, HuskySearch, AskJeeves, CNN ve ProFusion gibi diğer meta arama motorlarının sonuçlarını birleştirmektedir ("Internet: A Brief History of Search Engines", 2004).

Inquirus, NEC araştırma enstitüsü tarafından geliştirilmiştir. Arama motorlarından gelen sonuçlardaki dokümanların orijinallerini getiren, bunlar üzerinde analiz yapan meta arama motorudur (Lawrence, 1998). Sonuçları oluşturan arama motorlarının rank algoritmalarını da kullanmaz. Daha sonraki eklentilerle, Inquirus2 geliştirilmiştir ve konu bazlı sorgu iyileştirme, makine öğrenme yöntemleri ile sınıflandırma ve kendisine özgü rank algoritması eklenmiştir. Metaweaver, Tokyo Teknoloji Enstitüsü tarafından geliştirilmiştir. Profusion

ve SavvySearch gibi daha önceki sorgu sonuçlarına bağlı olarak öğrenebilen ve yeni sorgularda hangi arama motorunun seçilmesi gerektiğine karar veren adaptif bir sisteme sahiptir. Sorgu sonuçlarının tıklanma durumlarına göre sürekli kendisini güncellemektedir.

Ithaki, İngilizcenin yanı sıra 14 dilde arama yapabilmektedir (Pinkerton, 2000). Önemli arama motorlarını kullanmakta ve çok farklı kategorilerde (mp3, haberler, resim) dikey arama yapabilmektedir. Bu ve benzeri dikey meta arama motorları farklı alanlarda özel aramalar yapmak için geliştirilmiştir. Örneğin, MedSpider medikal alanda arama yapmak için, NanoSpider nanoteknoloji konusunda arama yapmak için, Cancer Meta Spider kanser alanında arama yapmak için geliştirilmiştir (Chau ve diğerleri, 2002; Chen, Fan, Chau ve Zeng, 2003).

Vivisomo, Carnegie Mellon Üniversitesi tarafından geliştirilmiştir. Vivisomo, arama motorlarından gelen metin tabanlı sonuçları gruplandırmak için sezgisel algoritma kullanmaktadır (“Internet: NetRatings: Search engine ratings”, 2006). MetaSpider, Arizona Üniversitesi tarafından geliştirilmiştir. MetaSpider, metin tabanlı öbekleme yöntemi kullanan, aynı anda 6 arama motoruyla birlikte çalışan ve gelen sonuçları birleştiren meta arama motorudur. Self organized map kullanılarak Web sayfaları öbeklenmektedir.

CSams, Binghamton Üniversitesi tarafından geliştirilmiştir. Yüksek seviyede ölçeklenebilirlik sağlamaktadır ve çok sayıda farklı alanda araştırma yapmak için kullanılmaktadır (Z. Wu, Meng, Yu ve Li, 2001). Mearf, Minnesota Üniversitesi tarafından geliştirilmiştir. Kullanıcı sorgusuna bağlı rank algoritmasına sahiptir (Oztekin, Karypis ve Kumar, 2002). Rank algoritması uzman görüşlerine dayalı elde edilen sonuçların rank değerlerini oluşturmakta ve kullanıcıya sunulmaktadır. Kartoo, Flash teknolojisi kullanılarak elde edilen sonuçların sorgudaki anahtar kelimelerle bağlantısını kullanıcıya göstermektedir (Brin ve Page, 1998). Turbo10, özellikle deep Web kaynakları içerisinde arama kapasitesine sahip olan meta arama motorudur (Hamilton, 2003). Turbo10, farklı arama motorlarından gelen sonuçların çıkartılması ve birleştirilmesi için kendine özgü araçlara sahiptir.

Precise resim meta arama motorudur. Bulanık mantık ve yapay sinir ağları kullanarak Web içerisindeki insan fotoğraflarını aramak için kullanılmaktadır (Anagnostopoulos I., Anagnostopoluos C., Psoroulas I., Loumos V., 2003). Squirrel, Uppsala Üniversitesi

tarafından geliştirilmiştir. Az kullanılan diller için metin bilgileri toplar. Eğitimciler için güncel ve yeterli düzeyde materyal elde etmek için kullanılmaktadır (Nilsson, 2003).

Guided Google, Melbourne Üniversitesi tarafından geliştirilmiştir ve Google Web servislerini kullanır. Bu meta arama motoru anahtar kelime bazında arama yapmak için kullanılmaktadır (Ding ve Buyya, 2004).

LinbaCrawler, Teksas A&M Üniversitesi tarafından geliştirilmiştir. Sonuçlarda otorite doküman bulundurmaz tüm dokümanlar kendilerine link yapılan dokümanlardır yani hub dokümanlardır (Li, Zhang, Shang ve Liu, 2004). Helios, Pisa Üniversitesi tarafından geliştirilmiştir. Web bilgi çıkarımı araçları tasarlamak için geliştirilmiş açık kaynak bir meta arama motorudur. Doküman elde edilmesi aşamasında paralel çalışan bir yapıya sahiptir. Böylece doküman elde etme süresi oldukça kısaltılmaktadır (Gulli ve Signorini, 2005).

Snaket meta arama motoru, Helios meta arama motorunu kullanarak geliştirilmiştir. Kullanıcıya iki tane sonuç listesi sunar. Birincisi, klasik rank listesi diğeri ise hiyerarşik olarak sonuçlar arasındaki bağlantıları gösteren sonuç listesidir (Gulli ve Signorini, 2005). Takip eden yıllarda, arama motorları da derin web'i indekslemek için meta arama motorları ile benzer yöntemleri kullanmıştır. Öte yandan, arama motorları da meta-arama motorları tarafından sonuçlarının kullanılması konusuna sıcak bakmamaktadır ve konunun hukuki tarafı halen tartışma konusudur. Derin Web kavramından sonra, Meta arama motorları için yapılan çalışmalar, arama motorları örümcekleri tarafında da derin Web'in sorgulanması amacıyla kullanılmıştır.

Mevcut arama motoru seçme yöntemleri

Mevcut yaygın kullanılan meta arama motorlarının seçme algoritmaları incelenmiştir. Meta arama motorlarının kullandığı arama motorunun seçilmesine yönelik çok sayıda çalışma yapılmıştır. Meta arama motorları kullanıcının geçmiş sorgularına göre, alınan ilgililik geri bildirimi durumuna göre, sorgu içeriğine göre farklı yöntemler kullanarak arama motoru seçme işlemini gerçekleştirmektedir. Kullanılan yöntemler üç farklı kategoride değerlendirilmektedir:

- Genel yaklaşımlar
- İstatistiksel yaklaşımlar
- Öğrenen yaklaşımlar

Genel yaklaşımlar, belirli bir alana yönelik arama motorlarının kullanıcı sorgusundan elde edilen domain bilgisi ile uyumluluk durumuna göre seçme işlemini gerçekleştirmektedirler (ALIWE, WAIS). Burada başarılı bir seçme işlemi için sorgu içeriğinin doğru bir şekilde sınıflandırılması gerekmektedir.

İstatistiksel yaklaşımlar, seçilecek arama motorlarından daha önce elde edilen bilgiler de göz önüne alınarak kelime sıklıklarına dayalı seçme işlemi yapmaktadırlar (D-WISE, CORINET, g-GLOSS, ENPUD). Burada da başarılı bir seçme işlemi Web arama motorlarının sağlayacağı içerikteki kelime sıklıklarının doğru belirlenmesine bağlıdır.

Öğrenebilen yaklaşımlar, geçmiş sorgularda elde edilen sonuçları öğrenebilen sistemler üzerinde işleyerek kullanıcı relevance feedback durumuna da bağlı bir şekilde sonraki sorgularda seçilecek arama motorlarını belirlerler (QUERY SIM, MRDD, SAVVY SEARCH, PROFUSION). Lui ve arkadaşları (Liu, Xu ve Long, 2017) YSA yöntemleri kullanarak arama motoru seçimi yapmaktadır. Burada da başarılı bir seçme işlemi için öğrenebilen modelin iyi oluşturulmasına bağlıdır.

Meta arama motorlarında ranking ve algoritmaları

ProFusion, daha önceki arama bilgilerine dayanarak arama motorlarının güvenilirliklerini belirlemekte ve gelen sonuçların manüel veya otomatik olarak kategorize edilmesini sağlamaktadır (Gauch ve diğerleri, 1996). Gelen sonuçlardan aynı olanları çıkarma, sorunlu linkleri çıkarma, gelen sonuçları birleştirme ve yeniden rank değeri hesaplama işlemleri yapmaktadır.

Harvester, biyoinformatik üzerine araştırma yapmak için kullanılan dikey meta arama motorudur (Liebel, Kindler ve Pepperkok, 2004). Temel olarak genler ve protein içerikli bilgileri aramak için kullanılır. Çapraz linkler biyo-informatik kaynaklarında çok popülerdir ve Harvester çapraz aramalar yapmak için kullanılır. Harvester'in kullandığı Rank algoritması Google'ın kullandığı PageRank algoritmasına benzer yapıya sahiptir.

ProThes, kendisine özel sorgu girişi ve sonuçların görüntülenmesi için grafik arayüze ve sorguların özelleştirilmesi için sözlüğe sahiptir (Braslavski, Alshanski ve Shishkin, 2004; Braslavski ve Shishkin, y.y.). Ayrıca, elde edilen sonuçların tek bir liste halinde birleştirilerek gösterilmesi için basit sezgisel kuralları ve rank algoritmasını kullanmaktadır.

Meta aramada en önemli işlem, arama motorlarından alınan sonuçların birleştirilmesidir. Öncelikle, farklı arama motorlarından gelen sonuçlardan aynı olanların belirlenmesi gereklidir. Elde edilen sonuçlarda bulunan aynı sayfaların silinmesi ve sonuç listede bir tane yer alması gereklidir. Ayrıca, meta arama motorunun elde edeceği liste için farklı arama motorlarının rank değerlerinin birleştirilmesi gereklidir. Elde edilen sonuçların benzerliğine göre veya rank pozisyonlarına göre birleştirme işlemi yapılmaktadır. Literatürde yaygın kullanılan ranking yöntemleri:

- Borda ranking
- Condorcet ranking
- Reciprocal ranking
- Öğrenen yaklaşımlar

olarak sıralanabilir.

Borda ranking

Aday sayfaların oylanmasına dayanan birleştirme yapmaktadır. Toplam n tane aday varsa, bir oylama sonucunda birinci sırada yer alan n puan, ikinci sırada yer alan $n-1$ puan, ... şeklinde devam eder. Eğer bir aday oylanmazsa kalan puan oylanmayan adaylar arasında eşit dağıtılır. En yüksek puan alan aday kazanan adaydır. Sıralama puanı yüksek olandan düşük olana doğru yapılmaktadır.

Condorcet ranking

Condorcet ranking kullanan bir rank çalışması Dwork ve arkadaşları tarafından yapılmıştır. (Dwork, Kumar, Naor ve Sivakumar, 2001). Her adayın ikili olarak diğerleriyle sıralamadaki yerinin karşılaştırılmasına dayanmaktadır. İkili sıralama şu üç halden biri ile son bulabilir:

- Bir oylamada bir aday diğetine göre daha yüksek rank değerine sahipse kazanmış olur (win).
- Bir oylamada bir aday diğetine göre daha düşük rank değerine sahipse kaybetmiş olur (lose).
- Bir oylamada iki adayda oylanmadıysa birbirlerine bağlanmaktadır (tie).

Reciprocal ranking

Her oylamada birinci sıradaki 1 puan, ikinci sıradaki 1/2 puan, üçüncü sıradaki 1/3 puan, ... şeklinde puan almaktadır. Eğer bir aday oylanmazsa boş geçilir. Sonuç rank değerleri elde edilen skorların toplamı alınarak elde edilir. Skor değeri yüksek olan ilk sırada yer alır. Üst sıradaki adaylara Borda ranking yöntemine göre daha yüksek puan verir.

QuadRank (Akritidis, Katsaros ve Bozaniş, 2011) Reciprocal ranking'e benzer bir bir yaklaşıma arama yapan kullanıcının yerel konumu ile arama adres bilgisinin domain yerelliğinin birbirine yakınlığını dikkate alan ve ayrıca arama teriminin arama sonuçlarındaki terim frekansı/ters dokuman frekansı gibi yardımcı bilgilerle de birleştiren bir rank algoritması kullanır.

Öğrenen yaklaşımlar

Arama sonuçlarını sıralarken öğrenen yaklaşımlardan faydalanan çalışmalar da mevcuttur. Vijaya ve arkadaşları (Vijaya, Raju ve Ray, 2016) Yapay Sinir Ağları(YSA) öğrenmesine dayalı bir sıralamadan yararlanmıştır. Yapılan çalışmada ASK'lara ait başlık, sıra, özet metin, tekrar sayısı, Web adresi domaini gibi bileşenler girdi parametresi olarak verilmiştir. QuadRank'a göre biraz daha uzun bir çalışma zamanı ile başarımını biraz daha artırdığını göstermiştir.

Kumar yaptığı çalışmada rank işleminin genetik algoritma ile yapılabilirliğini incelemiştir. (Kumar, 2017)

Kullanıcıların meta arama motorundan beklentileri

Srinivas ve arkadaşları tarafından yapılan çalışmada kullanıcıların Meta-Arama motorlarından beklentileri incelenmiştir. Arama motoru olarak, Ask, Bing, Google ve Yahoo incelenmiştir (Pandey, 2018). Meta arama motoru olarak da Web Crawler, Meta Crawler, Dogpile, Brainboost, Culsty ve Chunkit incelenmiştir. Meta arama motorları kullandıkları arama motoru sayısı, cevap süresi ve ranking algoritması açısından değerlendirilmiştir. Çalışmanın yapıldığı zamana ait en fazla arama motoru kullanan meta arama motoru olan Brainboost; 16 tane arama motorunu eş zamanlı olarak kullanmaktadır. En az kullanan ise Metacrawler olup; 5 tane arama motorunu kullanmaktadır. Meta arama motorunun cevap süresinin kullanılan arama motoru sayısının yanı sıra uygulanan ranking algoritmasına ve yapılan işlem sayısına (duplikasyonların elimine edilmesi, ranking işlem karmaşıklığı) bağlı olduğu görülmüştür.

Yadav ve Sangwan tarafından yapılan çalışmada ise arama motorlarından kullanıcıların memnuniyet düzeylerinin tahmini için yapay sinir ağları kullanılmıştır (Yadav ve Prakash Sangwan, 2011). Kullanıcı memnuniyet düzeylerini aşağıdaki faktörlerin etkilediği görülmüştür:

- Kullanıcı dostu arayüz
- Ekstra hizmetler (spell check, anlık haberler, resimler, lokal sonuçlar, ...)
- Uzun anahtar kelime listesi
- Cevap süresi
- Sonuçların sunumu (sıralama, numaralama şekli)
- Kapsama alanı (Web üzerinde toplam içerik miktarı)
- Sorgu çeşitliliği (Belirsizlik içeren sorgularında sonuçlandırılması)
- Sonuçların güncelliği (Orişinal Web sayfalarındaki değişimi algılama süresi)
- Sonuçların ilgililik düzeyi
- Tazelik (Değişen sayfaları sürekli güncelleme kapasitesi)
- Coğrafiik bilgilerin sağlanması
- Güvenilirlik (Linklerde hata olmaması)
- Yardım veya destek (Kullanıcıya ihtiyaç duyduğunda yardım sağlanması)
- Mevcudiyet (Her zaman meta arama motorunun kullanılabilir olması)

- Sürekli upgrade edilmesi (Meta arama motorunun sürekli güncellenmesi)

Yadav ve Sangwan, bir grup kullanıcıya yazılı anket uygulamışlar ve aşağıdaki tabloda verilen değerleri elde etmişlerdir.

Çizelge 1.1. Meta arama motorlarından kullanıcı beklentileri

Önem derecesi	Faktör	Normalize edilmiş skor
Primary 1	Cevap süresi	0,037
Primary 2	Sonuçların güncelliği	0,027
Primary 3	Güvenilirlik	0,037
Primary 4	İlgililik	0,034
Secondary 1	Sonuçların sunumu	0,088
Secondary 2	Kullanıcı dostu arayüz	0,076
Secondary 3	Sorgu çeşitliliği	0,073
Secondary 4	Coğrafik bilgiler	0,086
Secondary 5	Ekstra hizmetler	0,113
Secondary 6	Az sıklıktaki sorgular içinde faydalı bilgi sunulabilmesi	0,073
Secondary 7	Kapsama alanı	0,076
Secondary 8	Mevcudiyet	0,067
Secondary 9	Yardım veya destek	0,104
Secondary 10	Sürekli upgrade edilmesi	0,110

Benzer çok sayıda çalışmada meta arama motorlarının kullanıcı memnuniyetini ölçmeye yönelik araştırmalar yapılmıştır (Hamilton, 2003; P. V. S. Srinivas and A. Govardhan, 2011). Bu araştırmalarda da yukarıda verilen faktörler farklı önem sırasına göre elde edilmiştir.

Bu tez çalışmasında geliştirilen meta arama motorunda, farklı arama motorlarından gelen sonuçların anlamsal eşleştirilmesi ve elde edilen sonuçların yeniden sıralanması için ranking algoritması geliştirilmesi temel amaçlar olarak alındığından aşağıdaki faktörlerin verilen sırada göz önüne alınmasına karar verilmiştir.

- Cevap süresi
- İlgililik
- Sonuçların sunumu
- Güvenilirlik
- Kullanıcı dostu arayüz
- Mevcudiyet
- Sonuçların güncelliği
- Sorgu çeşitliliği

Tez çalışmasının yapısı

Bu tez çalışmasında ikinci bölümde Web Arama Motorları hakkında genel bilgi verilmekte ve yapılan çalışmalarda kullanılan yöntemler incelenmiştir. Üçüncü bölümde meta arama motorlarının mimari gelişimi ve bileşenleri anlatılmıştır. Dördüncü bölüm otomatik şema eşleme yaklaşımları hakkındaki çalışmaları incelemekte ve yöntemleri anlatılmıştır. Beşinci bölümde geliştirilen meta-arama motoru ve bileşenleri yer almaktadır. Altıncı bölüm deneysel çalışmaları içermektedir. Son bölüm ise sonuçlar, değerlendirmeler ve gelecekte yapılabilecek çalışmaları özetlemektedir.

2. WEB ARAMA MOTORLARI

Bilgi çıkarımı, ihtiyaç duyulan bilgiyi bir grup doküman içerisinde arama sürecidir (Langville ve Meyer, 2011). Bilgi çıkarımı, kullanıcının istediği bilgiye kolay erişimini sağlamak için bilginin gösterimi, saklanması, organizasyonu ve erişimiyle ilgilenir (Baeza-Yates, Ribeiro-Neto ve Others, 1999). Klasik bilgi çıkarımı yöntemleri birbirleri arasında bağlantı olmayan ve iyi yapılandırılmış belirli bir grup doküman üzerinde arama yapar. Ancak, Web bilgi çıkarımı için arama yapılan dokümanlar dünya genelinde çok büyük boyutlardadır, heterojen yapıya sahiptirler ve aralarında bağlantılar bulunmaktadır. Klasik bilgi çıkarımı amaçlı arama yapılan dokümanlar genellikle iyi yapılandırılmış ve belirli bir biçime sahiptir. Ancak, Web bilgi çıkarımı için kullanılan dokümanlar son derece değişken yapıya sahiptir ve Dünya üzerinde çok farklı yerlerde, çok farklı yazılım ve teknolojiler kullanılarak saklanmış şekilde bulunmaktadır.

Arama motorları Web Arama Motorları, seçme temelli arama motorları, meta arama motorları, masa üstü arama motorları, web portal arama motorları ve düşey market arama motorları olmak üzere çeşitli gruplarda incelenmektedir (Brown, 2013). Düşey arama motorlarının alanlara göre dağılımı Çizelge 2.1.'de yer almaktadır.

Çizelge 2.1. Belli başlı düşey arama motorları

Alan	Arama motorları
Uçuş/Seyahat	SkyScanner.net Momondo.com Hipmunk
Blog	IceRocket.com BlogPulse.com eBuzzing Labs
Resim	TinEye.com PicSearch.com
İnsan	Pipl.com 123people.com
Forum	BoardReader.com
Müzik	MixTurtle.com SongBoxx.com Music-Map.com AudioGalaxy.com Grooeshark

Çizelge 2.2. (devam) Belli başlı düşey arama motorları

Video-Görüntü	Truveo PodScope.com Blinkx.com
Kaynak	SlideFinder.net FileDigg.com
Domain	Panabee.com NameNinja.com
Icon	Iconfinder.com. Iconseeker.com IconArchive.com
Özel arama	DuckDuckGo.com
Benzer web siteleri	SimilarSites.com
Etkinlik	Zvents Eventful
Ürün	Amazon theFind Shopping
Sağlık	WenZher Medlio Medical World Search
Biyoinformatik	Harvester
Çok yönlü	Blekk.com KeoTag.com Scour.com Greplin.com

İnternet üzerindeki bilginin elde edilmesi için farklı yöntemler kullanılabilir ancak en hızlı ve kolay yolu arama motorlarını kullanmaktır. Kullanıcılar çoğunlukla araştırma, eğitim, iş, alışveriş veya eğlence amacıyla İnternette bilgi aramaktadırlar. Günümüzde en yaygın kullanılan üç büyük arama motoru Google, Yahoo! ve Bing arama motorudur. Bunlar kullandıkları veri tabanı büyüklüğü, arama etkinliği, kapasitesi ve teknolojileri açısından farklılıklara sahiptir. Her birisi bu farklılıklardan dolayı avantajlara ve dezavantajlara sahiptir. Herhangi birisinin bir sorguya oluşturduğu cevap ile diğerlerinin oluşturduğu cevap farklı olmaktadır.

Günümüzde arama motorlarının başarılı bulunmasının birkaç nedeni vardır. Bunlardan bir tanesi, İnternet üzerindeki veri çok büyük boyutlarda ve çok dağınık şekilde bulunduğundan kullanıcıların arama motoru olmadan istediği herhangi bir verinin yerini belirleyip ulaşmaları mümkün değildir. Diğer bir neden ise, arama motorları kullanıcılara rahat ve hızlı bir şekilde veriye ulaşma imkânı sağlamaktadırlar. Ayrıca, kullanıcılar istedikleri yerden ve

istedikleri zaman veriye ulaşabilmektedirler. Daha önce erişmiş oldukları bir kaynak yeniden ihtiyaç duyduklarında çalışmıyor da olsa alternatif bilgi kaynakları arama motoru tarafından kendilerine sunulmaktadır. Kullanıcıların hemen hemen tamamı arama motorları çok etkin sonuçlar vermese bile Web'in kullanımını kolaylaştırdıklarını düşünmektedir.

Bir Web sayfasındaki bilgi ne kadar doğru ve önemli olursa olsun arama motorları tarafından kullanıcıya sunulmadığı sürece bir değeri bulunmamaktadır. Özellikle ticari içerikli Web sayfalarının arama motorlarının oluşturduğu listelerde kullanıcılara sunulması firmalar açısından çok büyük öneme sahiptir. Aynı şekilde kullanıcının da aradığı bilgiyi en doğru ve hızlı bir şekilde elde etmesi temel amacdır. Ancak, kullanıcıların hiçbirisi kendilerine sunulan listenin en iyi içeriğe sahip olup olmadığı konusunda değerlendirme yapabilecek bir bilgiye sahip değildir.

Herhangi bir arama motoru Web'in üçte birini ancak indeksleyebilmektedir ve farklı arama motorlarının indekslerinin çakışma oranı %1,4'ün altındadır (Bharat, 1998; Sander-Beuermann ve Schomburg, 1998). Web üzerinde indekslenbilir sayfa sayısı yaklaşık olarak 11,5 milyar sayfadır. Google, Yahoo, Bing ve Ask arama motorlarının indekslerinin %28,6 'sı kesişmektedir. Bu yüzden, yapılan herhangi bir aramada farklı arama motorlarından farklı sonuçlar alınması ve bu sonuçların içerisinde aynı olanların farklı rank değerlerine sahip olarak farklı sıralarda yer alması sıklıkla karşılaşılan bir durumdur. Herhangi iki arama motoru aynı veritabanını kullansa bile rank algoritmaları farklı olduğundan bir sorgu için oluşturulan sonuçlar farklı olmaktadır. İşte bu nedenlerden dolayı birden fazla arama motorunun sonuçlarının birleştirilmesi çok açıdan önemli faydalar sağlamaktadır.

Web sitesi tasarımcıları, hazırladıkları içeriklerin belirli anahtar kelimelere göre arama motorlarının listesinde bulunması ve hatta üst sıralarda yer almasını sağlayan yöntemler uygulamaktadırlar. Bu yöntemler arama motorlarının nasıl çalıştığını bilmeyi gerektirmektedir. Arama motorlarının çalışma şekline göre yapılan düzenlemeler arama motoru optimizasyonu (search engine optimization - SEO) olarak adlandırılmaktadır (Gözüdeli, 2009). Bu tür düzenlemeleri iyi bir şekilde uygulayan Web sitesi, arama yapılan sorguyla çok iyi derecede uyumlu olan bir Web sitesinin önüne geçmektedir ve kullanıcıya sunulmaktadır. Bu durumda kullanıcı kendisine çok daha faydalı bir içeriği elde edememiş olmaktadır (Linner, 2008). Arama motorları bu şekildeki yapay düzenlemelere karşı sürekli önlem almakta ve yöntem değişikliği yapmaktadırlar (Thurrow, 2003). Farklı arama

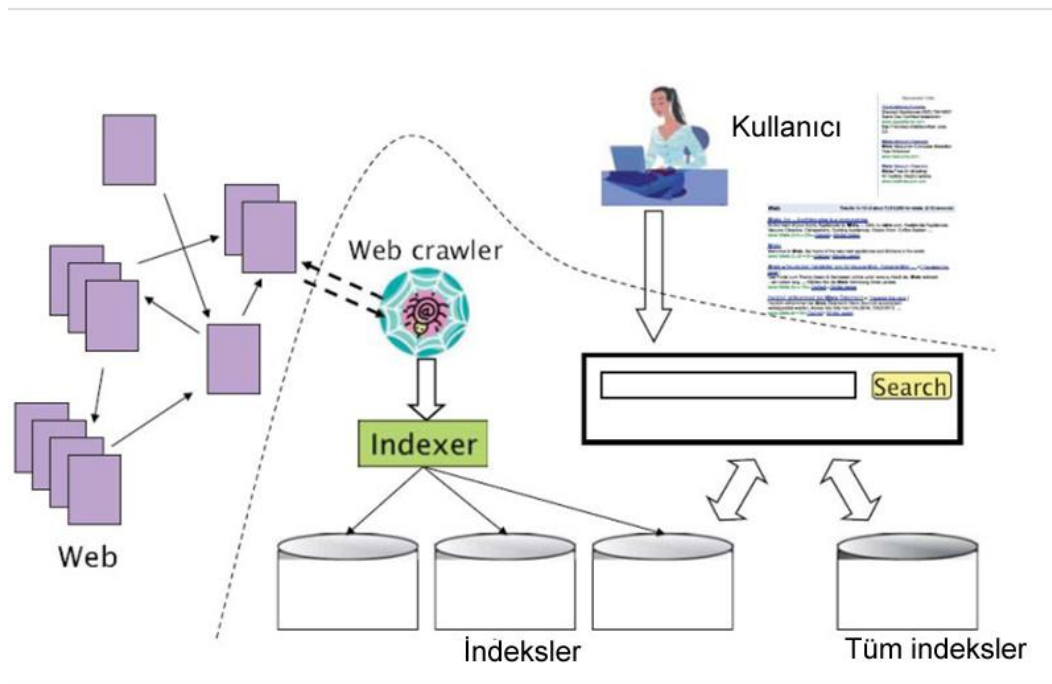
motorları farklı yöntemleri kullandığından bir arama için elde edilen sonuçlarda da farklılık olmaktadır.

Her arama motoru çok büyük boyutlardaki verinin tamamını kendi veri tabanlarında bulundurmaz ve arama işlemlerini, sakladığı indeks verileri üzerinde gerçekleştirir. Bir kullanıcı Web üzerinde bir arama yaptığına aslında İnternet üzerinde arama yapmaz arama motorunun Web üzerindeki bilgiyle oluşturduğu indeks üzerinde arama yapar. Farklı arama motorlarının elde ettiği sonuçlar bu yüzden farklılık göstermektedir. Bir arama motoru temel olarak üç kısımdan oluşur;

- Web Crawler
- İndeks oluşturucu
- Sorgu işleyici

Bu bileşenlerden, Web Crawler ve indeks oluşturucu, arka planda çalışırken, sorgu işleyici bileşeni sorgu ile eşzamanlı ve kullanıcı ile etkileşim halinde çalışır.

Genel olarak bir arama motorunun çalışmasını gösteren blok şema Şekil 2.1’de verilmiştir.



Şekil 2.1. Arama motoru bileşenleri

Web crawler veya Web spider, İnternet üzerindeki verileri otomatik olarak toplayan bir robot yazılımdır. Bu yazılımlar, sürekli Web sayfalarını bulundukları yerden alırlar, içeriğini incelerler, içeriğindeki kelimelere veya deyimlere göre bir veritabanında indekslerler. Verinin indekslenmesi sürecinde ise elde edilen sayfanın üzerinde ön işlemler yapılır. Bunlar; tekrarlı verilerin atılması veya içerik açısından anlamsal bir öneme sahip olmayanların atılması gibi işlemlerdir. Sorgu işleme aşamasında kullanıcıdan gelen sorguya cevap oluşturulur. Bunun için veritabanı üzerinde sorguyla ilgililik düzeyi en yüksek olandan en düşük olana doğru bir liste kullanıcıya sunulur.

Her arama motoru kullanıcı sorgusu için indekslenmiş dokümanlardan ilgili olanları elde ederken farklı yöntemler kullanırlar ve farklı kriterleri farklı ağırlıklarla kullanarak sorguyla ilgili dokümanların listesini oluştururlar. Bu algoritmalar sıralama algoritmaları (ranking algorithms) olarak adlandırılır ve SEO'nun başarısı için çok büyük öneme sahiptirler.

Arama motorlarının oluşturduğu listedeki sonuçlar iki gruba ayrılır. Birincisi, organik sonuçlar olup Web crawler tarafından bulunan sayfaları içerir. İkincisi ise reklam sonuçları olup belirli bir ücret karşılığı görüntülenen sonuçlardır. Reklam amaçlı görüntülenen sonuçlar genellikle kullanıcının aradığı içerikle çok yüksek düzeyde ilgili olan dokümanlar değildir.

Arama motorlarının sıralama sonuçları, kullanıcı sorgusuna bağımlı olan ve kullanıcı sorgusuna bağımlı olmayan faktörler olarak iki gruba ayrılır. Kullanıcı sorgusuna bağımlı olan sıralamada, kullanıcı sorgusundaki kelimelerin dokümanlarda bulunma sıklığı ve bulunduğu yer, sorgunun hangi dilde olduğu ve sorgunun yapıldığı yere coğrafik olarak yakınlığı göz önüne alınır. Kullanıcı sorgusuna bağımlı olmayanlar ise, sayfasını kalitesi, hub veya otorite sayfa olup olmadığı gibi faktörlerdir.

Kullanıcı sorgusuna bağımlı olmayan sıralama, dokümanlardaki link analizine dayanır. Bunlar, dokümana arama motoru tarafından atanan kalite değerini belirler. Kullanıcı aramak istediği bilgiyi sorgu içerisinde ne kadar iyi ifade ederse etsin sorguya bağlı olmayan diğer faktörlerde elde edeceği sonuçları etkilemektedir. Bir Web dokümanına bir arama motoru ilgililik düzeyi yüksek değer hesaplayarak üst sıralarda yer verirken diğer bir arama motoru ilgililik düzeyi düşük hesaplayarak alt sıralarda yer vermektedir.

Yapılan araştırmalara göre kullanıcıların İnternet kullanım amaçları ve arama motoru sonuçlarına yönelik davranışları aşağıdaki gibidir (Goodwin, 2017):

- Kullanıcıların %75'i sorgu sonucunda gelen ilk sayfadaki sonuçlara bakmaktadır. Sadece %25'i ikinci sayfaya geçmektedir.
- Kullanıcıların %70'i doğrudan organik sonuçları seçmektedir ve reklam içerikli sonuçlara bakmamaktadır.
- İnternet üzerinde en sık yapılan işlemler arama motoru üzerinden sorgu yazarak arama ve e-posta göndermektir.
- Şirketlerin %81'i kendileri için yazılan blog bilgilerini dikkate almaktadırlar.
- Google arama motorunun sunduğu organik listedeki sonuçlardan 1.sıradakini seçen kullanıcı oranı %18, 2.sıradakini seçen %10 ve 3.sıradakini seçen %7 oranındadır.
- Aynı araştırma Bing için de yapılmıştır. Bing arama motoru organik sonuçlarından 1.sıradakini seçen kullanıcı oranı %9,7, ikinci sıradakini seçen %5,5 ve 3.sıradakini seçen %2,7 oranındadır.

Yukarıdaki sonuçlara göre meta arama motorlarının farklı arama motorlarından gelen sonuçları tek bir liste halinde birleştirirken ilk sayfaya ve ilk sıralara en çok ilgili olanları seçmesi ve kullanıcıya sunulan listede ilk sıralarda sorguya en uygun sonuçların sunulması gerekmektedir.

Yine yukarıda verilen bilgilerin sonucunda, bir kullanıcının oluşturduğu sorguya her arama motorunun farklı sonuçlar getirmesinden dolayı elde edilen listenin en iyi ve tek sonuç olmayacağı açıktır. Ayrıca, arama motorlarının elde ettikleri listelerde aynı sonuçların farklı formatlarda bile olsa yer alabileceği de görülmektedir. Verilen bir sorgu için farklı arama motorlarından elde edilen sonuçları alıp bunları birleştirerek tek bir listeyi kullanıcıya sunmanın daha etkin bir çözüm olacağı açıkça görülmektedir.

Meta arama motorları, birden fazla arama motoru kullanarak arama işlemi gerçekleştirir. Bunun sonucunda, kullanıcıya birleştirilmiş tek liste sunulmakta, böylece kullanıcının hızlı bir şekilde ilgililik düzeyi yüksek dokümanlara ulaşması sağlanmaktadır. Bu şekilde yapılan aramada, kullanıcı aynı sorguyu birden fazla arama motoruna tekrar tekrar girmek zorunda

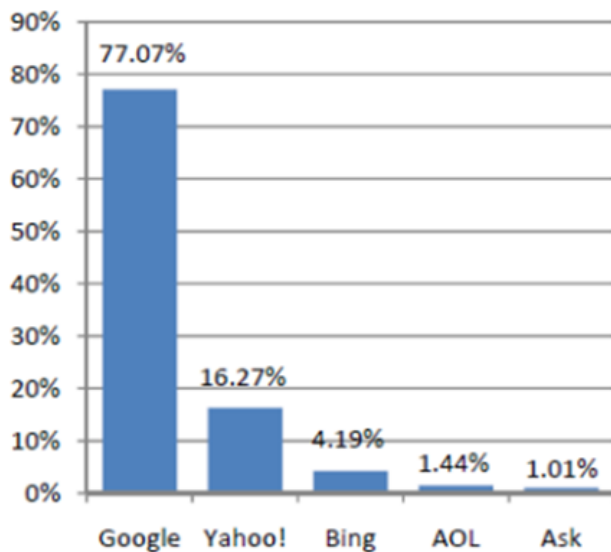
kalmamakta ve farklı arama motorlarının sonuçlarına ayrı ayrı bakmak yerine birleştirilmiş tek bir sonuca bakmaktadır.

Meta arama motorları verilen sorguyu eş zamanlı bir şekilde farklı arama motorlarına gönderir ve kendilerine ait crawler'a veya indekse sahip değildir. Ancak, farklı arama motorlarından gelen sonuçları filtrelerler ve tekrarlı olanları silmek gibi işlemleri yaparlar. Ayrıca, gelen sonuçları kullandıkları rank algoritmasıyla yeniden ilgililik düzeyine göre sıralanmış bir şekilde kullanıcıya sunarlar.

Kullanıcılar bir bilgiyi arayacakları zaman arama motorlarından birisini seçerler ve sorgularını bu arama motoruna girerler. Kullanıcılar arama motorlarını seçerlerken üç özelliğe önem verirler. Bunlar,

- ilgili sonuçlar vermesi
- kolay okunabilir arayüze sahip olması
- aramayı genişletip daraltabilecek seçenekler sunması

olarak sıralanmaktadır ("İnternet: The Best Search Engines of 2019", 2019). Günümüzde, Google, Yahoo, Bing ve Ask en yaygın kullanılan arama motorlarıdır. Aşağıdaki grafikte en iyi arama motorlarına yönelik kullanıcılar üzerinde yapılan araştırma sonuçları verilmiştir ("İnternet: Best search engine in the world", 2010).



Şekil 2.2. En iyi arama motorlarına yönelik araştırma sonuçları

Arama motorlarının bir sorguya karşılık oluşturduğu sonuç listesi, kullandıkları veritabanı boyutuna, sonuçların güncelliğine, sorgu üzerinde arama motoru tarafından yapılabilecek işlem kapasitesine ve teknolojisine bağlı olarak değişir.

2.1. Veri Tabanı Boyutu

Arama motorunun veritabanı boyutunu dışarıdan yapılan işlemlerle tam doğru olarak bilmek mümkün değildir. Bunun yerine tahmin edilmeye çalışılır. Herhangi bir dilde hemen hemen her dokümanda geçen bir kelimeyi aradığımızda dönen sonuç sayısı önemli bir göstergedir. Yapılan çalışmalarda, İngilizce “the” kelimesi Google, Yahoo ve Bing arama motorlarında aratıldığında sırasıyla 12 milyar, 9 milyar ve 0,9 milyar sonuç listelenmiştir (Lewandowski, Wahlig ve Meyer-Bautor, 2006). Tabii ki elde edilen sonuçlar veri tabanının tam boyutunu göstermez. Ayrıca, bir arama motorunun veri tabanı boyutu arama motorunun kalitesini göstermemektedir. Gelen sonuçların sorguya ilgililik düzeyi daha önemlidir.

2.2. Güncellik

Web sayfalarının içerikleri değiştikçe crawler yazılımlarının yeni içerikleri alıp indekslemesi gerekmektedir. Yapılan araştırmalarda farklı arama motorlarının Web sayfalarını güncelleme sıklıklarının farklı olduğu görülmüştür. En sık güncelleme işleminin Google arama motorunun spider yazılımı olan Googlebot tarafından günlük olarak yapıldığı görülmüştür (Lewandowski ve diğerleri, 2006).

Arama motorları üzerinden elde edilen sonuçların bir kısmının indekslendikten sonra değişmiş olma olasılığı bulunmaktadır. Bu yüzden farklı arama motorlarını kullanarak sorgu yapan meta arama motorları aynı Web sayfasının iki versiyonunu bulunduran sonuç listesi de oluşturabilmektedir. Ayrıca, güncellenmiş versiyondaki Web sayfasına yeni alanlar eklenmiş veya var olanlar çıkarılmış olabilmektedir. Bu tür değişikliklerin de meta arama motorları tarafından birleştirilmiş sonuçlarda göz önüne alınması gereklidir.

2.3. İşlem Kapasitesi

Kullanıcıların büyük çoğunluğu sadece aradığı bilgiyi içeren kelimeleri sorgu içerisinde kullanır ve arama motorunun sunduğu sonuçları inceler. Ancak, günümüz arama motorları

AND, OR, NOT gibi mantıksal operatörler, deyim tanımlama, doküman türü tanımlama veya bir alan adı üzerinde aramayı sınırlandırma gibi özellikleri de kullanıcılara sunmaktadır. Tüm arama motorları bu tür özellikleri farklı oranlarda desteklemektedirler. Bazılarında AND operatörü kullanılırken bazılarında aynı iş için “ ” işaretleri kullanılmaktadır. Bu yüzden meta arama motorunun birden fazla arama motorunu kullanarak elde ettiği sonuçlar farklı olabilmektedir.

Her arama motorunun desteklediği dil sayısı da farklıdır. Örneğin, Google 46 dili desteklerken, Yahoo 32 dili desteklemektedir. Bing ise 41 dili desteklemektedir. Bu yüzden her arama motorunda ayrı ayrı arama yapıp sonuçların incelenmesi gereklidir. Meta arama motoru tarafından yapılan sorguya da her arama motoru farklı sonuçlar getirecektir ve bu sonuçlar birleştirilmelidir.

2.4. Teknoloji

Arama motorlarının iki önemli özelliği kullanıcılar için çok önemlidir. Bunlar, hız ve gelen sonuçlardaki ilgililik düzeyidir. Yapılan araştırma çalışmalarında arama motorlarının cevap hızı düştükçe kullanıcıların başka arama motorlarına yöneldikleri görülmüştür. Yine aynı şekilde arama motorunun sunduğu sonuçların ilgililik düzeyi düştükçe yine kullanıcıların başka arama motorlarına yöneldikleri görülmüştür. Bu yüzden meta arama motorlarının eş zamanlı sorguyu göndermek için kullanacağı arama motorlarının seçimi de oldukça önemlidir.

Her arama motoru elde ettiği sonuçları ilgililik düzeyine göre bir rank algoritmasıyla sıralayarak kullanıcıya sunar. Arama motorları dokümanları, kalitesine ve kullanıcı sorgusuna ilgililik düzeyine göre rank değeri yüksek olandan düşük olana doğru sıralarlar. Meta arama motorlarının farklı rank değerlerine sahip aynı Web sayfasını da belirleyerek yeniden bir rank değeri hesaplaması gerekmektedir. Bu tez kapsamında, geliştirilen meta arama motoru farklı arama motorlarından gelen sonuçlar için yeniden rank değeri hesaplayan bir algoritma geliştirilmiştir.

2.5. Meta-Arama Motorlarına Duyulan İhtiyaç

En yaygın kullanılan üç arama motorunun anahtar özellikleri Çizelge 2.2’de verilmiştir.

Çizelge 2.2. Google, Yahoo ve Bing arama motorlarının özellikleri

	Google	Yahoo!	Bing
İndeks Boyutu	Rank1	Rank2	Rank3
Örümcek Adı/ Güncellik	Googlebot Günlük Güncelleme	Yahoo Slurp Belirgin olmayan sıklığı	MSNBot Sık güncelleme
Yetenekler			
Arama Operatör Gelişmiş Arama	Mantıksal Operatör - Hariç + dahil "" ile tam uyum arama Joker karakter Kök arama Meta kelime arama Detaylı arama Formu 46 Dil desteği	Mantıksal Operatör - Hariç + dahil "" ile tam uyum arama Yahoo Kısayol Joker karakter Kök arama Detaylı arama Formu 32 Dil desteği	Mantıksal Operatör - Hariç + dahil "" ile tam uyum arama Hızlı Cevap seçeneği Kök arama Detaylı arama Formu 42 Dil desteği
Teknoloji			
Hız	Çok Hızlı (anında cevap)	Tam olarak bilinmiyor	Tam olarak bilinmiyor
Sıralama	PageRank, hipertext eşleme analizi	Anahtar kelime tıklama popülerliği	Otomatik, anahtar kelime vurgulama, dahili bağlantı

Yukarıda sunulan bilgilerden görüldüğü gibi her arama motorunun kullandığı yöntemlerden kaynaklanan avantajları ve dezavantajları bulunmaktadır. Kullanıcının her arama motoruna ayrı ayrı sorgu girmesi ve sonuçlarını incelemesi hem zaman hem de maliyet açısından iyi bir yöntem değildir. Bunun yerine, bir meta arama motoru arayüzü kullanarak farklı arama motorlarının birleştirilmiş bir şekilde elde edilmesi kullanıcı açısından çok büyük kolaylık sağlamaktadır. Ancak, bu şekilde farklı arama motorlarından alınan sonuçların birleştirilmesinde önemli zorluklar ve problemler bulunmaktadır.

Arama motorlarından alınan sonuçların çakışması önemli sorunlardan bir tanesidir. Meta arama motorunun gelen sonuçlardaki çakışmaları belirleyerek ortadan kaldırması gerekmektedir. Diğer bir önemli sorun ise farklı arama motorlarından alınan sonuçlarda aynı

Web sayfasının çok farklı rank değerine sahip olması ve elde edilen listelerde çok farklı sıralarda yer almasıdır. Bu durumda farklı arama motorlarından alınan aynı Web sayfalarından hangisinin rank değerinin alınacağı da önemli bir konudur.

Yapılan araştırma çalışmalarında, 19000'den fazla kullanıcının arama motorlarından elde edilen sonuçlar üzerindeki davranışı incelenmiştir. Yapılan inceleme sonuçlarına göre kullanıcıların nadiren (yaklaşık olarak %25 oranında) ikinci sayfa sonuçlarına geçtiği görülmüştür (Lewandowski ve diğerleri, 2006). Bu yüzden meta arama motorlarının farklı arama motorlarından gelen sonuçlardan ilk sayfada yer alanlar üzerinde birleştirme, çift olanları çıkarma gibi işlemleri yapması daha maliyet etkin bir çözüm olarak görülmektedir. Bu tez kapsamında geliştirilen meta arama motoru ağırlıklı olarak ilk sayfalarda yer alan sonuçları dikkate almaktadır.

Yine arama motorlarının elde ettiği ilk sayfa sonuçları üzerinde yapılan araştırmalarda, bir arama motorunun elde ettiği ilk sayfa sonuçlarında tekil olanların oranı %88,3 olarak çıkmıştır. İki farklı arama motorunun ilk sayfa sonuçlarındaki çakışma oranı %8,9, üç arama motorunun ilk sayfa sonuçları arasında çakışma oranı %2,2 ve dört arama motorunun ilk sayfa sonuçları arasındaki çakışma oranı ise %0,6 olarak çıkmıştır ("İnternet: Dogpile.com", 2010). Buradan da görüldüğü gibi meta arama motorunun kullandığı arama motorlarının Web sayfalarına atadığı rank değerleri çok farklıdır ve elde edilen sonuçlarda çok farklı çıkmaktadır.

Bunlara ek olarak, bir kullanıcı sadece Google arama motorunu kullanırsa sorgusuyla ilgili yaklaşık olarak Web'teki %72,7 en iyi sonucu ilk sayfada görememektedir. Eğer kullanıcı sadece Yahoo arama motorunu kullanırsa %69,2 ve sadece Live (Bing) kullanırsa %69,9 en iyi sonucu ilk sayfada görememektedir ("İnternet: Different Engines, Different Results: Web Searchers Not Always Finding What They're Looking for Online", 2010).

Çizelge 2.3'de Google, Yahoo, Live ve Ask arama motorlarının tüm sonuçları üzerinde çakışma oranları verilmiştir.

Çizelge 2.3. Google, Yahoo, Live ve Ask arama sonuçlarının çakışma oranları

	Çakışma Oranları
Tüm 4 arama motorları ortak	% 0,6
3 Arama motoru ortak	%2,2
Herhangi 2 arama motorunda ortak	%8,9
Her bir arama motorunda tekil	%88,3

Meta arama motorlarının hangi arama motorları üzerinden sonuçlar alacağı da oldukça önemlidir (Mohamed, 2005). Meta arama motorları farklı arama motorlarından gelen sonuçları birleştirirken aynı Web dokümanındaki ayrı veri parçalarının farklı arama motoru sonuçlarında geldiğini algılayabilmeli ve bunları da birleştirebilmelidir. Örneğin, aynı başlangıç, aynı hedef, aynı gün ve aynı saate sahip bir sonuç için farklı arama motorlarından birisinde hareket noktası olan ve diğerinde varış noktası olan başka bir sonuç birleştirilebilmelidir. Bu tez kapsamında geliştirilen meta arama motorunun aynı anlama sahip farklı kelimelerle ifade edilen sonuçları da birleştirmesi sağlanmıştır.

İyi bir arama motoru araştırmamanın amacını tam olarak anlayabilmeli ve buna yönelik gerek duyulan tam uyumlu dokümanları sağlayabilmelidir. Bu tez kapsamında geliştirilen meta arama motoru, anlamsal eşleştirme yaparak dokümanları birleştirerek elde edilen sonuçları anlamsal olarak sorguyla ilgililik düzeyine göre listeleyebilmektedir. Günümüzdeki bazı meta arama motorları belirli birkaç tane arama motoruyla birlikte çalışırken bazıları binden fazla arama motoruyla birlikte çalışmaktadırlar (Renda ve Straccia, 2003).

Meta arama motorlarının en büyük faydası birden fazla arama motoruyla çalıştığından dolayı kapsadığı alan çok fazla olmasıdır. Meta arama motorlarının sağladığı faydalar aşağıdaki gibi sıralanabilir:

- Web üzerindeki bilgiler çok büyük boyuttadır ve meta arama motorları normal arama motorlarından daha büyük bir kısımda arama yapar.
- Kullanıcıların farklı arama motorlarına sorgu girerek gelen sonuçları incelemeleri yerine tek bir sorgu ile birleştirilmiş tek bir liste halindeki sonuçları almalarını sağlar.
- Meta arama motorları eş zamanlı birden fazla arama motoru kullandığından Dünya üzerinde çok daha geniş bir alanda daha etkin ve hızlı arama sağlar.

- Meta arama motorları kullanıcının girdiği sorguya hangi arama motorunun daha uygun olduğuna çok hızlı ve doğru bir şekilde karar verebilir.

Meta arama motorları farklı türlerde geliştirilmiş olabilir. Bunlar yaptıkları işleve göre aşağıdaki şekilde gruplandırılır:

- Sadece farklı arama motorlarından gelen sonuçları sunarlar. Arama motorlarından gelen listeler üzerinde herhangi bir işlem yapmazlar.
- Çok sayıda arama motorunu kullanarak sorgu yaparlar, gelen sonuçları birleştirerek sunarlar. Bu meta arama motorları gelen listelerdeki tekrarlı sonuçları silme ve elde edilen listedeki sonuçlar için yeniden rank değeri hesaplama gibi işlemleri yapar.

Meta arama motorlarının temel amaçları daha derin arama yapmaktır. Derin İnternet'teki bilgilere ulaşmayı amaçlarlar.

3. META ARAMA MOTORLARI

Meta arama motorları klasik arama motorları gibi kullanıcıya basit bir arayüz sağlarlar. Kullanıcı sorgusunu arayüz üzerinden girer ve sonuçlarını arayüz üzerinden almalarına olanak sağlarlar. Sırasıyla, kullanıcıdan sorgunun alınması, sorgunun seçilen arama motorlarına gönderilmesi, gelen sonuçların birleştirilmesi, birleştirilen sonuçlar için yeni rank değerlerinin hesaplanması ve elde edilen sonuçların kullanıcıya sunulması işlemleri meta arama motorları tarafından gerçekleştirilir.

Meta arama motorları 1997 (Selberg ve Etzioni, 1997) yılından bu yana üstünde çalışıla gelen bir alandır. SavvySearch (Adele E. ve Dreilinger, 1997) ilk öğrenebilen Meta arama motorudur. Daha sonra geliştirilen meta arama motoru modelleri Helios (Gulli ve Signorini, 2005) ve Tadpole (Mahathi S. ve Singitham, 2002) mimarileridir. Helios açık kaynak bir meta arama motoru mimarisi olup Pisa Üniversitesi bünyesinde geliştirilmiştir. Helios, Web dokümanlarından sonuç çıkarımı süreçlerini eş zamanlı çoklayarak çalışma süresini indirgeyen bir mimari sunmaktadır.

3.1. Meta Arama Motorlarının Mimarisi

Meta arama motorlarının en büyük faydası birden fazla arama motoruyla çalıştığından, kapsadığı alanın çok fazla olmasıdır. Meta arama motorlarının sağladığı faydalar aşağıdaki gibi sıralanabilir:

- Web üzerindeki bilgiler çok büyük boyuttadır ve meta arama motorları normal arama motorlarından daha büyük bir kısmında arama yapar.
- Kullanıcıların farklı arama motorlarına sorgu girerek gelen sonuçları incelemeleri yerine tek bir sorgu ile birleştirilmiş tek bir liste halindeki sonuçları almaları sağlanır.
- Meta arama motorları eş zamanlı birden fazla arama motoru kullandığından Dünya üzerinde çok daha geniş bir alanda daha etkin ve hızlı arama sağlarlar.
- Meta arama motorları kullanıcının girdiği sorguya hangi arama motorunun daha uygun olduğuna çok hızlı ve doğru bir şekilde karar verebilirler.

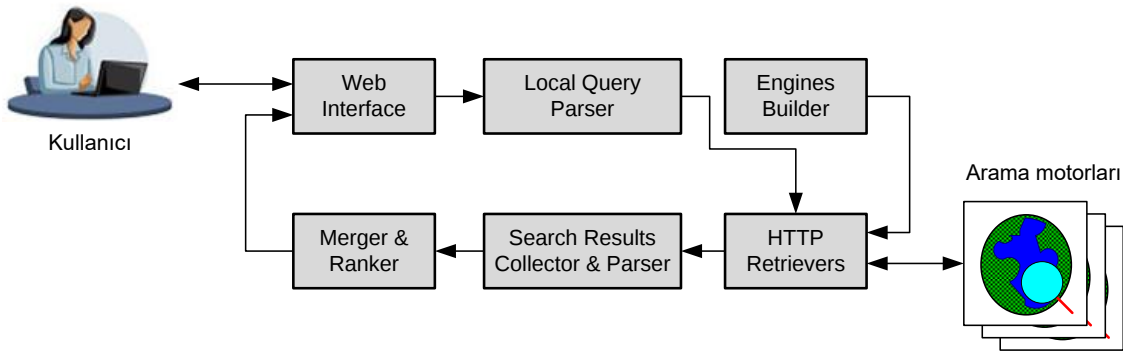
Meta arama motorları farklı türlerde geliştirilmiş olabilir. Bunlar yaptıkları işleve göre aşağıdaki şekilde gruplandırılırlar:

- Sadece farklı arama motorlarından gelen sonuçları sunarlar. Arama motorlarından gelen listeler üzerinde herhangi bir işlem yapmazlar.
- Çok sayıda arama motorunu kullanarak sorgu yaparlar, gelen sonuçları birleştirerek sunarlar. Meta arama motorları gelen listelerdeki tekrarlı sonuçları silme ve elde edilen listedeki sonuçlar için yeniden rank değeri hesaplama gibi işlemleri yaparlar.
- Temel amaçları daha derin arama yapmaktır. Derin İnternet'teki bilgilere ulaşmayı amaçlarlar.

Günümüzde en yaygın Helios, TadPole ve TreeMap meta arama mimarileri kullanılmaktadır.

3.1.1. Helios mimarisi

Helios, kişiselleştirilebilir bir meta-arama motoru mimarisi sunar. Meta-arama motoruna ilişkin mimari, Şekil 3.1.'de gösterilmiştir.



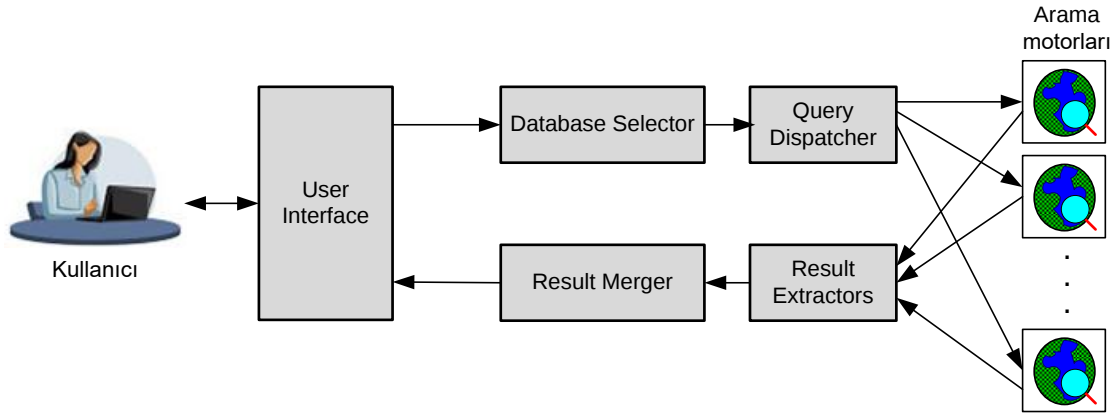
Şekil 3.1. Helios mimarisi

Helios mimarisinde, kullanıcılara sorgularını göndermek ve arama motoru seçmek için bir arayüz sunulur. Local Query Parser ve Emitter tarafından kullanıcı sorgusu seçilen arama motorlarına göre yeniden düzenlenir. Engines Builder uzaktaki arama motorlarına bağlantı için gerekli ayarları bulundurur. HTTP Retriever modülü ağ bağlantılarını gerçekleştirir. Sonuçlar arama motorlarından geldiğinde Searcher Results Collector & Parser tarafından

ilgili bilgiler XML biçiminde elde edilir. Kullanıcı isterse kendi seçimlerini Merger & Ranker birimiyle arama sonuçlarına yansıtabilmektedir. Burada farklı arama motorlarından gelen sonuçlardaki alanların birleştirilmesi gibi işlemler tanımlanabilmektedir.

3.1.2. TadPole mimarisi

TadPole mimarisini Helios meta arama motorlarından ayıran en belirgin özellik, paralel prosesler ile eş zamanlı arama motorlarına bağlantı sağlayabilmesidir. TadPole mimarisine sahip bir meta arama motoru Şekil 3.3.'de gösterilmiştir.



Şekil 3.2. TadPole meta arama mimarisi

TadPole mimarisinde de kullanıcı sorgusunun girilmesi için bir arayüz sunulur. Girilen sorgudan sonra thread'ler oluşturulur ve farklı arama motorlarına sorgu paralel bir şekilde gönderilir. Her bir arama sonucu verilen süre içerisinde geri alınır gelmeyen sorgu iptal edilir. Farklı arama motorlarından gelen sonuçlar birleştirilir, tekrar sıralanır ve kullanıcıya sunulur.

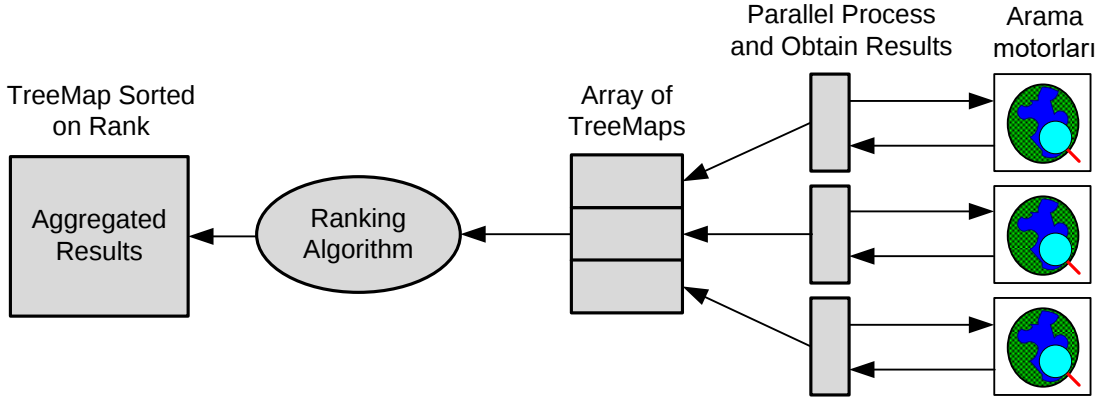
3.1.3. TreeMap mimarisi

TreeMap mimarisi, TadPole mimarisine göre rank işlemleri ve isteğe göre seçilen arama motorlarına olanak sağlaması ile daha gelişmiştir.

TreeMap mimarisinde kullanıcıdan alınan sorgu eş zamanlı bir şekilde seçilen arama motorlarına gönderilir. Her arama motoru için sorgu yeniden düzenlenir. Gelen sonuç sayfaları parse edilir ve başlıkları, tanım bilgileri, URL adresleri, rank değerleri TreeMap

veri yapısı içine aktarılır. Ardından TreeMap rank algoritması tarafından yeniden sıralanarak kullanıcıya sunulur.

TreeMap mimarisi Şekil 3.5.'de verilmiştir.



Şekil 3.3. TreeMap mimarisine sahip meta arama motoru

3.2. Meta Arama Motorlarının Süreç ve Bileşenleri

Genel olarak mimarisi ele alınan arama motorlarında verinin eşleştirme, dönüşüm ve birleştirme süreçlerini her bir aşama için ayrı ayrı aşağıdaki şekilde incelenmiştir:

3.2.1. Web arama arayüz çıkarımı

Bir meta-arama motoru, birçok yatay ve dikey arama motorunda arama yapıp sonuç birleştireceğinden öncelikle sorgu yapılan arayüzleri tanımlaması gerekir. Tanımlanmış arayüzler, sorguların farklı veri kaynaklarına ulaştırılabilmesi için bir zorunluluktur. Bu süreç hakkında yapılmış çalışmalar ışığında aşağıdaki şekilde özetlenebilir:

Öznetelik çıkarımı

Bir meta-arama arayüzü için ilk işlem, arama arayüzü sağlayan arama motorlarının arayüzlerinin şema çıkarımının yapılmasıdır. Arayüzleri çıkarıldıktan ve şema eşlemesi yapıldıktan sonra ancak arama motorları ile başarılı şekilde sorgulanabilir bir bağlantı sağlanması mümkün hale gelir. Sorgu arayüzleri için geliştirilen şema çıkarım algoritmaları, sorgu arayüz şemasının nasıl gösterildiği ile doğrudan ilişkilidir. Genel olarak çalışmalarda

tercih edilen iki farklı şema çıkarım modeli mevcuttur: Düz model ve hiyerarşik model. Düz model olarak yapılan çalışmalara örnek olarak WISE-3 verilebilir.

WISE 3 level modeli

Bu modelde her bir sorgu arayüzü üç seviyede gösterilir:

- Site seviyesi: temel site bilgilerini içerir. (Arama motorunun URL bilgisi, domaini, yatay-dikey olması gibi) Ayrıca, arama arayüzündeki özniteliklerin listesi de bu seviyededir.
- Öznitelik seviyesi: Her bir özniteliğin adı, sırası konumu, etiketi gibi bilgilerinin yer aldığı seviyedir. WISE-i Extractor aracı bu türden bilgilerin çıkarımı için geliştirilmiş bir araçtır. Araç, her bir özniteliğe ait elemanı otomatik gruplayabilmektedir.
- Eleman seviyesi: arayüzde bulunan her bir HTML elemanı hakkındaki bilgilerin yer aldığı seviyedir (He, Meng, Lu, Yu ve Wu, 2007). Benzer bir çalışma olarak geçen model de düz modele örnek olarak verilebilir (Zhang, He ve Chang, 2004). Ancak bu model WISE 3' e göre daha düşük veri elde edebilmektedir.

Hiyerarşik model

Bu modelde her bir en alt düğüm, sorgu arayüzündeki bir eleman ile bağlantılıdır. Her bir ara düğüm de sorgu arayüzündeki elemanlar veya eleman grupları ile bağlantılıdır. Aynı ataya bağlı düğümlere kardeş düğümler denir. Kardeş düğümler, arayüzde karşılıkları olan elemanların dizilişlerine göre sıralı varsayırlılar. Bu model farklı seviyedeki eleman gruplarına ait sıralı semantik bilgileri çıkarılabilmektedir (W. Wu, Yu, Doan ve Meng, 2004). Benzer şekilde sorgu arayüzünü hiyerarşik modelde çıkartan ve sunan bir araç çalışması yapılmıştır (Dragut, Kabisch, Yu ve Leser, 2009).

Öznitelik analizi

Öznitelik analizi esnasında, elemanlara ait ilişki tipi-domain tipi-default değer varsa default değeri- değer tipi ve birimi gibi bilgiler toplanır. Bir öznitelik birden fazla alt elemandan oluşuyorsa domainin semantiği ve alt elemanların ilişkisinin çıkartılması gerekir. İlişki, grup

tipi, aralık tipi veya parçalı tipte olabilir. Domain tipi de benzer şekilde aralık-sınırsız-mantıksal veya sonlu olabilir (He, Meng, Yu ve Wu, 2004).

3.2.2. Şema öznitelik/eleman eşlemesi

Öznitelik eşlemesi, şema eşlemenin en temel iş bileşenlerindendir. Uzun yıllar üstünde çalışılmıştır. Öznitelik eşleştirme konusunda (Rahm ve Bernstein, 2001) yapılan çalışmaları özetler. Farklı iki arama arayüzünde bulunan A1 ve A2 özniteliklerinin eşlenmesini ele alalım. Farklı bilgiler esas alındığında farklı şekillerde eşleştirilebilirler.

- **Sözlük seviye bilgi:** Bu kategoriye ait bir sözlük üstünden eşleştirilebilen özniteliklerin eşleştirilmesi şeklindedir. Bir sözlük eşliğinde kullanılabilir. Örneğin WordNet (Miller, 1995) bu türden bir sözlük olarak kullanılabilir. İki farklı yerel arayüzde birbirinden okunmuş olarak farklı ama sözlüğe göre anlamları yakın ya da ortak olan iki öznitelik-eleman eşlenmiş varsayılır.
- **Şema Seviye Bilgi:** Bu türden bir veri kullanıldığında şemada yer alan öznitelik-elemanlara ait, veri tipi, değer tipi, tekil bir ayrıştırıcı olup olmadığı gibi meta-verilerden faydalanılır. Birbiri ile uyumlu olan veya eşit görünen Öznitelik/eleman'lar eşleştirilmeye çalışılır.
- **Eleman Seviye Bilgi:** Eleman seviye bilgide, bir öznitelik-eleman'a ait anlık olarak içerdiği değerden yola çıkarak eşdeğeri kestirilmeye çalışılır. Radyo buton, checkbox, selection list gibi HTML elemanlarının içerdiği veriler bu seviyede başarılı bir eşleştirme için oldukça kullanışlıdır (He ve diğerleri, 2007).

İki farklı öznitelik-eleman verildiğinde bu iki öznitelik eşitlenip eşitlenmediği konusunda fikir vermek üzere kullanabilecek iki temel yaklaşım vardır (Rahm ve Bernstein, 2001):

- **Hibrit yöntem:** Yukarıdaki yöntemleri bir arada ve amaca uygun olarak kullanan yöntemdir.
- **Bütünleşik (Composite) Yöntem:** Yukarıdaki yöntemlerin tamamını deneyerek ağırlıklı toplam gibi bir yöntem ile bütün ara sonuçları hesaba katarak iki özneliğin eşleşip eşleşmediğine karar veren yöntemdir.

Çoğu eleman-öznitelik eşleştirme tekniği, 1-1 bir eşleştirme varsayımına dayanır. Yani birinci tarafa ait bir adet öznitelik ikinci tarafta bir tek öznitelik ile eşleşiyor varsayımıyla çalışır. Ancak bazı durumlarda bir elemana karşılık çok eleman gelebilir. Bu durum yapılan çalışmada iki farklı şekilde ifade edilmektedir (W. Wu ve diğerleri, 2004): Grupsal (aggregate) tipler ve as-üst tipler (is-a). Örneğin tarih bilgisi, yıl-ay-gün bilgilerinin birleşmesinden oluşurken; toplam yolcu sayısı yetişkin ve çocuk yolcu sayısının toplamına eşit olabilir. Bu durumda, çocuk ya da yetişkin elemanları birer yolcu elemanı türünden as elemanlardır. İlgili çalışmada ayrıca, 1-m ilişki ters yönde de desteklenmektedir.

Pozitif Eşleme Temelli Kümeleme

Pozitif eşleme temelli kümeleme yaklaşımında, öznitelikler arasındaki eşleşmeler üstünden kümeleme yapılır. Pozitif eşleme temelli kümelemede, iki öznitelik değerlerinin gerçek bir eşleştirmeye (gerçek kelime eşleşmesi, yaklaşık string temelli eşleşme, synonymy, hypernymy) uygun olduğu varsayılır. Öncelikle eşik değer kullanılır. Arkasından ön öznitelik eşleştirmesi yapılarak öznitelikler kümelendirilir. Bu aşamada kümeye bir temsili öznitelik adı (representative attribute name-RAN) atanır. Bir kümeye ait RAN değeri Global öznitelik adı için adaydır.

Kestirimsel eşleme temelli kümeleme

Şayet iki öznitelik arasında yaklaşık isim eşleşmesi, isimler arasında kosinüs benzerliği, domain tipi eşleşmesi, değer eşleşmesi, birim eşleşmesi, default değer eşleşmesi ve değer-desen eşleşmesi varsa kestirimsel eşleme temelli kümeleme ile bu iki öznitelik eşleştirilebilir. Örneğin, “Yayın Tarihi” ile “Basım Yılı” öznitelik isimleri birbirinden oldukça farklı isimler olduğundan pozitif eşleme temelli kümeleme ile eşleştirilebilmeleri mümkün değildir. Bu durumda, kestirim eşlemesi temelli kümeleme ve meta-bilgi ile bu iki öznitelik eşleştirilebilir.

Başlangıçta hiç küme yoktur. İlk arayüz ele alındığında, her bir A_i özneliği için Kümeler ardışık olarak oluşturulur. Eşlemeler öznitelik-küme dağılımı şeklindedir. Öncelikle bir özneliğin herhangi bir var olan öznitelik-küme dağılımına eklenip eklenmediğine bakılır. Şayet bir eşleştirmeye rastlanırsa yerel arayüzde bulunan A_i özneliği var olan bu kümeye dâhil edilir. Şayet rastlanmazsa RAN (A_i) için eşleştirilebilir durumda bir küme olup

olmadığı kontrol edilir. Şayet uygun bir küme bulunursa, A_i bu kümeyle eşleştirilir. Bu yeni eşleştirme öznitelik-küme dağılımına eklenir. İkinci arama da başarısız olursa, A_i var olan kümelerden (C_k) ağırlığı en yüksek olana eklenir.

A_i ve A_j öznitelikleri için 7 farklı ölçü kullanılarak ağırlık hesabı eşitlik (3.1) 'e göre hesaplanır:

$$W(A_i, A_j) = W_{am} + W_{vss} + W_{cd} + W_{vtm} + W_{cu} + W_{dv} + W_{vp} \quad (3.1)$$

- W_{am} : yaklaşık string ve alt string eşlenebilirliği
- W_{vss} : cosinüs benzerlik eşlenebilirliği
- W_{cd} : domain eşlenebilirliği
- W_{ctm} : Değer tipi eşlenebilirliği
- W_{cu} : birim eşlenebilirliği
- W_{dv} : default değer eşlenebilirliği
- W_{vp} : değer desen eşlenebilirliği

Sonuç olarak arayüz şema eşleme fazında özniteliklerin eşlenmesi aşamasında farklı arayüzlerden elde edilen özniteliklerin semantik olarak kümelenmesi yaklaşımı kullanılır.

3.2.3. Global öznitelik analizi

Her bir arama arayüzünün analizi ve arama arayüzleri için öznitelik eşlemesi yaptıktan sonra global bir sorgu arayüzünün oluşturulması süreci başlar. Bu süreçte

- Global öznitelik adı
- Global öznitelik domain tipi
- Global öznitelik değer tipi

karakteristiklerinin belirlenmesi sürecidir.

Öznitelik eşleme süreçlerinde (pozitif eşitleme ve kestirimsel eşitleme) elde edilen öznitelik adlarından herhangi biri çoğunluk kuralı işletilerek seçilir. Öznitelik seçimi farklı

domainlere sahip olabilir fakat global arayüzün sadece bir tek domaini olmalıdır. Bu durumda global arayüz için bir domain tayin yaklaşımına gereksinim duyulur.

Yapılan çalışmada global arayüz için öznitelik domain tayinini şu şekilde belirlenmektedir (He ve diğerleri, 2004):

- (sonsuz + sonlu) → sonlu
- (aralık + herhangi) → aralık
- (sonlu + sonsuz) → hibrit
- (Sonlu + hibrit) → hibrit
- (sonlu + mantıksal) → sonlu

Birleştirilen arayüz, semantik olarak tekil veya uyumlu arama motorlarına ait değerleri içermek zorundadır.

Sadece belli domainlerde arama yapan bir global arayüz geliştirmek gerekirse, alt grup arama motorları için yeni bir global arayüzü sıfırdan geliştirmek veya global arayüzden çıkarılabilir bazı bileşenleri tespit edip atmak yöntem olarak tercih edilmekte ve (Dragut, Fang, Yu ve Meng, 2009)'de ikinci durumun daha hızlı bir yöntem olduğu gösterilmektedir.

GE (Global Elements), altındaki her bir elemanı bir veya daha fazla LE (Local Elements)'ye eşitleme işlemidir. Durum korumalı şema eşitleme ve durum korumasız şema eşitleme olmak üzere iki farklı eşitleme tercih edilebilir. Eşleştirmede şema elemanları iki grupta ele alınmaktadır (He, Meng, Yu ve Wu, 2003); metinsel değerler ve sayısal değerler.

Birleştirilmiş arama arayüz üretimi

Birleştirilmiş arayüz oluşturulurken sonsuz global öznitelikler için textbox girdiler kullanılır. Sonlu global öznitelikler için de select list elemanından yararlanılır. Daha sonra öznitelikler için arayüz yerleşimleri yapılır. Her bir yerel arayüzdeki öznitelik pozisyonlarının grup ortalaması ile global arayüzdeki özniteliklerin konumları tespit edilir.

Birleştirilmiş arayüz tamirâtı

Global arayüzü oluşturduktan sonra bazı yerel arama ekranlarında yeni bir arama bileşeni eklenebilir ya da var olan bileşenlerden biri çoğu yerel arama arayüzünden zaman içerisinde çıkartılabilir. Bu durumda, şayet yeni bir bileşen eklenmişse pozitif eşleştirme temelli kümeleme adımları çalıştırılarak yeni eklenen öznitelikler var olan kümelerden uygun olanlara eklenir. RAN değeri yeni yeni ve eski istatistiklerle semantik bilgi doğrultusunda güncellenir. Eşleme bilgileri oluşturulur. Son olarak global arayüze gerekli yeni öznitelikler eklenir.

Şayet öznitelik çıkarımını gerektiren bir durum varsa eşleniği değerler kümelerden, ilgili eşlemeler, eşleme bilgilerinden silinir. Şayet gerek duyuluyorsa yeni domain tipleri belirlenir.

3.2.4. Arama motoru bağlantısı

Arama motorları ile nasıl bağlantı sağlanacağı ve hangi özniteliklerin eşitleneceğinin tayin edilmesinden sonra bağlantının otomatik olarak sağlanması, HTTP isteklerinin oluşturulması ve yanıtların kaydedilip işlemde geçirilmesi gibi süreçlerin de yönetimine ihtiyaç duyar. Ayrıca arama motorlarında meydana gelecek değişimlerin de algılanarak tamir edilmesi gerekir.

Arama motoru seçimi

Meta arama motorlarının kullanıcı sorgusu için farklı arama motorlarından hangisini seçeceklerine karar vermek için arama motorlarının özelliklerini bilmeleri gereklidir. Örneğin, esnek sorgu yapılıp yapılmadığı, mantıksal operatörler kullanılıp kullanılmadığı gibi bilgilere sahip olması gerekir. Meta arama motorları, kullanıcıdan gelen bir sorunun içeriğine göre arama motorlarına uygunluk değeri verir ve rank değerlerine göre sıralar. Rank değeri en yüksek olandan itibaren belli sayıdaki arama motoruna kullanıcı sorgusu gönderilerek sonuçları alınır. Diğer bir arama motoru seçim yönteminde ise, arama motorlarının daha önceki sorgu sonuçlarını kullanıcının tıklama durumuna göre sonraki sorgularda seçilme durumu değerlendirilir.

Bu tez kapsamında geliştirilen meta arama motoru sorguya en uygun arama motorlarının seçimini yapmaktadır.

Arama motoru bağlantısı

Birçok durumda HTML form etiketleri bağlantı sağlamak için yeterli meta-bilgiyi içermektedir. Genellikle, arama motoru eşleme sürecinde bağlantı için gerekli bilgiler elde edilmiş olur. Bu aşamada elde edilen ve kaydedilen bilgiler kullanılır. Ancak Meta-arama motorunun her bir yerel arama motoruna sorguları iletebileceği bir HTTP istek bileşenine ihtiyaç duyulur.

Sorgu yeniden yazımı

Bazı çalışmalarda (Gollapudi, Jeong, Ntoulas ve Paparizos, 2011; Zhang ve diğerleri, 2004) sorgular yerel arama motorlarının daha iyi sonuç getirebilmeleri için her bir yerel arama motoruna özgü olarak yeniden yazılabilmektedir. Basit sorgular için bu adım göz ardı edilebilir.

Sorgu genişletme ve sorgu daraltma

Benzer şekilde bazı yerel arama motorları için global sorguyu daraltmak ya da genişletmek gereksinimi olabilmektedir. Ancak basit sorgular için bu adım da göz ardı edilebilir.

3.2.5. Arama sonuç çıkarımı

Arama motorlarının sonuçları, dinamik olarak oluşturulan HTML belgeleri şeklinde gösterilir. Meta arama motorunun bu sonuçlardan arama sonuç kayıtları (SRR) değerlerini çıkartarak ele alabilmesi gerekir. SRR'lerden bazıları konuyla ilgili ancak reklam amaçlı olabilir. Bazı arama motorlarında ise meta arama motorunun kullanmayacağı linkler bulunabilmektedir. Meta arama motorunun bu linkleri veya bilgileri ayıklayarak organik sonuçlara ulaşması gerekir. Yapılan aramaya göre bazı arama motorlarında daha az öznitelik varken bazılarında daha fazla öznitelik döndürülüyor olabilir (Dorn ve Naz, 2008).

Yumusak ve arkadaşları ise bağılı sonuçların SPARQL ile çıkarımının performans açısından incelemesini yapmışlardır. (Yumusak, Dogdu, Kodaz, Kamilaris ve Vandenbussche, 2017)

Yerel sonuç-global sonuç eşlemesi

Yerel sonuç-global sonuç için eşleme aşamasında 3.3.2’de verilen tekniklerle eksik öznitelik vs. gibi durumlar dikkate alınarak sonuç eşleştirmesi yapılır. Global sonuç için de benzer şekilde 3.3.3’de anlatılan yöntemlerden yararlanılır.

Global sonuç birleştirme

Farklı arama motorlarından gelen sonuçların birleştirilerek kullanıcıya tek bir liste halinde sunulması gereklidir. Meta arama motorlarının kullandığı farklı birleştirme yöntemleri bulunmaktadır. Farklı arama motorlarından gelen sonuçlarda aynı bilgiyi içeren sonuç olsa bile bu veriler olduğu gibi aynı olmayabilir. Örneğin, seyahat planlaması için bir hedefe yönelik araştırma yapan kullanıcıya aynı hedef için farklı havayolu şirketlerine ait veriler farklı formatta ve içerikte gelebilir. Bu verilerin anlamsal olarak içeriğinin incelenmesi ve birbirini tamamlar şekilde birleştirilmesi gereklidir. Bazı havayolu şirketleri kalkış yerine “origin” bazıları ise “start” şeklinde etiket kullanabilmektir. Bu anlam benzerliklerine göre gelen sonuçların analiz edilmesi gereklidir.

3.2.6. Arama sonuç gösterimi

Arama sonuçlarının eşleştirilme, ayıklama ve birleştirme sürecinden sonra daha kullanıcı dostu bir şekilde gösterilebilmesi için arama sonuçlarının yeniden bir sıralamaya tabi tutulması gerekir. Ayrıca, şayet aranan kelimenin ontolojisi hakkında yorum yapabilecek bir algoritma kullanılarak kullanıcının amacına yönelik daha spesifik bilgiler verecek arama kaynaklarından alınabilecek sonuçlar ek olarak kullanıcıya gösterilebilir.

Derecelendirme yaklaşımları

Arama sonuçlarının tekrardan sıralanmasında esas alınabilecek çeşitli veriler mevcuttur. Bunlar iki temel grupta ele alınabilir: arama motorlarından gelen veriler ve arama sonucunda elde edilen verilerin anlamsal değerleri. Bir arama motorundan elde edilen i) Her bir ASK’nın

arama motorları sonuçlarındaki orijinal rank değeri, ii) ASK-tekrarlama sayısı: Her bir ASK'nın arama motorlarından kaç tanesinde yer aldığı gibi bilgiler kayıtların yeniden sıralanmasında birçok araştırmada yararlanılmıştır (Akritidis ve diğerleri, 2011).

Arama motorlarından veya farklı Web kaynaklarından elde edilen sonuçların sıralanmasında farklı yöntemler kullanılmaktadır (Branch, 2012; Diaz, De ve Raghavan, 2004; Lawrence, 1998; Lu, Meng, Shu, Yu ve Liu, 2005; Vijaya ve diğerleri, 2016). Ancak bir sonucun arama ile ne kadar ilgili olduğunu bulmak için temelde kullanıcı tarafından elle-gözle sonucun sorgu ile ilgili olup olmadığının belirtilmesi (Lu ve diğerleri, 2005; Vijaya ve diğerleri, 2016) veya arama motorlarının birinin diğerine kontrol ettirilmesi (hangi arama motorunda hangi sırada geldiği ve arama motorlarının ağırlıklandırılması gibi varyasyonlar da dahil) gibi yöntemlerle yapılmaktadır (Diaz ve diğerleri, 2004). Meta arama motorları gelen sonuçlar için yeniden bir rank değeri hesaplarlar ve kullanıcıya o değere göre bir liste hazırlayarak sunarlar (Lu ve diğerleri, 2005).

Sorgu sonucunda farklı arama motorlarından gelen sonuçların tekrar listelenmesindeki amaç her bir sonucun sorguya uygunluk değerini tahmin etmeyi amaçlar. Diğer bir yöntemde ise, meta arama motorları gelen sonuçlara ait dokümanları fetch ederler ve yeniden uyumlulukla ilgili değerlendirme yaparlar. Bu şekilde elde edilen listedeki rank değerleri aynı ölçütlere göre hesaplanmış olur.

Ancak, tüm dokümanların tekrar kaynağından alınması ve yeniden içeriğine göre rank değeri hesaplanmasından dolayı uzun zaman gerektirir (Dwork ve diğerleri, 2001). Sonuçların bir kısmı farklı arama motorları tarafından aynı şekilde elde edilmiş olabilir. Bu sonuçlara farklı arama motorları farklı rank değerleri atamış olabilir. Meta arama motorlarından bazıları bu tür aynı sonuçları, farklı arama motorlarından gelen rank değerlerinin toplamını alarak sonuç listeye aktarmaktadırlar.

Kullanıcı dostu ve esnek gösterim

Google Knowledge Graph aranan bir kelimeyi ontolojik olarak eşledikten sonra kendi veri tabanlarında bulunan 500 milyon nesne, 3.5 milyar gerçeklik ve bunlar arasındaki ilişkiler üstünden aramaya özgü bilgiler sağlayabilmektedir (Steiner ve Mirea, 2012). Ancak aynı çalışmada bu büyüklükte bir genel kullanıma açık bilgi graf kaynağı olmadığı bu nedenle de

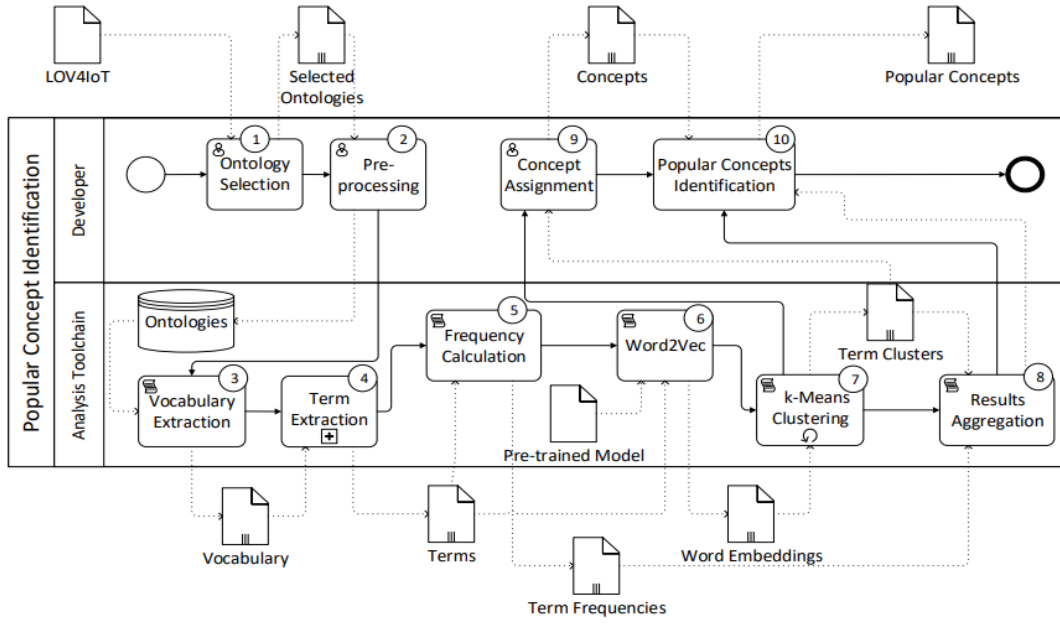
kullanıcı ara yüzlerinde gelen sonuçların bir veri tabanında birleştirilmesi şeklinde bir çalışma yapılmıştır (Steiner ve Mirea, 2012). Ancak aranan kelimenin ontolojik analizi yapılarak uygun dikey-yatay arama motorları seçilerek bilgi grafi üretilmesi ile ilgili çalışmalar tez kapsamında incelenmiştir.

3.2.7. Web bilgi çıkarımı ve makine öğrenmesi

Arama Sonuçlarından elde edilen verilerin yanı sıra bazı çalışmalarda arama motoru sorgularından elde edilen veriler de meta-arama motorlarında bilgi çıkarımı amacıyla kullanılmıştır. Paşca ve arkadaşları tarafından yapılan çalışmada (Paşca, 2007) arama motorundaki arama sorgularından hareketle belli domainlere ilişkin varlık çıkarımı ele alınmıştır. Semantik Web Çıkarımı halen aktif bir çalışma alanıdır. Benzer şekilde Noura ve arkadaşları (Noura, Gyrard, Heil ve Gaedke, 2018), Konsept ontolojilerle ilgili gruplama çalışmaları yapmış ve IoT ve WoT alanında en popüler 14 ontolojiyi tanımlamışlardır. Bu tanımlama ve kullanılan yöntemler, var olan ontolojilerin yeniden kullanılması ve birleştirilmesi gibi süreçleri yönlendirecek yöntemler içermektedir. Bu amaçla geliştirilen sistem Şekil 3.4’de yer almaktadır.

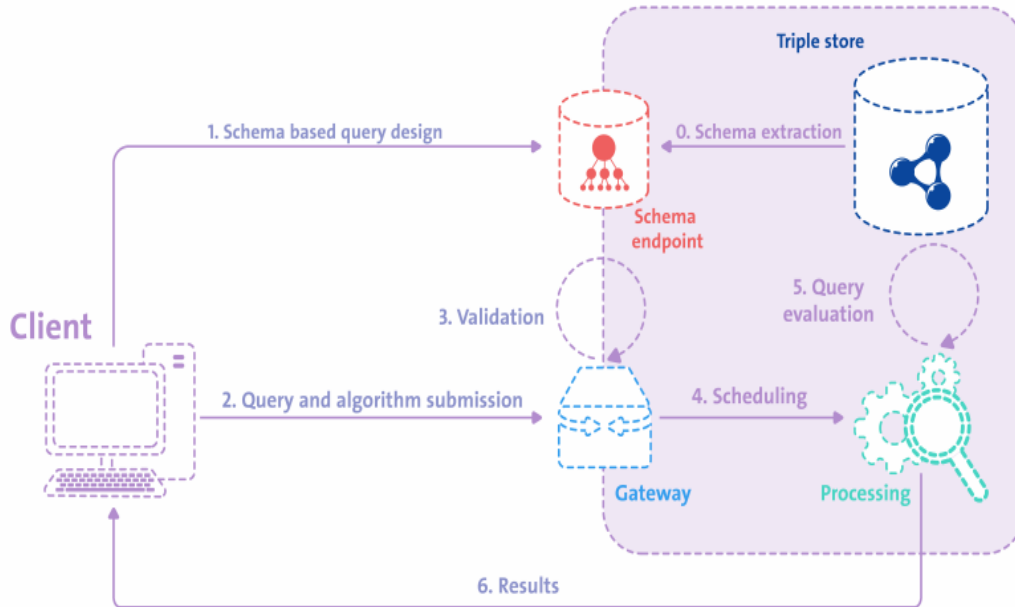
- Ön işleme
- Ontoloji seçimi
- Kelime çıkarımı
- Terim çıkarımı
- Frekans hesaplaması
- Yapay Sinir Ağları temelli word2vec algoritması ile kelimelerin tekilleştirilmesi ve ontoloji dışı kelimelerin çıkarılması
- K-means ile kümeleme
- Sonuç birleştirme
- Konsept ataması
- Popüler kavram kimliklendirme

aşamalarından oluşmaktadır.



Şekil 3.4. Kendi kendine ontolojik öğrenme yapan bir sistemin bileşenleri

Gleim ve diğerleri (Gleim ve diğerleri, 2018), yaptıkları çalışmada Web çıkarımı sırasında kişisel verilerin korunması amaçları doğrultusunda otomatik şema çıkarımı sağlayan bir sistem önermişlerdir.



Şekil 3.5. Otomatik şema çıkarım sistem mimarisi.

SPARQL (SPARQL Protocol and RDF Query Language) şema çıkarım sorguları kullanılmıştır. Şema çıkarımının başarım ölçütü için HCLS ölçüm sistemi kullanılmış ve

schema.org ve foaf sözlüklerinin karakteristiği ile kıyaslanmıştır. Kıyaslama sonuçları Çizelge 3.1.'de yer almaktadır.

Çizelge 3.1. Şema çıkarım algoritmasının schema.org ve foaf ile kıyaslanması.

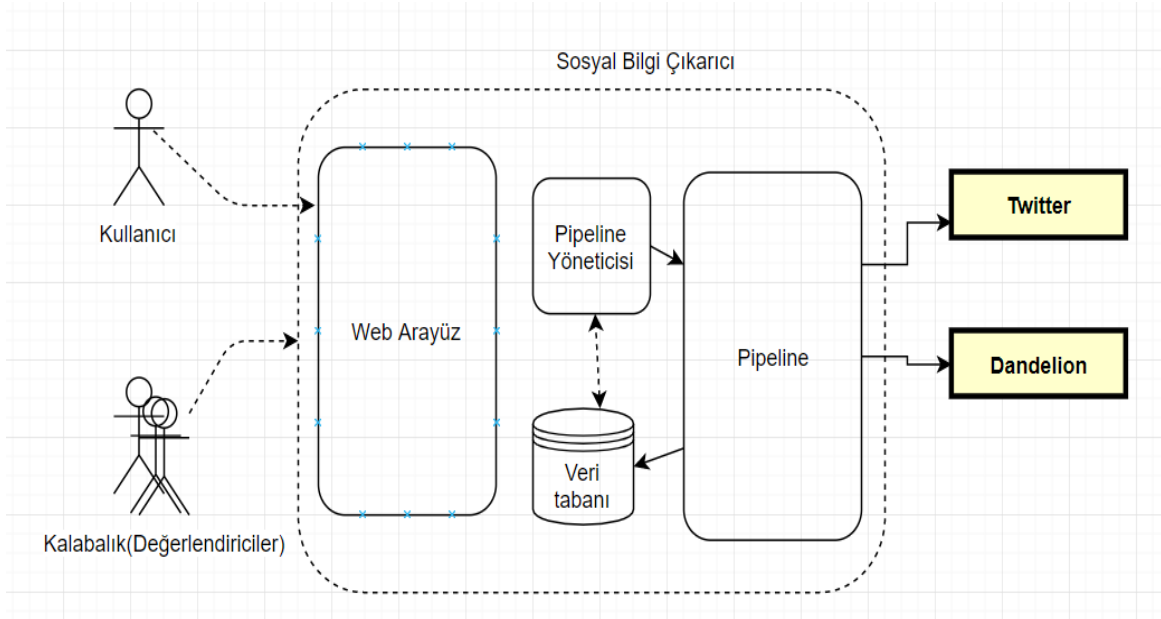
HCLS	Şema Numarası	Şema	foaf	TOPLAM	ÇIKARIM
6.6.1.1	Üçlükler	8427	631	9058	317
6.6.1.2	Tekil, Tip Tanımlı varlıklar	1617	84	1701	86
6.6.1.3	Tekil Konular	1619	86	1705	86
6.6.1.4	Tekil Özellikler	15	15	30	14
6.6.1.5	Tekil Nesneler	476	38	514	43
6.6.1.6	Tekil Sınıflar	31	9	40	8
6.6.1.7	Tekil Literaller	3193	154	3347	105

(Brambilla ve diğerleri, 2018) tarafından yapılan çalışmada sosyal medya içeriğinden gelişen bilgi üretimini hedeflemişlerdir. Bu yaklaşımda domain uzmanlarınca belirlenen çekirdek terimlerden hareketle Karışık Sentaktik-Semantik metot ortalamaları ile gelişen bilgi üretilebilmektedir. Çekirdekten hareketle çeşitli adaylar çıkartılmakta, daha sonra bu adaylar çekirdeğin merkezine uzaklık ve terim frekansı gibi değerlerden yararlanılarak en öne çıkan adaylar çekirdeğe dâhil edilmekte ve bu iterasyon tekrarlanmaktadır.

Yapılan çalışma (Brambilla ve diğerleri, 2018), Döngüsel tekrarlamaların düzenlenmesi amacıyla şu sorulara cevap vermektedir:

- Rekürsif olarak çıkarımı yapılan bilgiler yeniden çekirdek olarak kullanılırsa, inşa edilen domain bilgisi nasıl gelişir?
- İnşa edilen domain bilgisi coğrafi olarak nasıl dağılım gösterir?
- Yöntem bilginin geçmişi bugünü ve geleceğini analiz etmek için kullanılabilir mi?
- Yöntem gelişen bilgiyi bulmak için kullanılabilir mi?

Çalışmada yer alan iterasyonlara ilişkin bileşenler ve döngü Şekil 3.9.'da yer almaktadır.



Şekil 3.6. İteratif bir Bilgi Çıkarım aracının gerçekleştirme mimarisi

Aynı çalışmada elde edilen farklı domainlere ilişkin bilgi çıkarım miktarı başarımının bir ölçümü olarak ele alınmıştır ve Çizelge 3.2’de yer almaktadır.

Çizelge 3.2. 1 aylık dönemde göre 4 farklı domainde bilgi çıkarım miktarları

Zaman Aralığı	Satranç	Finans	Yazarlar	Moda
2016/01-2017/09	545	151	780	153
2016/01-2017/06	310	52	329	123
2016/01-2017/03	210	45	237	103
2016/01-2017/12	146	0	177	95
2016/01-2017/09	78	0	94	79
2016/01-2017/06	43	0	45	61
2016/01-2017/03	10	0	25	27

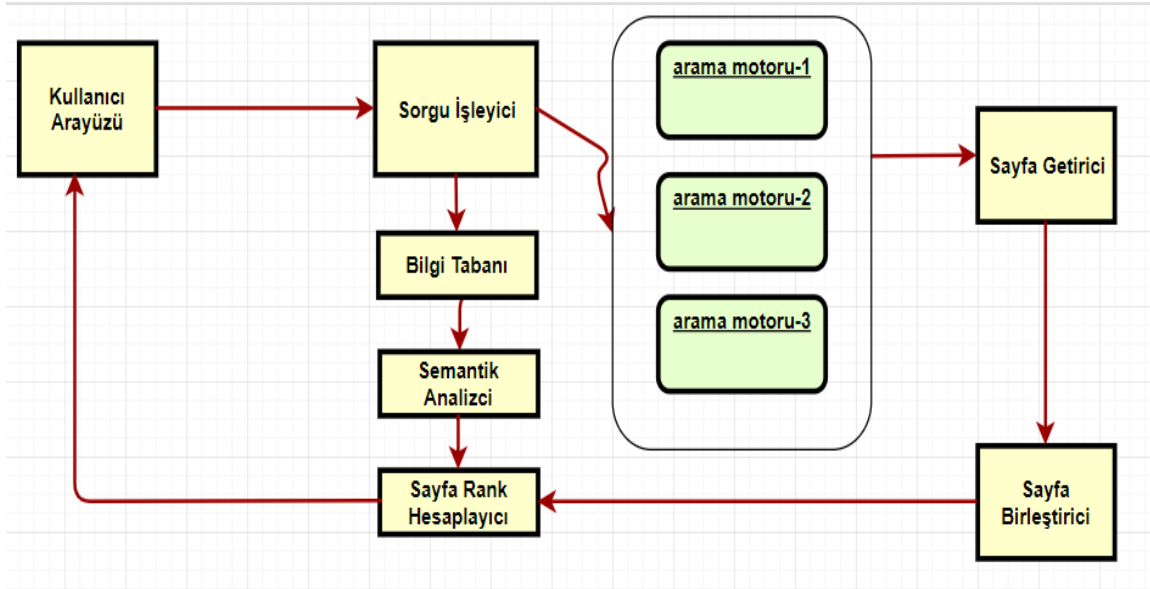
Çizelge 3.2. Çıkarımı yapılan varlıklar için doğruluk kıyaslaması

Domain	Geliştirilen Varlık Sayısı
Moda Tasarım	100%
Finans	77%
Satranç Oyunu	42%
Avustralyalı Yazar	36%

(Fayzrakhmanov, Sallinger, Spencer, Furche ve Gottlob, 2018) tarafından yapılan çalışmada, yeni nesil Web uygulamalarının ürettiği verilerin İnternet Gezgini olmadan Web Veri Çıkarımının yapılmasını ele almışlardır. Özellikle son yıllarda neredeyse tüm modern uygulamaların REST API (Representational State Transfer- Application Protocol Interface)’ler üstünden Tek Sayfa Uygulama (Single Page Application-SPA) mimarisi ile geliştiriliyor olması yapılan bu çalışmayı önemli kılmaktadır. REST API’lerin genel olarak JavaScript Obje Notasyonu (JSON) kullanmasından dolayı bu yapılan bu çalışma JSONPath seçinini de destekleyecek bir Algoritma önermektedir.

Çalışmada, browser olmadan veri çıkarımı için FASTWRAP algoritması geliştirilmiştir.

(Kumar, 2017) tarafından yapılan çalışmada, arama motorlarından gelen sonuçların birleştirilme sıralamasının belirlenmesinde domain uyumluluğu üstünden çalışan bir genetik algoritma kullanılmıştır. Genetik Algoritmanın uzun çalışma ve hesaplama maliyeti göz önüne alındığında, yöntem çalışma zamanının uzun olması nedeniyle pratikte uygulanabilir bulunmamıştır. Çalışmada önerilen sistemin temel bileşenleri Şekil 3.7.’de yer almaktadır.



Şekil 3.7. Semantik analiz ve sorgu işleme ile meta-arama motoru mimarisi

Sistem, kullanıcıdan aldığı sorguları işleyen bir birim ile farklı arama motorlarına bağlanmakta, sonuçlarını sayfa getirici ve birleştirici sistemler ile aldıktan sonra rank hesaplayıcı ile sonuçları sıralamaktadır. Sayfa rank hesaplayıcısı, Bilgi tabanı ve semantik analizciden de yararlanmaktadır.

4. OTOMATİK ŞEMA EŞLEME

Web arama sonuçlarının çıkarımı ve birleştirilmesi sırasında otomatik şema eşleme yaklaşımlarından yararlanılmıştır. Bu bölümde, otomatik şema eşleme hakkında yapılan çalışmalara yer verilmiştir.

4.1. Otomatik Sonuç Çıkarımı Algoritması Geliştirilmesi

Bir Web arama sonucuna erişmek ve programatik bir ortamda arama sonuçlarını tekrardan kullanabilmek web-veri demetleme, meta arama motorları, Web madenciliği, sosyal web madenciliği gibi birçok alanda kolaylık sağlar. Ancak önde gelen arama motorları için herkese açık ve iyi tanımlı bir arama sonuçlarına erişim sağlayan ortam mevcut değildir. Bazı arama motorlarından çok azı API (Application Program Interface) sağlasa da bu desteğin sürekliliğini garanti etmemekte ve ansızın sonlandırabilmektedir.

Meta-arama motorları ve veri demetleme sistemlerinde herkese açık arama motorlarından elde edilen arama sonuçlarından her birine Arama Sonuç Kaydı (ASK) denilmektedir. Web sonuçlarının otomatik olarak çıkarımı için birçok çalışma yapılmıştır. Otomatik Web sonuç çıkarımında iki temel yaklaşım, sarmaç (wrapper) üretimi ve otomatik şablon üretimidir. Bu iki yönteme ek olarak, ikisinin birlikte kullanıldığı hibrit yaklaşım da başka bir yöntem olarak kullanılmaktadır.

Bir sarmaç, Web çıkarım sistemlerinde sarmal indükleme uygulaması tarafından kullanılır. Sarmaç, farklı çalışmalarda farklı işlevleri yerine getirebilen bir bileşen olarak ele alınmaktadır. Bu çalışmada, genel anlamı ile “Web sonuç çıkarımı için kullanılan herhangi bir ön tanımlı kod veya sembolik tanımlama” şeklinde kullanılmıştır. Sarmaç indükleme temelli yaklaşımlar genel olarak 3 adımda kullanılırlar:

- 1) Sarmaç üretimi,
- 2) Sarmaç çalıştırma,
- 3) Sarmaç tamirâtı.

Bir sarmaç kullanılabilmesi için öncelikle bir yazılımcı veya bir yazılım tarafından tanımlanmak zorundadır. Arkasından bu tanım veya kod devreye sokularak ASK'lar çıkarılabilir. Web gösterimleri zaman içerisinde değişebileceğinden, sarmaçların da bu değişime uygun şekilde tamir edilmesi gerekir.

Bir diğer popüler Web çıkarım kategorisi otomatik şablon oluşturmadır (Crescenzi, Mecca ve Merialdo, 2001). Bu yaklaşım, HTML sonuç sayfalarının sunucu tarafında otomatik olarak üretilmelerinden hareketle belli bir genel desenden geçmesi gerektiği varsayımına dayanmaktadır. Otomatik şablon oluşturma genellikle ağaç yapısını kullanarak tekrarlayan desenleri bulmayı hedefler. Bazı araştırmacılar görsel öğelere odaklanıp Görsel Blok Ağaçlar (Visual Block Trees)'dan yararlanırken, bazıları da HTML etiketlerinin işlenmesi ile ilgilenirler.

Bir web sonuçları sayfasının düzenli desenlerden oluşan sonuçları göstermesi beklenirken, genellikle düzenli kayıtlardan ayrılan ASK'ların da düzenli ASK'larla aynı listelerde yer alırlar (Şekil 4.1). Bu çalışmada, olağan desendeki kayıtlardan ayrılan bu tür ASK'ları Düzensiz ASK olarak adlandırılmıştır. Düzensiz ASK'lar, tekrarlayan desenlerin algılanmasını zorlaştırmaktadır. Bazı hallerde, ASK'lar çok fazla karmaşıklıkta HTML verileri içerdiğinden, ASK'ların tam olarak hangi bölgede başladığını bulmak daha da zor bir hal alabilmektedir.

← → ↻ https://www.google.com.tr/?gws_rd=ssl#q=gazi+university

Google gazi university

Web Görseller Videolar Haberler Daha fazla ▾ Arama araçları

Yaklaşık 582.000 sonuç bulundu (0,30 saniye)

gazi üniversitesi için olan sonuçlar da gösteriliyor
Yalnızca gazi university araması yap

Gazi Üniversitesi
gazi.edu.tr/ ▾
Michigan State Üniversitesi (MSU) ile **Gazi Üniversitesi** Her Konuda Tam Bir İşbirliği ...
Gazi Üniversitesi Vakfı Özel Okulları 2014-2015 Eğitim Öğretim Yılı Açılışı ...
3,9 ★★★★★ 26 Google yorumu · [İnceleme yazın](#)

Emniyet Mh., 06560 Ankara
(0312) 491 3866

gazi.edu.tr sitesinden sonuçlar

Öğrenci İşleri Bilgi Sistemi ... Ankara - Türkiye, Location : Ankara - Turkey, Öğrenci İşleri ... Kütüphane Kütüphane bilgileri, çalışma saatleri, duyurular, projeler ... Akademik Gazi Eğitim Fakültesi - Mühendislik Fakültesi - Fen Fakültesi - ...	Tercih Robotu GAZİ ÜNİVERSİTESİ Tanıtım Sayfası için tıklayınız... GAZİ ... Zorunlu Basın Açıklaması Zorunlu Basın Açıklaması. Bir gazetenin Ankara ekinde hiçbir ... İdari Birimler İdari Birimler ... Halkla İlişkiler Müşavirliği · İdari ve Mali İşler ...
---	--

Gazi University - Facebook
<https://tr-tr.facebook.com/gaziuniversitesi?fref=nf> ▾
Gazi Üniversitesi, Ankara. 118.875 beğenme · 1.707 kişi bunun hakkında konuşuyor · 39.821 kişi buradaydı. **Gazi Üniversitesi** resmi Facebook sayfasına...

Gazi Üniversitesi - Ankara - Üniversite ve Yüksekokul ...
<https://tr-tr.facebook.com/gaziuniversitesi> ▾
Gazi Üniversitesi, Ankara. 119.078 beğenme · 1.707 kişi bunun hakkında konuşuyor · 40.141 kişi buradaydı. **Gazi Üniversitesi** resmi Facebook sayfasına...

gazi university ile ilgili görseller Görseller hakkında kötüye kullanım bildirin



gazi university için diğer resimler

Gazi Üniversitesi - Vikipedi
tr.wikipedia.org/wiki/Gazi_Üniversitesi ▾
Gazi Üniversitesi, Ankara'da bulunan bir devlet üniversitesidir. 1926 yılında Ankara'da kurulmuştur. Sanat eğitimi alanında Cumhuriyet tarihinin öncüsüdür.

Gazi University - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Gazi_University ▾ Bu sayfanın çevirisini yap
Gazi University (Turkish: **Gazi Üniversitesi**) is a public university primarily located in Ankara, Turkey. It was established in 1926 by Mustafa Kemal Atatürk as Gazi ...

Gazi Üniversitesi (@Gazi_University) | Twitter
https://twitter.com/Gazi_University ▾
The latest Tweets from **Gazi Üniversitesi** (@Gazi_University). **Gazi Üniversitesi** resmi Twitter hesabı. / Official Twitter account of **Gazi University**. Teknikokullar ...

Şekil 4.1. Düzensiz ASK'lar içeren örnek bir Google arama sonucu

Tez çalışmasında, ASK'ların kullanıcı etkileşimi olmaksızın çıkarımı için yeni bir yaklaşım geliştirilmiştir.

Var olan çalışmalarla kıyaslandığında, bu çalışmada üç temel katkı sağlanmıştır:

- Bir arama sonuç sayfası nüshasından hareketle otomatik olarak ASK'ları algılayabilecek bir yöntem geliştirilmiştir.
- ASK'lara düzenli olmayan sonuçlar da dâhil edilmiştir.
- Geliştirilen metot, genel web veri kaynakları üstünde ve geniş bir arama anahtar kümesi ile test edilmiştir.

Bir düzenli ASK, HTML kodu, süslü parantez gösterimi ve basitleştirilmiş Şekil 4.2.'de verilmiştir.

ANKARA İçin 5 Günlük Hava Tahmini - Meteoroloji Genel Müdürlüğü

<https://www.mgm.gov.tr/.../tahmin-il-ve-ilceler.aspx?m=ANKARA>

Ankara Merkez · Akyurt · Altındağ · Ayaş · Bala · Beypazarı · Çamlıdere · Çankaya · Çubuk · Elmadağ · Etimesgut · Evren · Gölbaşı · Gündül · Haymana · Kalecik ...

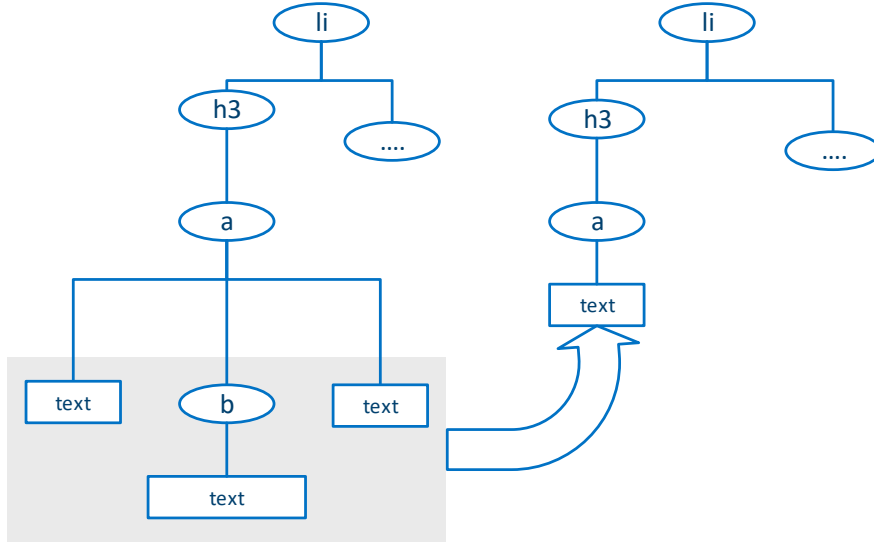
```
<h3 class="r"><a href="/url?q=
https://www.mgm.gov.tr/mobile/tahmin-il-ve-ilceler.aspx?3Fm%3DANKARA&sa=U&ved=0ahUKEwivtseIh9vQAhXpIJoKHQpeDzkQFqg_MAc&usq=AFQjCNEaeJolPtqsY4_BP0240ltF_QTTBQ"><b>ANKARA</b> İçin 5 Günlük Hava Tahmini - Meteoroloji Genel Müdürlüğü
</a></h3>
<div class="s">
<div class="kv" style="margin-bottom:2px">
<cite>https://www.mgm.gov.tr/.../tahmin-il-ve-ilceler.aspx?m=<b>ANKARA
</b></cite>
<div class="_nBb">
<div style="display:inline" onclick="google.sham(this):" aria-expanded="false"
aria-haspopup="true" tabindex="0" data-ved=
"0ahUKEwivtseIh9vQAhXpIJoKHQpeDzkQ7B0IQDAH">
<span class="_OO"></span>
</div>
<div style="display:none" class="am-dropdown-menu" role="menu" tabindex="-1">
<ul>
<li class="_Ykb"><a class="_Zkb" href="/url?q=
http://webcache.googleusercontent.com/search?3Fq%3Dcache:fShm09nd3ycJ:https://
www.mgm.gov.tr/mobile/tahmin-il-ve-ilceler.aspx?3Fm%3DANKARA%252Bankara%26
hl%3Dtr%26ct%3Dcink&sa=U&ved=0ahUKEwivtseIh9vQAhXpIJoKHQpeDzkQIAhCMac
&usq=AFQjCNERfcQVGD6pDH_Ft2vSsXwCqqlPFA">Önbellek</a></li>
</ul>
</div>
</div>
</div>
<div>
<span class="st"><b>Ankara</b> Merkez · Akyurt · Altındağ · Ayaş · Bala ·
Beypazarı · Çamlıdere · Çankaya <br> · Çubuk · Elmadağ · Etimesgut · Evren ·
Gölbaşı · Gündül · Haymana · Kalecik&nbsp;...</span>
<br>
</div>
```

```
li{ h3{ a{ #text b{ #text} #text)} div{ div{ cite{ #text}
div{ #text div{ span} div{ ul{ li{ a{ #text}} li{ a{ #text}}}}}}
span{ #text b{ #text} #text br #text} br)}
```

```
li{ h3{ a{ #text}} div{ div{ #text div{ #text div{ span}
div{ ul{ li{ a{ #text}} li{ a{ #text}}}}}} span{#text} br)}
```

Şekil 4.2. Bir düzenli ASK için basitleştirilmiş ağaca ait gösterimler.

HTML dokümanlarının doğası nedeniyle bazı ASK DOM'ları çok karmaşık olabilir. Bu nedenle ağacın bir algoritma ile sadeleştirilmesine ihtiyaç duyulmaktadır. Basitleştirme işlemi Şekil 3'te gösterilmiştir.



Şekil 4.3. ASK DOM ağacından bir kısmı ve bu kısmın basitleştirilmesi.

Basitleştirme işleminin amacı Tree Edit Distance algoritmasının işlem maliyetini azaltmaktır. Basitleştirme işlemi karmaşık ağaç yapısını düzleştirir ve içeriğin formatlanması ile ilgili alt ağaç bileşenlerini bir üst düğümde birleştirir. Böylece, iki farklı düğümün benzerliği için aşırı dallanmaların hatalı sonuçlar çıkarmasının önüne geçilmiş olur.

Bazı hallerde, ilgisiz veriler (örneğin sayfa numaralandırmaları gibi), ASK ile aynı düğüm içerisinde yer alabilmektedir. Bu tür durumlarda, ilgisiz alt düğümlerin, düzensiz arama kayıtlarından ayrıştırılması için son bir filtreden geçirme işlemi uygulanmıştır. Bu filtre etiket taraması yaparak, reklam içeriklerini ve görünürlüğü olmayan alt bileşenleri algılama, yoğun alt düğümden oluşan HTML içeriklerinin çıkarılması gibi işlevleri yerine getirmektedir.

4.1.1. Deneysel sonuçlar

Hedeflenen yöntem önde gelen arama motorlarında deney amacıyla oluşturulan bir veri kümesi ile test edilmiştir. Oluşturulan anahtar kelime listesi farklı domainlerden seçilmiş olup aşağıdaki veri kaynaklarından elde edilmiştir.

- Google Arama Trendleri (150 kelime)
- AOL Arama Trendleri (100 kelime)

- Indeed.com (10 trend iş arama kriteri)
- Amazon.com (En çok satan 15 öge)
- Ebay.com (En çok aranan 20 kelime)
- Twitter (Trend olan 5 hashtag)
- IMDB (en çok beğenilen 200 film etiketi)

Deney 1:

Veri kümesi Google, Bing ve Yandex’de aranmış ve yapılan arama sonuçlarında düzensiz ASK’ların toplam ASK’lara oranı hesaplanmıştır. Sonuçlar Çizelge 4.1’de gösterilmiştir.

Çizelge 4.1. Veri Kümesi için düzenli ve düzensiz ASK’ların oranları

	En az 1 Düzensiz ASK	Hepsi Düzenli ASK
Google	%88	%12
Bing	%79	%21
Yandex	%76	%24

Deney -2:

Google.com, Bing.com ve Yandex.com’da veri kümesi için arama işlemleri yapılmıştır. Daha sonra, el ile tasarlanmış Sarmaç tanımı kullanılarak bulunan sonuçlar doğru kabul edilmiş ve geliştirilen yöntemin bulduğu sonuçlar kıyaslanmıştır.

Literatürde incelenen çalışmalarda kabul görmüş başarımlar ölçüt kriteri olarak Duyarlık ve Doğruluk ölçümleri kullanılmıştır (W. Liu, 2010; A.H. Laender, 2002; C.H. Chang, 2006; E. Ferrara, 2012). Duyarlık ve Doğruluk değerleri sırayla formül (4.1) ve formül (4.2) kullanılarak elde edilmiştir.

$$kesinlik = \frac{|dogru_pozitif|}{|dogru_pozitif| + |yanlis_pozitif|} \quad (4.1)$$

$$duyarlik = \frac{|dogru_pozitif|}{|dogru_pozitif| + |yanlis_negatif|} \quad (4.2)$$

Çizelge 4.2. Farklı arama motorları için Duyarlık ve Doğruluk değerleri

	kesinlik	duyarlık
Google.com	%98,3	%97,1
Bing.com	%97,5	%96,4
Yandex.com	%90,6	%87,8

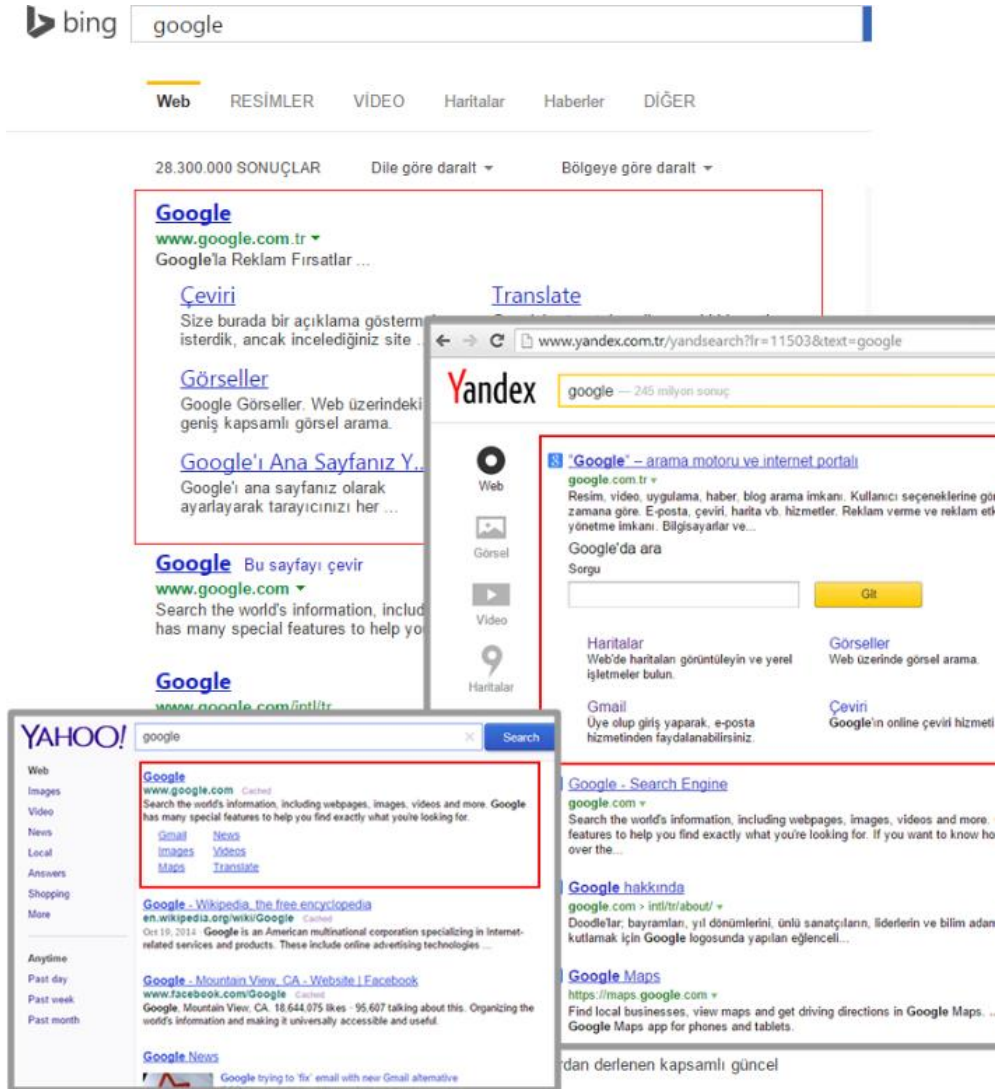
Deneysel Sonuçlar, iyi tasarlanmış arama sonuç sayfalarında yöntemin kullanılabilir olduğunu göstermiştir.

4.2. Otomatik Arama Sonuç Çıkarımı Algoritması Geliştirilmesi

Web arama sonuç kayıtlarının otomatik çıkarımı ile ilgili çalışmada iyileştirmeler yapılmış, sonuçların otomatik eşlenmesi algoritması geliştirilmiş ve geliştirilen algoritmalar genişletilebilir ve yeniden kullanılabilir bir API haline getirilmiştir.

Otomatik Web Çıkarım metotları, çıkarım yaparken yapıları ele alış biçimlerine göre iki farklı kategoride ele alınmaktadır: Görsel Blok Ağaçları (Visual Block Tree) ve Etiket İşleme (Tag Processing). Vide ve DWDE-R Görsel Blok Ağaçları (GBA) kullanırlar. GBA, bir Web sayfasının kullanıcı Web gezgininde görüntülediği şeklini tekrardan programatik olarak dikdörtgenlerden oluşur şekilde üreterek, her bir ögenin HTML ve Cascading Style Sheets (CSS) verilerinden yararlanarak en-boy bilgisi ve üstten-soldan uzaklığı gibi veriler ışığında Web sonuçlarını kestirmeye yarayan bir yapı üretir.

TagOfPath, RankingXPath, ContentDensitiy gibi çalışmalarda HTML Etiket işleme yöntemi kullanılmaktadır. Bu yöntemde, ASK'ların yerini bulmak için HTML-DOM ağacı özellikleri işlenmektedir. TagOfPath bütün ASK'ların birbiri ile tamamen aynı olduğunu, sapan ASK'ların bulunmadığını varsayar. ContentDensity ASK'ları düzenli ASK ve Düzensiz ASK olmak üzere iki farklı gruba ayırır ve sonuçta her iki türden ASK'yı da kapsar.



Şekil 4.4. Farklı arama motorlarında düzenli ve düzensiz ASK'lar

Elde edilen sonuçların birleştirilmesi için otomatik şema eşeme algoritmalarına ihtiyaç duyulmaktadır. Otomatik şema eşleme farklı kaynaklardan elde edilen verilerin birleştirilmesinde ortaya çıkan eski bir problemdir. Yapılan çalışmada hibrit olarak Şema Eşleme yaklaşımlarından yararlanılmıştır.

Yapısal seviye algoritmalar, şema eşlemek amacıyla yapısal bilgileri kullanırlar. Genel olarak farklı şemaları eşleştirmek için dairesel olmayan graf ve ağaç yapıları kullanılır. Graf temelli şemaların çoğu komşuluk benzerliklerini kullanarak benzer bireyleri bulmayı hedefler. Similarity Flooding algoritması buna örnek olarak verilebilir. Tree Edit Distance ve Graph Edit Distance algoritmaları, kaynak şemada çeşitli düğüm ekleme-çıkarma işlemleri ile hedef şemayı elde etmeye çalışırken arada gerekli işlem basamaklarının sayısını

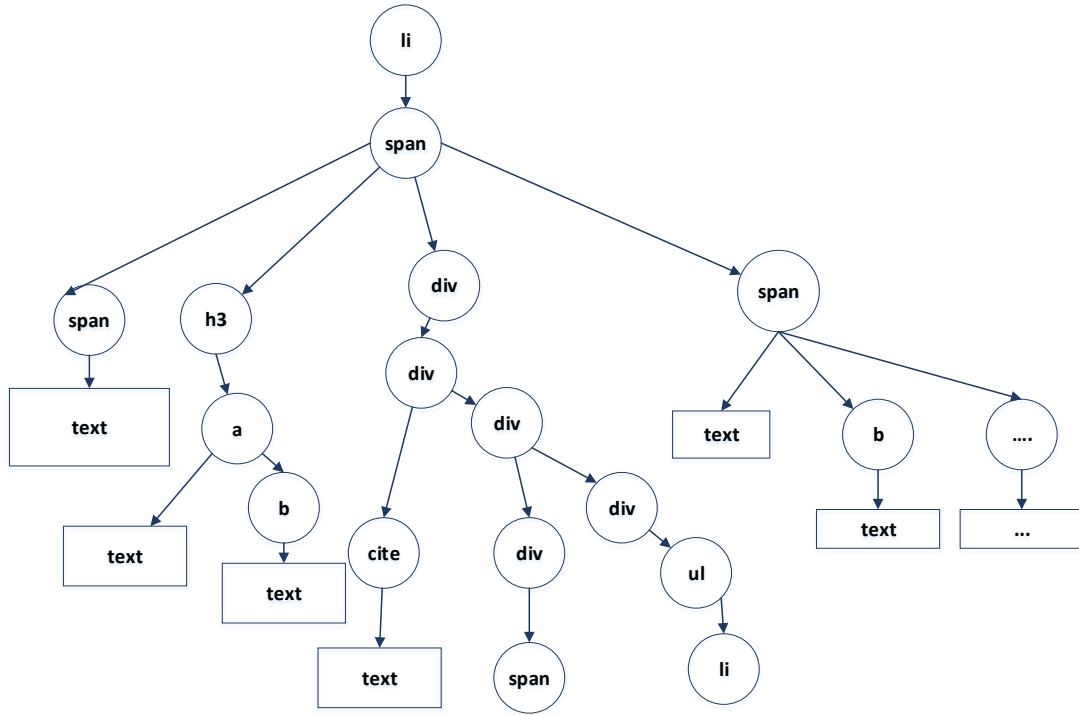
bir uzaklık ölçütü olarak kullanır. Karmaşıklık arttıkça iki graf ya da ağaç yapısının “Edit Distance” uzaklığını bulmak da zorlaşır.

Örneklem seviye şema eşleme algoritmaları bir veri içeren örneklemden hareketle şema bilgilerini çıkarmayı hedefler. Özellikle şemanın az bilindiği veya hiç bilinmediği durumlarda, örneklem veri şema hakkında ek bilgi elde etmek veya şemayı elde etmek için kullanılabilmektedir.

Otomatik şema eşleme konusunda Cupit, CLIO, COMA, OLA ve S-Match gibi birçok çalışma yürütülmüştür. Bu platformlar birbirinden farklı algoritmaları bir arada kullanarak şema eşleme kalitesini artırmayı hedeflemişlerdir. COMA, bu platformlardan halen aktif olarak geliştirilmeye devam eden bir çalışmadır.

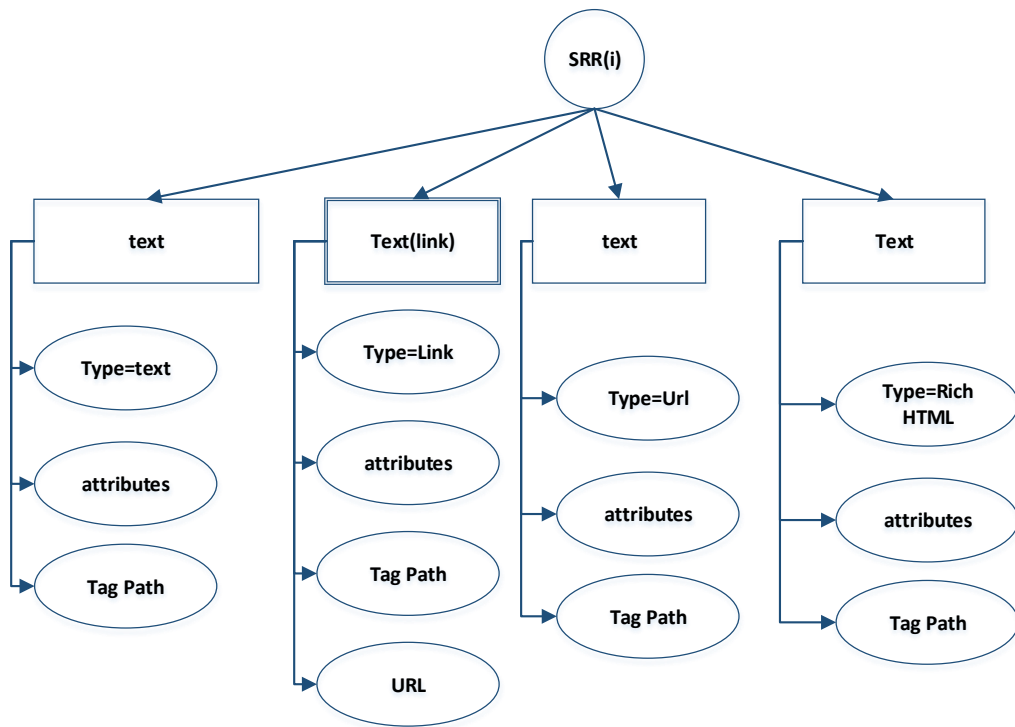
Çalışma kapsamında, bir tek arama sonuç sayfasından hareketle ASK’ların otomatik olarak alt öğeleri ile birlikte çıkartılıp, bir dönüşüm algoritması uygulayarak HTML-DOM ağaçlarının tabular veri formatına dönüştürülmesi hedeflenmiştir. Tabular verilerin eşlenmesi ağaç ya da graf yapılara göre oldukça kolay bir süreçtir.

Bu amaca yönelik olarak, öncelikle uç seviyelerdeki içerik formatlama etiketleri göz ardı edilmiştir. Arkasından bütün uç seviye (Leaf) düğümler gezilerek, düzleştirilmiş veri tabanı formatına çekilmiştir. Tasarlanan bu yaklaşım, ağacın seviyesini 1’e indirgemeyi amaçlamıştır.



Şekil 4.5. Bir ASK kaydının HTML-DOM ağacı ile gösterimi

Şekil 4.4.'deki DOM ağacına ait basitleştirilmiş ağaç yapısı Şekil4.5.'de verilmiştir.



Şekil 4.6. Bir HTML-DOM ağacının seviyesinin indirgenmiş hali

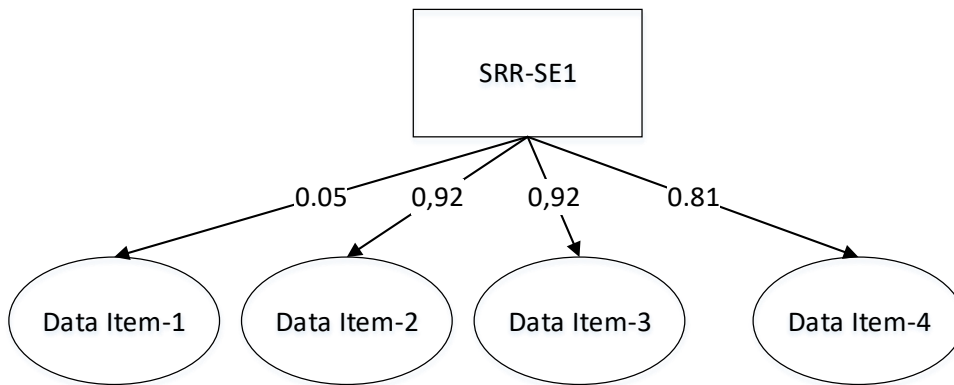
Yukarıda bahsi geçen dönüşümü yapabilmek için bir tip sistemine ihtiyaç duyulmaktadır. Bu çalışma kapsamında, ASK domaini için aşağıdaki şekilde bir Tip sistemi önerilmiş ve kullanılmıştır:

ASK Tip Sistemi = { Sayısal Değer, Fiyat, Tarih Zaman, E-Posta, URL, Bağlantı, Metin, HTML Zengin Metin }

Bu tip sisteminde yer alan Bağlantı ASK'lara özel bir tip olup, bir arama sonunun global Web'deki konumunu göstermektedir. Bütün aramada elde edilen sonuç bu bağlantı bilgisi olduğundan diğer veri tiplerinden farklı olarak ele alınması gerekmektedir.

Ağaç birleştirme işleminden sonra düzleştirilen ve tek seviyeye indirgenen ağaç yapısında Düzleştirilmiş Ağaç yapısı (Şekil 4.6.) denilmektedir. Bir Düzleştirilmiş Ağaç yapısı birçok alandan ve her bir alan için geçerli veri tipinden oluşmaktadır. Ayrıca her bir alan için, indirgeme sırasında yol üstünde geçen HTML ve CSS bilgileri ile HTML Tag bilgisi de tutularak, şema eşleme aşaması için kullanılmak üzere muhafaza edilmektedir.

Farklı arama motorlarından dönen arama sonuçlarının farklı şema yapısı tanımı bulunmaktadır. (Gozudeli ve diğerleri, 2015) çalışmasında bahsedildiği gibi her ne kadar düzensiz ASK ve düzenli ASK tanımı yapılmış olsa da bir tek arama sonuç sayfasında bile bir ASK'da var iken diğer bir ASK'da bulunmayan öğelere rastlanılmaktadır. Bu nedenle her bir veri öğesinin normalize edilmiş ASK'ya eklenerek, her bir öğenin sonuçlardan kaç tanesinde bulunduğu (cardinality) hesaplanarak veriler için Frekans Ağırlıklı öge ağacı kullanılmıştır (Şekil.4.6.).



Şekil 4.7. ASK'lar için frekans ağırlıklı öge ağacı

Parametrik bir kesme değeri kullanarak örnekte (0.05), bu frekanstan daha az tekrarlayan öğeler hesaba katılmamıştır.

Çalışmada, her bir şemanın bir diğeri ile 1:1 bir eşleme ile bağlanacağı varsayılmıştır. Farklı arama motorlarına ait şemaları birleştirmek için 4 farklı öznitelikten yararlanılmıştır:

- WCT: Her bir ASK ögesine ait verinin tipinin benzerliğini
- WLT: ögeye ait etiket ve etiket yol bilgisinin benzerliğini
- WCSF: Öge içerik boyut faktör benzerliğini
- WFS: öge kardinalite benzerliğini

İfade etmek üzere, şemaların benzerliği için eşitlik(4.1) den yararlanılmıştır:

$$W(A_i, B_j) = WCT + WLT + WCSF + WFS \quad (4.1)$$

İfadede geçen, A_i ve B_j değerleri farklı yerel şemalardaki veri öğelerini ifade etmektedir.

Performans açısından kıyaslandığında, HTML Ağaç Basitleştirme yaklaşımı, daha kısa bir çalışma zamanı sunmasına rağmen COMA 3.0 ile benzer bir sonuç kalitesi sergilemiştir.

Web Arama Sonuçlarının Otomatik çıkarımı ve çıkarılan sonuçların otomatik birleştirilmesi ile ilgili geliştirilen algoritmalar, genişletilebilir ve yeniden kullanılabilir bir API haline getirilmiştir. Bu amaçla, Java programlama dilinde Jsoup üçüncü parti açık kaynak kütüphanesi ve SQLite gömülü veri tabanı kullanılmıştır.

API'de 3 temel sınıf kullanılmıştır:

- RegionFinder sınıfı bir arama motoruna anahtar kelime gönderilmesi, sonuç sayfasının analiz edilerek, sayfa içerisinde sonuçların hangi bölgede olduğunu algılamaya yarayan algoritmayı içermektedir.
- SRRExtractor sınıfı yerel frekans ağaç yapısını oluşturarak farklı arama motorlarından gelen sonuçların birbirleri ile benzerliklerini algılayan algoritmayı içermektedir.
- ResultMerger sınıfı farklı arama motorlarından gelen sonuçların birleştirilmesi süreçlerini yöneten algoritmaları içermektedir.

API'ye ait örnek bir ekran görüntüsü Şekil 4.7.'de gösterilmiştir.



[import.io | Web Data Platform & Free Web Scraping Tool](https://import.io/)

<https://import.io/>

Turn the **web** into data, today. Transform any **website** into a table of data or an API in minutes without even writing any code with our free app.

[WebHarvy Web Scraper - Visual Web Scraping Software | Web Data ...](https://www.webharvy.com/)

<https://www.webharvy.com/>

WebHarvy can automatically scrape data (Text,Image,Email,URL) from **web** pages and save the scraped content in different formats.

[WebHarvest - web data extraction tool download | SourceForge.net](https://sourceforge.net/projects/web-harvest/)

sourceforge.net/projects/web-harvest/

Kullanıcı oyu: 4,5 - 21 oy - Ücretsiz - Linux
17 Apr 2013 ... WebHarvest - **web data extraction** tool download. WebHarvest - **web data extraction** tool 2013-04-17 11:14:24 free download. WebHarvest ...

[DEiXTo | Web data extraction made easy!](https://deixto.com/)

deixto.com/

DEiXTo (or ΔEiXTo) is a **web content extraction** tool that provides a GUI to build well-engineered **extraction** rules and a command line executor to apply them on ...

[Software for Web Scraping - Web Scraping](https://scraping.pro/software-for-web-scraping/)

scraping.pro/software-for-web-scraping/

There are many **web data extraction** applications and some cloud services available and they vary widely in cost and features. Here we've summarized them to ...

[Website Data Extraction | Automation Anywhere](https://www.automationanywhere.com/webdataext)

<https://www.automationanywhere.com/webdataext>

Extract Web data, screen scrape, harvest **web** data as well as **web mining** - no

Şekil 4.8. Sonuçların birleştirilmiş hali

ASK'ların otomatik çıkarımı ile ilgili algorithmada iyileştirme yapılmış, otomatik sonuç eşleme algoritması geliştirilmiş ve geliştirilen algoritmaların genişletilebilir, yeniden kullanılabilir bir API haline getirilmesi işlemleri tamamlanmıştır.

- Bilinen bir Web sonuç bölgesindeki kayıtların otomatik çıkarımını yapan bir yaklaşım geliştirilmiştir.

- ASK içeren HTML DOM Ağacının basitleştirilmesi için pratik bir yöntem önerilmiştir.
- Yapısal olmayan HTML verilerinin yarı yapılanmış veriye dönüştürülmesi için bir yöntem önerilmiştir.
- Algoritmalar için genişletilebilir ve yeniden kullanılabilir bir API modeli tasarlanmıştır.

5. TASARLANAN META-ARAMA MOTORU

Bu kısımda tasarlanan Meta-Arama Motorunun mimarisi ve literatürdeki dayanakları anlatılmıştır.

5.1. Geliştirilen Otomatik Meta-Arama Motoru Modeli

Bu tez çalışmasında, kullanıcıdan sorgu alımından itibaren sonuçların gösterimine kadar uçtan uca otomatik çalışabilen bir meta-arama motoru modeli geliştirilmiştir. Var olan arama motorları veya genel erişime açık, heterojen web veri kaynaklarını sorgulayarak sonuçları birleştirip kullanıcılara gösterirler. Ayrıca, genel erişime açık arama motorları veya Web veri kaynakları, girdi veya çıktı formatlarını sıklıkla değiştirebilmektedir. Bütün bu koşullar, programsal ortamlardan Web arama sonuçlarının heterojen kaynaklardan alınıp birleştirilmesini zorlaştırmaktadır. Bu türden bir otomatikleştirme meta-arama motorlarının yanı sıra, Web Demetleme (Mash-up), Web madenciliği ve sosyal Web madenciliği, Arama motorları gibi birçok bilgi çıkarım araçları ve bu araçlar üstünden sağlanabilecek, ürün, hizmet ve fiyat kıyaslama gibi web hizmetlerine kolaylık sağlamaktadır. Öte yandan, bu arayüzler (giriş ve çıkış) değişmeden de heterojen kaynakları ortak olarak sorgulamak için iyi tanımlı bir yol mevcut değildir.

Otomatik bir meta arama motorunun birçok bileşeni mevcuttur. Kullanıcı sorgusunun çözümlenmesi ve semantik çıkarımı, web arayüz eşleme, web sonuç çıkarımı, Web sonuç birleştirme, Web sonuç sıralama gibi. Her bir arama motoru Arama Sonuç Kaydı (ASK) denilen özgün formatta sonuçlar döndürür. Otomatik bir meta-arama motorunun farklı arama motorlarından dönen bu ASK'ları tekil bir şemaya birleştirip, bir dize kurala göre yeniden sıralaması beklenir.

Helios mimarisi arama motorları için bir arama ve sonuç gösterimi amaçlı Web arayüzü sağlamaktadır. Yerel Sorgu Çözümleyici alt arama motorlarına özgü sorgular oluşturabilmektedir. Motor Üretici arama motorları ile ilgili bağlantı ve benzeri bilgileri sağlamaktadır. HTTP Getirici, uzak arama motorları ile bağlantıyı gerçekleştirerek sonuçlarını getirir. Arama sonucu toplayıcı ve sonuç çözümleyici bileşeni XML

formatındaki sonuçları bir araya getirir. Helios ayrıca kullanıcının kendine özgü dönen sonuç birleştirme formatı girmesine izin vermektedir.

TadPole mimarisi de bir kullanıcı arayüzü sağlar. Her bir arama motoru için bir thread oluşturularak sonuçları eş zamanlı olarak birleştirebilmektedir. Bir thread makul bir zamanda sonuç döndürmezse bu arama motoru üstündeki arama sonlandırılır. Sorgu aşaması sonuçlandıktan sonra bütün sonuçlar yeniden sıralanarak sonuç penceresinde gösterilir.

Yatay arama motorları bütün Web'i indekslerken, düşey arama motorları sadece belli bir alana ait Web sayfalarını indekslemektedir. Kullanıcıdan gelen aramalar için bir alan tahmini yapabilmek, sorguların daha amaca uygun arama motorları üstünde çalıştırılması açısından önemlidir. Böylece amaca uygun olarak bazı düşey arama motorları sorguya dâhil edilebilir ya da sorgudan çıkartılabilir. Ancak sorgular için alan tahmini yapmak özellikle sorguların tam bir cümle olmaması nedeniyle oldukça zordur ve İsimlendirilmiş Varlık Tanımlama (İVT) gibi cümlenin öğelerine dayalı çıkarım yapabilen birçok Doğal Dil İşleme (DDİ) yöntemi işe yaramaz. Bu nedenle de bunun yerine Latent Semantic Analizi (LSA), Random Indexing (RI), Naïve Bayes ve Support Vector Machine (SVM) gibi Desen Tanıma algoritmaları kullanılır.

5.1.1. Çalışmanın literatür geçmişi

Bir otomatik meta arama motorunun birçok bileşeni olduğundan bu alandaki yapılan çalışmalar da meta arama motoru modelleri, otomatik Web çıkarımı ve otomatik şema eşitleme gibi birçok alana yayılmış olarak incelenebilmektedir.

5.1.2. Web arama sorgu ayrıştırması

Kullanıcı tarafından girilen bir web aramasının anlamsal yapısına karar vermek bir meta-arama motoru için sentaksına karar vermekten çok daha önemlidir. Web arama sorgularının anlamsal analizi hakkında birçok çalışma yapılmıştır (Huang, Chien ve Oyang, 2003; Kang ve Kim, 2003; Page, Brin, Motwani ve Winograd, 1998). İstatistiksel makine öğrenmesi, metin sınıflandırma algoritmaları ve doğal dil işleme bu alanda kullanılan tekniklerdir. Bu alanda yapılan çalışmaları (dos Santos ve Milidiú, 2012; Jain ve Pennacchiotti, 2010; Kang ve Kim, 2003) özetlemektedir. Web arama sorgularında en çok kullanılan DDİ tekniği

İVT'ye dayalı analiz edilmesidir. Sorguların semantik olarak analiz edilmesi uzun süredir üstünde çalışılmalayan bir alandır (Jones, 1971; Kang ve Kim, 2003). Arama sorguları farklı kriterlere göre gruplandırılabilir. Kullanıcının niyetine göre gruplandırma (örn.. bilgi amaçlı, gezinme amaçlı, işlem amaçlı) (Broder, 2002; Uichin Lee ve Cho, 2005). (2) Anlamsal Sınıflandırma (örn. Alışveriş amaçlı, günlük Yaşam, Eğlence amaçlı) (Beitzel ve diğerleri, 2005; Shen, Sun, Yang ve Chen, 2006).

Semantik temelli sorgu sınıflandırma çalışmaları iki temel grup etrafında şekillenmektedir: (i) Baf of Words, (ii) İVT temelli çalışmalar (Jones, 1971). Aslında İVT tek başına bir sorgu sınıflandırma tekniği değildir fakat ontolojik olarak varlıkları sınıflandırmak için Bag of Words'e göre oldukça başarılı bir yöntemdir. Öte yandan, yapılan bir çalışmada (Guo, Xu, Cheng ve Li, 2009) sorguların % 29'unun İVT içermediği gösterilmiştir.

İVT'nin Web sorgu analizlerinde kullanımı Paşca'nın çalışmasından (Paşca, 2007) sonra yaygın olarak kullanılmaya başlanmıştır. Paşca az sayıda İVT ile başlamış, daha sonra zayıf denetimli öğrenme ile yüz milyonlarca Web sorgusunu analiz ederek %88 doğrulukta aynı ontoloji altında milyonlarca İVT bulmanın mümkün olduğunu ortaya koymuştur. Bu çalışma ayrıca, web sorgu analizleri amacıyla bir test kümesinin nasıl oluşturulabileceği konusunda teknikler önermektedir. Benzer şekilde (Guo ve diğerleri, 2009) olasılık teorisi çerçevesinde Paşca'nın çalışması ve İVT'lerin bağlam içerisinde keşfedilmesini incelemiştir. Literatürde bunların dışında da Web arama log kayıtları üstünden Web sorgularının sınıflandırılması hakkında çalışmalar mevcuttur (Shen ve diğerleri, 2006) (Shen ve diğerleri, 2005).

5.1.3. Web sonuç çıkarımı

Web çıkarım süreçlerinin otomatikleştirilmesi, otomatik bir meta arama motorunun temel bileşenlerindendir. Web sonuç çıkarımı hakkında literatürde birçok çalışma mevcuttur ve halen de aktif olarak çalışma yapılan bir alandır. (Bergman, 2001; Gözüdeli, 2009; Laender, Ribeiro-Neto, da Silva ve Teixeira, 2002) farklı zamanlarda yapılmış çalışmalara ait metot ve yaklaşımları sınıflandırarak inceleyen literatür araştırmalarıdır. İlk çalışmalar Wrapper indükleme temelli çalışmalar iken, zaman içerisinde Web çıkarımının insan eli ile etiketlenmesine bağlı öğrenilmesi hedeflenmiştir. Bu türden yaklaşımlar otomatik bir Web çıkarımı için oldukça ilkel kalmaktadır. Yarı otomatik Web çıkarım metotları metin temelli

ve ağaç temelli yaklaşımlar etrafında şekillenmiştir. Wien (Broder, 2002) ve Stalker (Uichin Lee ve Cho, 2005) metin temelli tekniklerin başlıcalarıdır. Bu çalışmalarda, Web sonuç çıkarımları düz metinler olarak var sayılmış ve bu düz metinler üstünden, el ile etiketlenmiş ayrıcılara ait öğrenim kümeleri ile sistemin öğrenmesi sağlanmıştır.

W4F (Shen ve diğerleri, 2006) ve Wrapper (Beitzel ve diğerleri, 2005) ise Web dokümanlarını hiyerarşik nesne modeli (DOM) ağaçları olarak ele almıştır. Bu çalışmalarda da insan etkileşimine ihtiyaç duyulmaktadır. Takip eden çalışmalarda otomatik Web çıkarımları önerilmiştir. (Guo ve diğerleri, 2009) IEPAD (Paşca, 2007), MDR (Shen ve diğerleri, 2005), RoadRunner (Gulli ve Signorini, 2005), EXALG (Laender ve diğerleri, 2002), DEPTA, Tag of Path (Kang ve Kim, 2003) ve Content Density (Gozudeli ve diğerleri, 2014) bu çalışmalardan başlıcalarıdır. Bazı yaklaşımlar sadece bir sayfadan hareketle Web çıkarımı yapmayı hedeflerken bazı diğer çalışmalar birden fazla sayfadan öğrenme yapabilmektedir.

IEPAD, MDR ve Content Density tek bir sayfadan hareketle sonuç çıkarımının öğrenilmesini hedeflemektedir. IEPAD tekrarlayan metinleri işaretleyerek tanımlamayı hedeflemektedir. MDR benzerlik temelli agresif yaklaşım ile iki segmentin benzerliğini inceler. Content Density sonuçların sayfanın büyük bir kısmını kapsadığını ve düzenli sonuçların birbirleri ile benzer karakteristikte olması gerektiğini varsayar.

5.1.4. Otomatik ASK çıkarımı ve birleştirme

Web sonuç çıkarımı tek başına otomatik bir meta arama motoru için yeterli değildir. Bunun yanı sıra Arama Sonuç Kaydı (ASK) bileşenlerinin de bilinmesi gerekir. Daha sonra bu bileşenlere göre farklı arama motorlarından gelen ASK'ların birleştirilmesi gerekir.

Otomatik Web Çıkarım metotları, çıkarım yaparken yapıları ele alış biçimlerine göre iki farklı kategoride ele alınmaktadır: Görsel Blok Ağaçları (Visual Block Tree) ve Etiket İşleme (Tag Processing). Vide ve DWDE-R Görsel Blok Ağaçları (GBA) kullanırlar. GBA, bir Web sayfasının kullanıcı Web gezgininde görüntülediği şeklini tekrardan programatik olarak dikdörtgenlerden oluşan şekilde üreterek, her bir ögenin HTML ve Cascading Style Sheets (CSS) verilerinden yararlanarak en-boy bilgisi ve üstten-soldan uzaklığı gibi veriler ışığında Web sonuçlarını kestirmeye yarayan bir yapı üretir.

TagOfPath, RankingXPath, ContentDensitiy gibi çalışmalarda HTML Etiket işleme yöntemi kullanılmaktadır. Bu yöntemde, ASK'ların yerini bulmak için HTML-DOM ağacı özellikleri işlenmektedir. TagOfPath bütün ASK'ların birbiri ile tamamen aynı olduğunu, sapan ASK'ların bulunmadığını varsayar. ContentDensity ASK'ları düzenli ASK ve Düzensiz ASK olmak üzere iki farklı gruba ayırır ve sonuçta her iki türden ASK'yı da kapsar.

Yapısal seviye algoritmalar, şema eşlemek amacıyla yapısal bilgileri kullanırlar. Genel olarak farklı şemaları eşleştirmek için dairesel olmayan graf ve ağaç yapıları kullanılır. Graf temelli şemaların çoğu komşuluk benzerliklerini kullanarak benzer bireyleri bulmayı hedefler. Similarity Flooding algoritması buna örnek olarak verilebilir. Tree Edit Distance ve Graph Edit Distance algoritmaları, kaynak şemada çeşitli düğüm ekleme-çıkarma işlemleri ile hedef şemayı elde etmeye çalışırken arada gerekli işlem basamaklarının sayısını bir uzaklık ölçütü olarak kullanır. Karmaşıklık arttıkça iki graf ya da ağaç yapısının "Edit Distance" uzaklığını bulmak da zorlaşır.

Otomatik şema eşleme konusunda Cupit, CLIO, COMA, OLA ve S-Match gibi birçok çalışma yürütülmüştür. Bu platformlar birbirinden farklı algoritmaları bir arada kullanarak şema eşleme kalitesini artırmayı hedeflemişlerdir. COMA, bu platformlardan halen aktif olarak geliştirilmeye devam eden bir çalışmadır.

Çalışma kapsamında, bir tek arama sonuç sayfasından hareketle ASK'ların otomatik olarak alt öğeleri ile birlikte çıkartılıp, bir dönüşüm algoritması uygulayarak HTML-DOM ağaçlarının tabular veri formatına dönüştürülmesi hedeflenmiştir. Tabular verilerin eşlenmesi ağaç ya da graf yapılara göre oldukça kolay bir süreçtir.

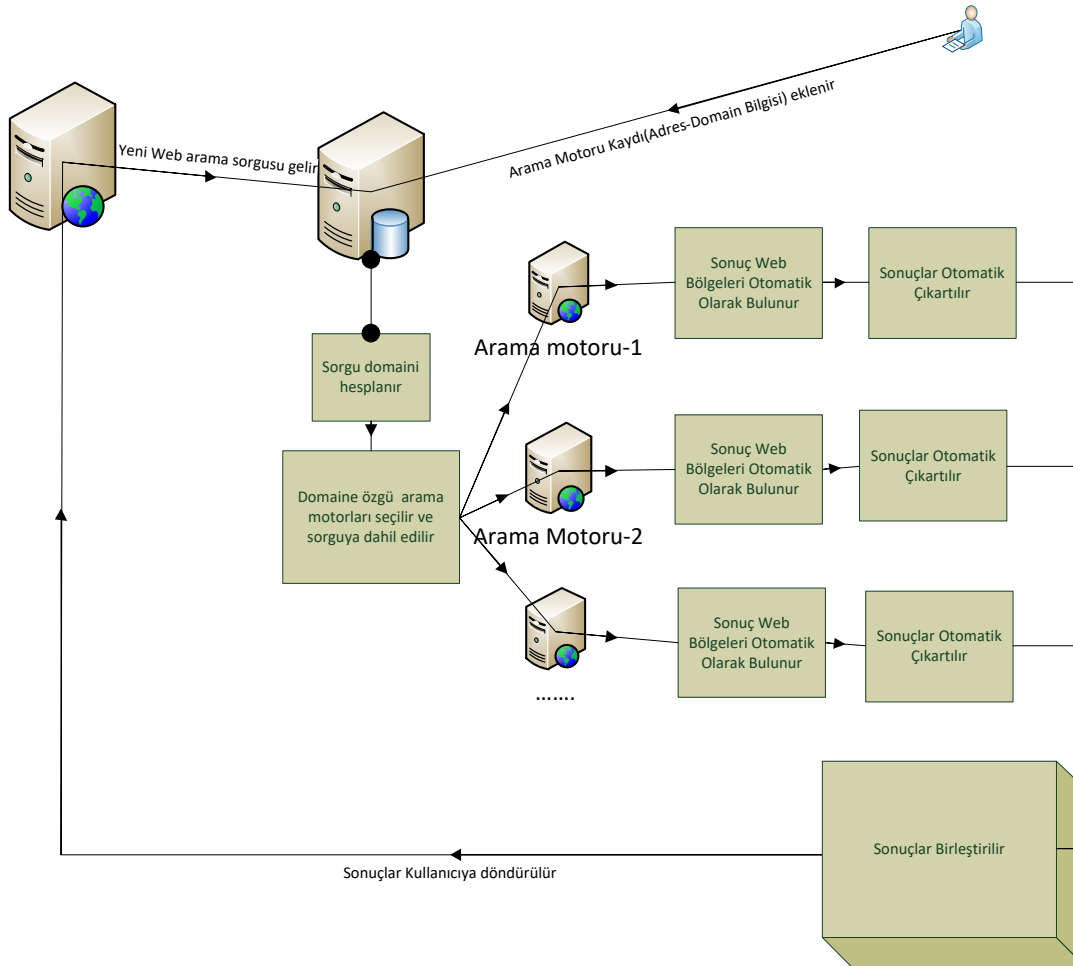
5.1.5. Meta arama sonuç sıralaması ve rank

Sonuçların birleştirilmesi aşamasında, bulunan ASK'ların yeniden sıralamaya tabi tutulması başka bir çalışma alanıdır. Sorgu ile sonuçlar arasındaki benzerlik genellikle arama motorlarına özgüdür. Bu nedenle de (Jones, 1971; Page ve diğerleri, 1998) arama motorlarından gelen sonuçları doğrudan kullanmazlar. Bunun yerine ASK'lar için normalize edilmiş olarak üretilmiş rank değerleri kullanılır. Bazı diğer yaklaşımlar ise benzerliği de

kullanmayıp kendileri yeniden ASK ve sorgu arasındaki benzerliği hesaplayarak kullanırlar (Mahathi S. ve Singitham, 2002).

5.2. Model ve Bileşenler

Yapılan çalışma kapsamında yeni bir otomatik meta arama motoru modeli önerilmiştir. Model, zayıf öğrenmeli bir Web sorgu çözümleme ve sınıflandırma birimi, otomatik Web sonuç bölgesi kestirim ve çıkarım birimi ile otomatik Web sonuç birleştirme ve sıralama birimlerinden oluşmaktadır. Model aşağıdaki şekilde yer almaktadır.



Şekil 5.1. Otomatik meta arama motoru modeli

5.2.1. İVT ile web arama sorgusunun anlamlandırılması ve sınıflandırılması

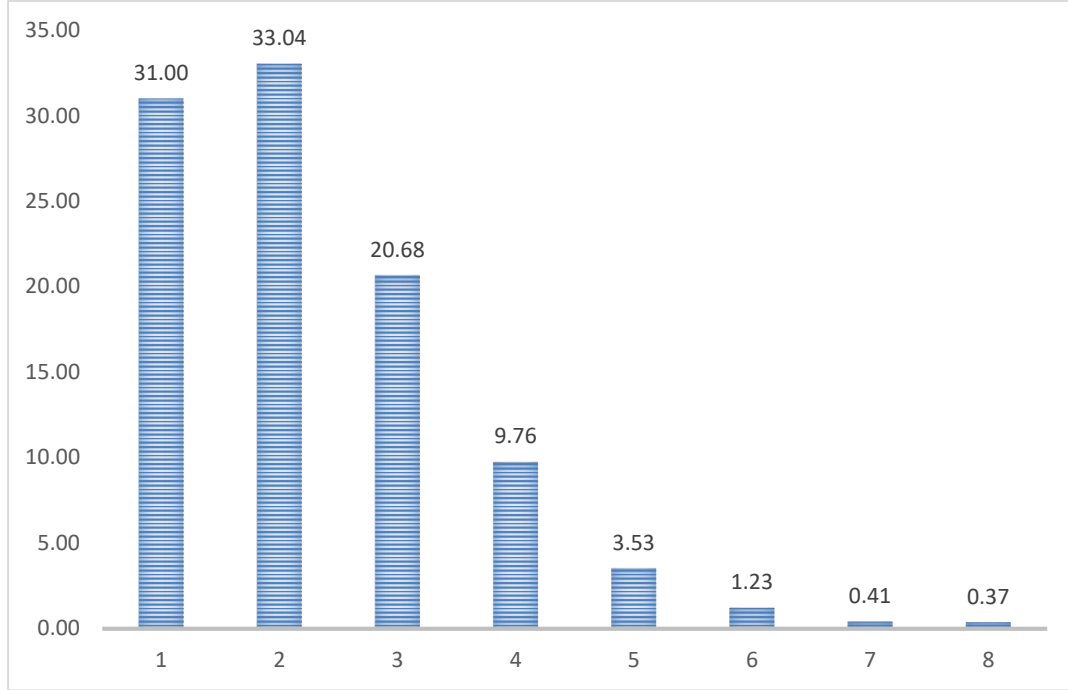
Kullanılan Web sorgu veri kümesinde toplam 36,389,567 satırda 10,154,742 tekil sorgu yer almaktadır. Araştırmaya tüm loglar dahil edilmemiş, sadece ilk 500.000 satırlık bir kısmı dahil edilmiştir.

Çekirdek olarak her bir domain için aşağıdaki İVT ve anahtar kelimeler ile başlanmıştır.

- Ürün (dvd, aqua glycolic, hat cap, mp3 player, corel draw 12, motorola razr v3, wgr614, swarovski, canon sd450, v5005us)
- Film (lord of the rings, back to the future, the pianist, toy story, the Shawshank Redemption, The Godfather, the dark Knight, Angry Man, Schindler's List, Fight Club)
- Kitap (Biggest Book of Slow Cooker Recipes, angels and demons, the house of the spirits, The Chocolate War, autumn letters, fahrenheit 451, guests of the sheik, The Kite Runner, what i believe, A Million Little Pieces)
- İnsan (micheal Jackson, steve jobs, mel gibson, bill gates, albert einstein, ataturk, abraham lincoln, bill clinton, sultanahmet, van gogh)
- Oyun (super mario, sim city, resident evil, prince of persia, battlefield 1942, hooking slimline, fifa 2006, need for speed, medal of honor, westwood, age of empires)

Veri kümesi arama yapan oturum bilgisinin anonimize edilmiş hali, yapılan arama sorgusu, aramanın zamanı, hangi sırada dönen sonucun tercih edildiği ve hangi adrese gidildiği bilgilerini içermektedir.

Arama veri kümesinde, her bir sorguda geçen kelime sayısı incelenmiştir. Aşağıda histogram grafiği kelime sayısına göre sorguların yüzdelik dağılımlarını göstermektedir.



Şekil 5.2. Sorgulardaki kelime sayılarına göre frekans yüzdelikleri

5.2.2. İVT'lerden kalıpların hesaplanması

Arama loglarında, çekirdek olarak alınan İVT'ler her bir domain için teker teker aranmış ve yer aldıkları her bir log İVT'den öncesi (B) ve İVT'den sonrası (A) olmak üzere iki farklı bağlama ayrılmıştır. Herhangi bir log için

$$q = (B, \text{İVT}, A) \quad (5.1)$$

ile tanımlı iken, q kullanıcıdan gelen Web arama sorgusunu, B İVT'den önceki bağlamı, A İVT'den sonraki bağlamı ifade etmek üzere, bulunan her bir İVT'den hareketle, loglarının kendi içinde benzerlik matrisi aşağıdaki şekilde hesaplanmıştır.

$$S = S(B, B') + S(A, A') \quad (5.2)$$

Arkasından B ve A matrislerinin her biri için, öncelikle yer alan kelimelerdeki önemsiz kelimecikler (the, on, to, gibi) çıkartılmış daha sonra her iki matrisin kendisi ile arasındaki uzaklık matrisi Levenshtein Distance algoritması kullanılarak hesaplanmıştır.

Levenshtein Distance algoritması, A matrisinin B matrisine uzaklığı (a ve b=a') için $A=a_1.....a_n$ ve $B=b_1.....b_m$ ile tanımlı iken d_{nm} ($m=n$ olmak üzere) şu şekilde hesaplanmıştır:

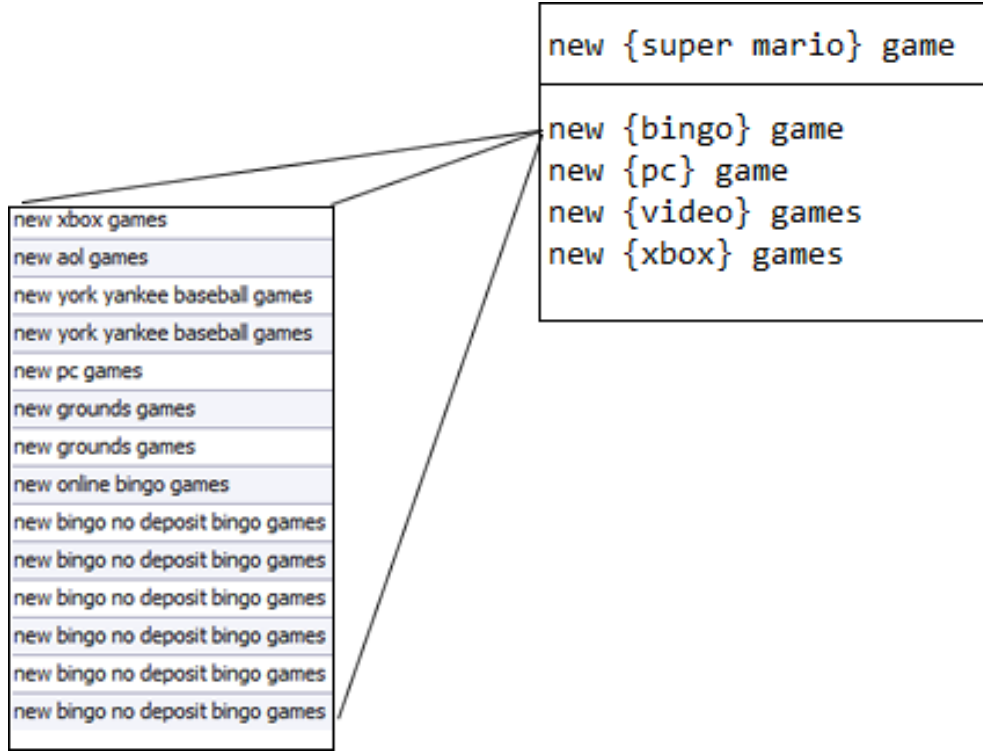
$$\begin{aligned}
 d_{i0} &= \sum_{k=1}^i w_{del}(b_k) & \text{for } 1 \leq i \leq m \\
 d_{0j} &= \sum_{k=1}^j w_{ins}(a_k) & \text{for } 1 \leq j \leq n \\
 d_{ij} &= \begin{cases} d_{i-1,j-1} & \text{for } a_j = b_i \\ \min \begin{cases} d_{i-1,j} + w_{del}(b_i) \\ d_{i,j-1} + w_{ins}(a_j) \\ d_{i-1,j-1} + w_{sub}(a_j, b_i) \end{cases} & \text{for } a_j \neq b_i \end{cases} & \text{for } 1 \leq i \leq m, 1 \leq j \leq n
 \end{aligned} \tag{5.3}$$

$$\varepsilon = 1 - \frac{D_{ij}}{\text{Max}(|a_i|, |b_j|)}$$

Belli bir eşik değerin üstündeki ($\varepsilon = 0,25$) benzerlikler, yeni kalıp öbekleri olarak alınmıştır.

5.2.3. Kalıp öbeklerinden yeni İVT'lerin keşfedilmesi

Bir önceki aşamada elde edilen A ve B öbekleri, arasında yer alan kelimeler bulunmuş, bu kelimelerden tekrarlayanlar yeni İVT'ler olarak tespit edilmiştir. Yukarıdaki işlemler, 50 İVT elde edilinceye kadar devam ettirilmiştir. Her bir domain'de yer alan İVT ile ilişkili keşfedilen İVT'lerin geçtiği sorgular, ilgili domaine(sınıfa) ait sorgulara dâhil edilmiştir. Şekil 11 'de "super mario" İVT'sinden hareketle bulunan *new {....} game* kalıbı ve bu kalıpta hareketle bulunan diğer İVT'ler gösterilmiştir.



Şekil 5.3. İVT’ler aracılığıyla arama loglarının öğrenilmesi

5.2.4. Tıklanan web adresi bilgisinin öğrenilen loglara dâhil edilmesi

Web arama sorgu logları, her bir sorgudan sonra ziyaretçi tarafından tıklanan Web sitesinin alan adı bilgisini de içermektedir. Alan adları, belli kelime grupları için gidilen web adresini göstermektedir. Bu siteye ait bilgilerin de sınıflandırmaya dâhil edilmesi birçok çalışmada başvurula gelen bir yöntemdir (Jain ve Pennacchiotti, 2010; Shen ve diğerleri, 2006). Ancak bu çalışmalarda Web sitesinin içeriğine ulaşarak bir içerik ögesi olarak değerlendirilmekte ve anlamsal olarak değerlendirilmeye katılırken, yapılan çalışmada bir istatistiksel öge olarak domaine ait yeni kelime öbeklerine ulaşmak amacıyla kullanılmıştır.

Yapılan çalışmada, her bir İVT’ye karşılık gelen alan adı bilgileri frekanslarına göre gruplandırılarak en çok tıklanan 3 web adresine karşılık gelen arama logları da ilgili domaine ait gruba dâhil edilmiştir. Böylece, ziyaretçilerin sonuç sayfasındaki davranışları sözcük hazinesinin genişletilmesine dâhil edilmiştir. Şekil 12’de “super mario” İVT’sinden hareketle öğrenilen 3 Web adresi gösterilmiştir.

TIKLANAN WEB ADRESİ	FREKANSI *
http://www.gamewinners.com	8
http://www.gamespot.com	7
http://www.mofunzone.com	6
http://www.neoseeker.com	4
http://www.freeonlinegames.com	4
http://cheats.ign.com	3
http://en.wikipedia.org	3
http://members.aol.com	3
http://www.acid-play.com	2
http://games.byond.com	2

Şekil 5.4. Super Mario İVT'si için tıklanan web adresleri

Şekil 5.4.'e göre ilk 3 satır eşik değerin üstünde bir tıklanma aldığı için domain açısından daha kuvvetli bir özellik olarak ele alınmaktadır.

5.2.5. Arama sorguları için domain öğrenme aşamaları

Bir önceki aşamada sınıflara ayrılan Web arama sorgu logları bu aşamada öncelikle Multinomial Naive Bayes kullanan otomatik kategorizasyon sistemine özellik kümesi olarak verilmiştir. Naive Bayes sınıflandırıcısı, sınıflandırmadaki özelliklerin birbirinden bağımsız olduğunu varsaysa da Manning ve arkadaşları tarafından yapılan bir çalışmada (Gozudeli ve diğerleri, 2014) düşük olasılık kestirimine rağmen metin sınıflandırma konusunda oldukça başarılı olduğunu ölçümlenmiştir.

Web sorgularının sınıflandırılması bir metin sınıflandırma problemi olduğundan, kelimelerden hareketle sorguları belli sınıflara dâhil etmek mümkün olabilmektedir. Öncelikle sınıflara ayrıştırılan kelime grupları üstünden her bir sınıfın ihtimali hesaplanır.

$$P(c) = \frac{N_c}{N} \quad (5.4)$$

(5.4)'de $P(c)$ sınıfına ait ihtimal hesabı, C sınıfına ait kelimelerin adedi olan N_c değerinin toplam terim sayısını gösteren N 'e oranı ile hesaplanır.

Multinomial Naive Bayes, Naive Bayes'ten farklı olarak kelimelerin tekrarlarını da sınıflandırmada bir özellik olarak kullanır. Buna göre bir w teriminin c sınıfına benzerliği (likelihood) (5.5)'deki gibi hesaplanmaktadır.

$$P(w|c) = \frac{\text{count}(w,c) + 1}{\text{count}(c) + |V|} \quad (5.5)$$

Bu eşitlikte, $\text{count}(w,c)$ w teriminin c sınıfında geçme sayısını, $\text{count}(c)$ c sınıfındaki toplam terim sayısını göstermektedir. $|V|$ değeri ise toplam tekil terim sayısını göstermektedir.

Eşitlik (5.4) ve (5.5)'den hareketle, bir t terimli bir q sorgusunun her bir sınıfa ait olup olmadığı (5.6)'daki eşitlik ile hesaplanmış ve en yüksek değer döndüren sınıfta olduğu varsayılmıştır.

$$P(c|q) = P(c) \prod P(w_i|c) \quad (5.6)$$

5.2.6. Web sonuç çıkarımı

Meta-arama motorunun farklı arama motorlarından gelen sonuç sayfalarından çıkarım yaparak sonuçları birleştirmesi gerekir. Bu bileşenin temel hedefi verilen bir Web arama sonuç sayfasından hareketle, aday sonuç listelerini önerebilecek bir algoritma geliştirilmesidir. Arama sonuçlarını bir HTML DOM Ağacından otomatik olarak bulup çıkartmak bir DOM Ağacı işleme problemidir. Algoritmanın detaylarına önceki çalışmamızda (Gozudeli ve diğerleri, 2014) yer verilmiştir. ASK'lar düzenli ve düzensiz ASK olarak aynı çalışmada gruplandırılmıştır.

Web sonuç çıkarımı, görsel özelliklerle ilgili bir kısım varsayımları gerekli kılar. Olağan bir arama motorunun arama sonuç sayfasının önemli bir kısmını arama sonuçlarına ayırması beklenir. Bu nedenle, ASK'ların içerik yoğun kısımlar etrafında konumlanmış olması beklenir. Bir çalışmada (Linner, 2008), bir sonuç bölgesinin sayfa içeriğine oranının yaklaşık

olarak 0,4 civarlarında olduğu varsayılmıştır. Yapılan deneylerde ise bu oranın yer yer 0,2'ye kadar düşebildiği görülmüştür.

Hedeflenen yaklaşım dört aşamadan oluşmaktadır.

1. İçerik özellik vektörü(İÖV) hesaplanır(İçerik Ölçü faktörü, Alt düğüm sayısı, Link yoğunluğu)
2. Aday düğümler, İÖV kümelmesi ile elde edilir.
3. İÖV hesabı ile kümelemenin dışında kalan düzensiz düğümler yok sayılırlar.
4. En iyi düğümler aday düğüm olarak döndürülür.

İçerik Ölçüm Faktörü, içeriğin kapladığı alan ile ilgili hesaplanabilen yüzeysel bir bilgi olup metin içerik için font büyüklüğü ve içeriğin uzunluğu gibi bilgilerle hesaplanabilirken, resim içerik için en ve boy bilgilerinden elde edilebilmektedir. İÖV temelli link yoğunluğu aşağıdaki denkleme göre hesaplanabilmektedir.

$$LinkYogunlugu = \frac{LinkSayisi}{AltDugumSayisi + 1} \quad (5.7)$$

Tablo-1'de, ilk ve ikinci satırların düzensiz takip eden satırların ise düzenli SRR olma ihtimali yüksek görünmektedir.

Çizelge 5.1. Örnek bir arama için oluşturulmuş aday düğüm

Düğüm Yol Bilgisi	Alt Düğüm Sayısı	İçerik Ölçü Faktörü	Link Yoğunluğu
table/tbody/tr/td/div/ol/li	3	4253	10,00
table/tbody/tr/td/div/ol/li	1	2548	35,00
table/tbody/tr/td/div/ol/li	2	1186	6,67
table/tbody/tr/td/div/ol/li	2	1340	6,67
table/tbody/tr/td/div/ol/li	2	1441	6,67

İÖV hesaplandıktan sonra Tree Edit Distance algoritması ile aday düğümler arasından en çok birbirine benzeyen alt düğümü olan aday sonuç bölgesi olarak seçilmektedir.

5.2.7. Web sonuç birleştirme

Bu bileşenin temel hedefi Web sayfası içerisinde yeri belirlenmiş olan sonuç bölgesinin otomatik olarak çıkarımını yapıp farklı arama motorlarından gelen sonuçları birleştirmek üzere bir algoritma önerilmesidir.

Özet algoritma akışına ait kaba kod aşağıda yer almaktadır:

```

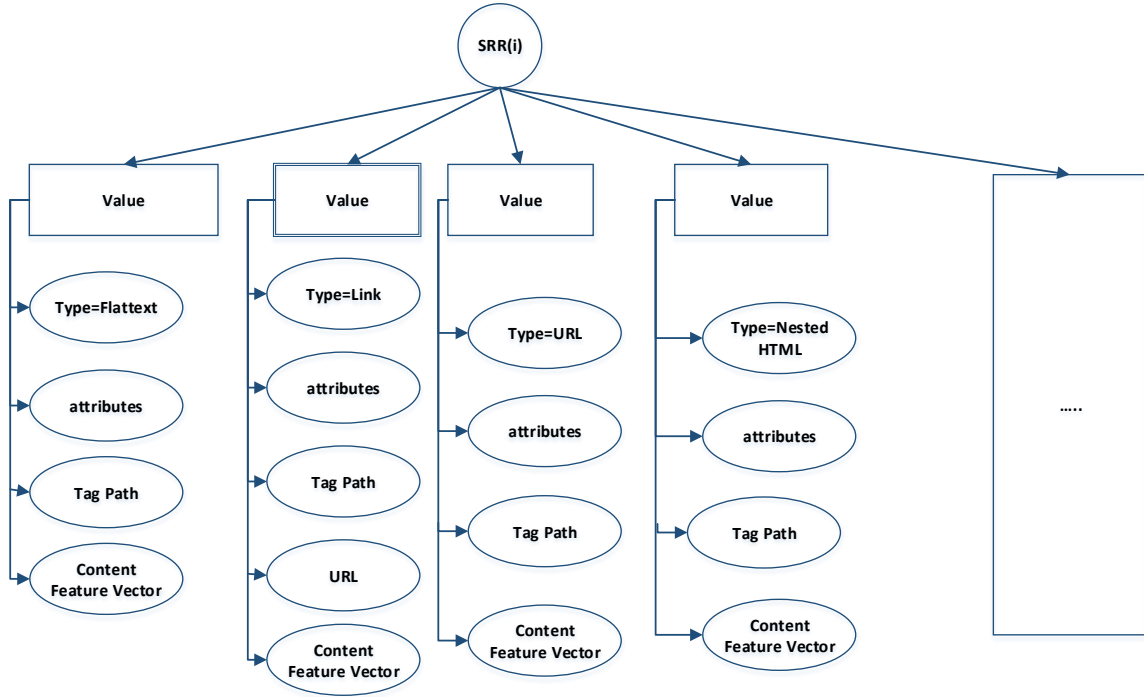
Algorithm overall follow
Input: test keyword, SE query url, SRR Data Region information
begin
    call retrieveSearchResult(keyword);
    call getSEURL();
    call getSRR_Region();
    foreach searchResults
        call findMaxLeafLevelContentSegmentItem()
        call detect dataTypeOfSegment()
        call collectLabelData()
        call flattenHTML_DOM_Tree()
    end
    foreach SearchEngineSource
        call weightedLocalTree()
    end
    call generateGlobalTree(weightedLocalTree1, weightedLocalTree2)
    foreach weightedLocalTreei
        rearrangeGlobalTree(weightedLocalTreei)
    end
end

```

Şekil 5.5. Web Çıkarım genel akış algoritması

5.2.8. ASK ağacı derinlik azaltma algoritmasının sisteme eklenmesi

Dikdörtgensel veri şema eşleme açısından oldukça kolay yönetilebilir ve işlenebilir bir formattır. Öte yandan HTML DOM ağacı ise dikdörtgensel veriye göre eşleme açısından oldukça fazla olasılık barındırdığından şema eşleme öncesi dikdörtgensel veriye dönüştürülmesi daha eski çalışmalarla (Gozudeli ve diğerleri, 2015) performans kıyaslaması ile önerilmiştir. Benzer şekilde, dal seviye düğümler üstünden HTML DOM ağacı dolaşarak dikdörtgensel veriye dönüştürülmüştür.

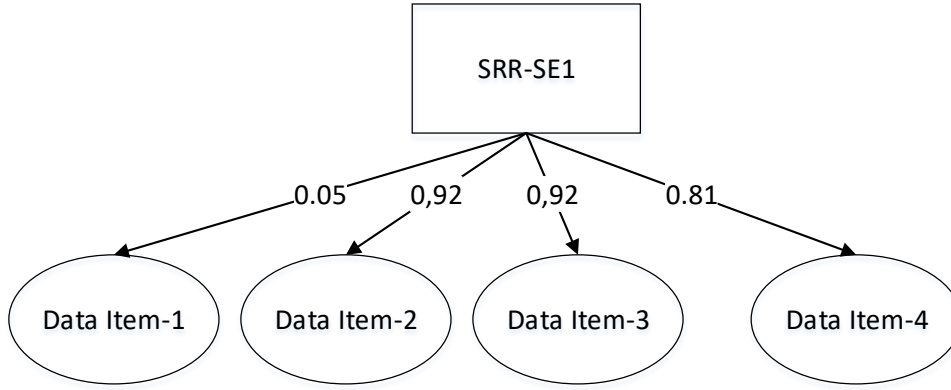


Şekil 5.6. Ağaç derinliğinin azaltılması

Dikdörtgene dönüştürme ile ağaç derinliği 1 ile sınırlandırılmış olmaktadır. Ancak bu amaç için bir tip sistemine ihtiyaç duyulmaktadır. Daha önceki çalışmalarda (Gozudeli ve diğerleri, 2015) kullanılan tip sistemine benzer bir tip sistemi bu çalışmada da kullanılmıştır: ASK Tip = {URL, Web Link, Numerical Data, Date, Email, Price, Simple Text, Nested HTML}

5.2.9. Şema eşleme sistem bileşeni

Her bir arama motorundan elde edilen sonuçları karşılar şekilde yerel bir ASK ağacı oluşturulmuştur. Bunun için rastlanan bütün farklı ASK öğelerinin frekansına bakılarak sonuç ağacında yer alıp almayacağına karar verilmiştir.



Şekil 5.7. Örnek bir Frekanslandırılmış ASK öge ağacı

Farklı arama motorları için elde edilen Frekanslandırılmış ASK kayıtlarının birleştirilmesi için şema eşleme fonksiyonundan faydalanılmıştır. Şema eşitlemede öğelerin bire bir ilişkili olduğu varsayımı yapılmıştır. Bu amaç için (4) ile verilen formül kullanılmıştır.

5.2.10. Web sonuçlarının sıralanması

Alan (Domain) Benzerlik skoru ve Arama Sonuç Kaydı (ASK) benzerliğine dayalı yeni bir ranking algoritması geliştirilmiştir. Meta-Arama motoru, öncelikle kullanıcıdan aldığı arama sorguları için bir alan ve arama motoru tahmininde bulunur.

Her bir arama motoru, ASK'lar içerisinde buldukları sonuçlara ait kısa bir temsil niteliğindeki metni sunduklarından ASK'ların domain ve kelime benzerliklerinin değerlendirilmesi, dokümanların sorgu ile benzerlikleri hakkında da bilgi vermektedir. Bu kuraldan hareketle her bir arama motorundan dönen ASK'lar, domain benzerliği ve arama sorgusu ile benzerliklerine göre derecelendirilmektedir. Benzerlik sıralaması algoritmasına benzer şekilde ASK (A) ve sorgunun alan D ve sorgu q olmak üzere, herhangi bir ASK için rank değeri şu şekilde hesaplanmaktadır:

$$DomainBenzerlik(A, q, D) = C_1 \frac{Ds - \text{Min}(Ds)}{\text{Max}(Ds) - \text{Min}(Ds)} + (1 - C_1) * Benzerlik(q, A) \quad (5.8)$$

C_1 , katsayısı domain benzerliği ile doğrudan sorgu benzerliği arasındaki ağırlığı belirlemek üzere sabit bir değer olarak alınmıştır.

5.2.12. Önerilen sıralama algoritması: SemiRank

Çalışmada önerilen yeni rank yönteminde, arama işleminin doğasına uygun olarak i) kullanıcıdan gelen sorgu ve sonuçların semantik ve kelime bazlı uyumuna ilişkin veriler ve ii) verilerin elde edildiği veri kaynaklarının verilerin ne kadar ‘güvenilir’ olduğuna dair verileri esas alınmıştır.

Bu iki kavramdan ilki bir önceki rank formülü (5.9) ile sağlanırken, kaynakların veri hakkındaki kestirimleri de sonuca katılmak üzere:

1-Arama sorgusu ile ASK arasındaki semantik benzerlik

- ASK genel- sorgu benzerliği
- ASK başlık-sorgu benzerliği
- ASK URL-sorgu benzerliği
- Sorgu ve sonuçlar arasındaki Semantik Domain Benzerliği

2-ASK-kaynak-rank verileri:

- Her bir ASK’nın arama motorları sonuçlarındaki orijinal rank değeri,
- ASK-tekrarlama sayısı

Bu iki değer, her bir ASK’nın arama motorlarından kaç tanesinde yer aldığı ve hangi sırada yer aldığı değeri, Akritidis ve arkadaşları tarafından yapılan çalışmada (Akritidis ve diğerleri, 2011) sonuçların veri kaynağındaki sıralarına göre rank değerlerinin hesaplanması (5.10) daki eşitlikten hareketle hesaplanmıştır.

$$K(c) = \sum_{i=1}^m (k + 1 - r_i(c)) \quad (5.10)$$

m: Arama motorlarını göstermek üzere, $r_i(c)$ c sonucunun i.sıradaki arama motorunda aldığı rank’ı göstermektedir. k katsayısı her bir değeri sabit tutmak için kullanılmıştır. Elde edilen bu verilerde farklılıkları azaltmak için (5.11) de gösterildiği şekilde Logaritmik filtreden geçirilmiştir.

$$R(c) = m \cdot \log(n_c K(c)) \quad (5.11)$$

Son olarak (5.10) ve (5.11) eşitlikleri ile kaynak rank ve tekraralama sayısı sıralamaya dâhil edilmiştir. Ancak ASK'ların sıralanmasında zaten veri kaynağı arma motorları TF/IDF değerlerinden yararlanılmaktadır. (Akritidis ve diğerleri, 2011) gibi çalışmalarda bu değerlerle tekrardan kayıtların sıralanmasının ilgililik sonuçlarını artırmaya katkısı görecelidir. Öte yandan, bütün arama sorguları için uygun bir domain bilgisi bulundurmamak bir meta-arama motoru için mümkün olmamaktadır.

Arama motoru veri kaynaklarından ilk gelen verilerin sıralamada yukarıda yer alması, güvenilirliklerinin yüksek olmasındandır. Sorgu domainin meta-arama motorunda bulunamadığı hallerde, her bir arama motorundan gelen ilk kayıt doğruya en yakın kayıt kabul edilmiş ve bu sonuçlardan elde edilen bilgilerle yeni bir domain eğitim seti oluşturulmuştur. Oluşturulan bu küme üstünden takip eden ASK'larının domaine benzerliği ölçülünerek sıralamaya dâhil edilmiştir. Bu durumda, *DomainBenzerlik* değeri (5.11)'deki formül ile ilk gelen ASK'lardaki kelimeler kullanılarak eğitilmiş domaine göre hesaplanmıştır. Bütün bu sıralamadan sonra SemiRank (5.12)'de gösterilen şekilde hesaplanmıştır:

$$SemiRank(i) = k \cdot DomainBenzerlik + (1 - k) \cdot R(c) \quad (5.12)$$

k domain benzerlik ile Veri Kaynaklarından gelen özelliklere dayalı sıralama arasındaki oranı belirlemek üzere kullanılmıştır.

5.3. Kullanılan Mimariler ve Sistem Özellikleri

Yazılımların tamamı Java dili ve kütüphaneleri ve Eclipse IDE ortamı kullanılarak gerçekleştirilmiştir. Web Çıkarım Kütüphanesi olarak Jsoup açık kaynak kütüphanesinden yararlanılmıştır.

Cossine Benzerliği, Multinomial Naive Bayes gerçeklemesi gibi algoritmalar için datumbbox Makine Öğrenmesi açık kaynak kütüphanesinden yararlanılmıştır.

Veri Tabanı Yönetim Sistemi olarak SQLite kütüphanesinden yararlanılmıştır. SQLite gömülü bir Veri Tabanı Yönetim sistemi olup seçme ve sıralama işlemlerini kolaylaştırmak için tercih edilmiştir.

Testlerde arama motorlarının DDoS saldırı önlemlerine takılmamak amacıyla öncelikle sorgular için sonuçlar bir sefer çekilerek yerel veri tabanına kaydedilmiş, daha sonra bu veri tabanı üstünden uzak veri-kaynakları simüle edilerek uygulamalar test edilmiştir.

Test Yapılan Sistem Kaynakları aşağıdaki şekildedir:

- 9750H (6 Core-12 Thread) Intel CPU.
- 24GB Bellek.
- M2 SSD -512GB.

6. DENEYSEL SONUÇLAR

Tasarlanan otomatik meta arama motoru üstünde sorguların otomatik sınıflandırılması, sonuç sayfalarındaki Web bölge tespitlerinin doğruluğu, ASK'ların çıkarımı ve eşitlenmesi ile sonuçların sıralanmasına ilişkin çalışmalar yapılmıştır.

6.1. Arama Sorgularının Sınıflandırılmasına İlişkin Başarımlar

Kullanılan Meta Arama Motorları şu şekildedir;

Yatay Arama Motorları

(ülkemizdeki kullanım oranları dikkate alınarak tercih edilmiştir)

- Google
- Yandex
- Bing

Düşey Arama Motorları ve Domain Bilgileri

- İconfinder (resim)
- İmdb (film) (artist) (insan)
- wenZher(bilim)
- twitter (insan)
- Amazon (ürün) (film) (kitap)
- Ea.com (oyun)

Önceden belirlenmiş farklı domainlere ait 10 sorgu için geliştirilen sistem test edilmiştir. Bir sonraki aşamada, son kullanıcı testleri ile sorgu sınıflamalarının kesinlik ve duyarlık başarımlarının tespit edilmesi planlanmıştır.

Tahmin edilen domainlere ait bir örnek Çizelge 6.1.'de yer almaktadır. (Hatalı olan domain hesaplamaları * ile gösterilmiştir.) Şekle göre, “the catcher in the rye” araması için hesaplanan domain Kitap olması gerekirken Ürün olarak hesaplanmıştır.

6.2. Web Sonuç Bölgesinin Bulunması

Test için daha önceki çalışmalarda kullanılan veri kümesi kullanılmıştır. Bu veri kümesinde farklı alanlardan toplanmış 500 örnek arama kelimesi yer almaktadır. Bu alandaki testlere sadece yatay arama motorları dâhil edilmiştir.

Test kümesinin farklı domainlere dağılımı şu şekildedir:

- Google Search Trends (150 sorgu)
- AOL Search Trends (100 sorgu)
- Indeed.com (10 trending job sorgusu)
- Amazon.com (en çok satılan 15 ürün)
- Ebay (en çok aranan 20 kelime)
- Twitter (5 trend konu)
- IMDB (200 film adı)

Web sonuç bölgesinin bulunması için yapılan deneylerde, precision ve recall değerleri (4.1) ve (4.2) ile hesaplanmıştır:

6.2.1. ASK bölgesinin bulunmasına ilişkin başarımlar

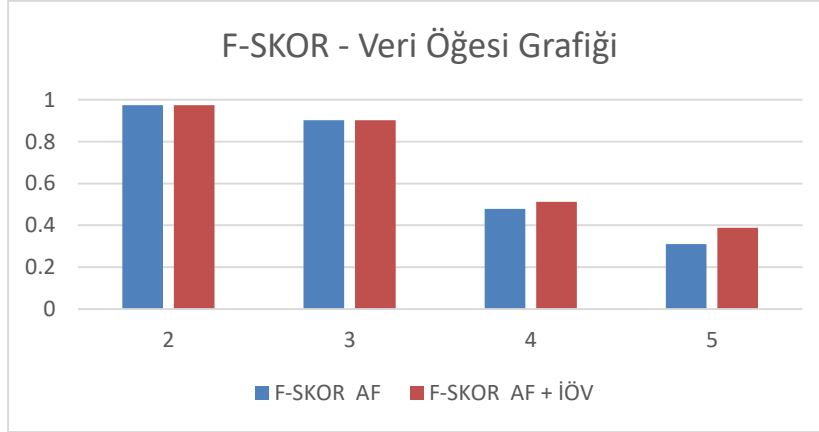
Bu amaçla sadece precision ve recall hesaplanmıştır. Precision ve Recall değerleri, el ile verilen sonuç bölgesindeki kayıtların otomatik sonuç çıkarım algoritması ile elde edilen sonuçlarda var olup olmaması durumlarına göre hesaplanmıştır.

Çizelge 6.2. Farklı arama motorlarına ilişkin precision ve recall değerleri

	Precision	Recall
Google	%98,3	%97,1
Bing	%97,5	%96,4
Yandex	%90,6	%87,8

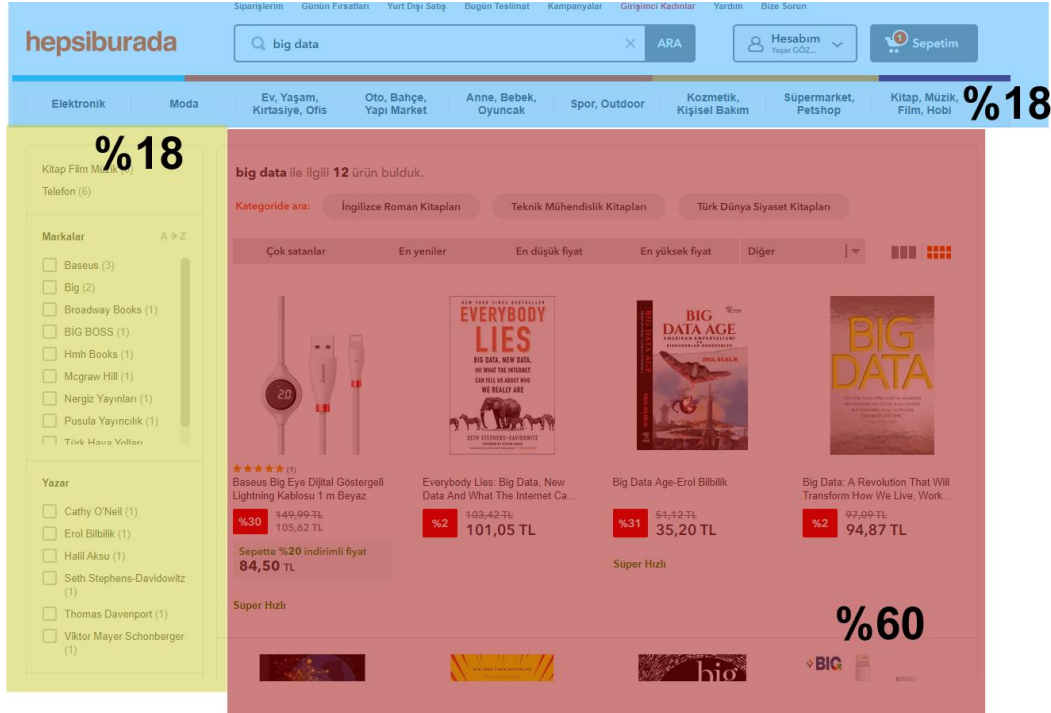
6.2.2. ASK öge çıkarım ve eşleştirmeye ilişkin ölçüm sonuçları

Bu amaçla f-skor ölçümlenmiştir. Ağaç basitleştirme algoritmasına ek olarak İçerik Özellik Vektörü (İÖV)'nin başarıma katkısı aşağıdaki şekilde ölçümlenmiştir.



Şekil 6.2. Eşleştirme başarımında ASK öge sayısına bağlı F-Skor grafiği

İlk üç veri ögesi için algoritmanın seçme başarımı oldukça güçlü iken öge sayısı arttıkça başarımları hızla azalmaktadır. Bu düşüşün nedeni, veri kümesindeki öğelerin benzerliklerinin azalmasına bağlı olarak otomatik eşleştirmenin zorlaşması ile ilgilidir. İçerik Özellik vektörü özellikle artan öge sayısına bağlı olarak sonuçta bir iyileşme sağlamıştır. Bu iyileşme daha fazla bilgidir hareketle daha başarılı eşleştirme yapılabildiği şeklinde yorumlanmıştır.



Şekil 6.3. Bir arama sonuç sayfasında sonuçların gösterildiği bölge alan oranları

6.3. Sonuçların sıralanmasına ilişkin başarımın ölçülmesi

Sıralama algoritmasının ve domain odaklı veri eklemenin etkisini ölçmek için (Gozudeli ve diğerleri, 2015)'de kullanılan örnek sorgular sistemde çalıştırılarak her ilk N sonuç için ortalama değerler 20 farklı kişi tarafından doldurulan ilgililik tablosu ile hesaplanmıştır. Ölçüm neticesi Tablo 6.3.'de yer almaktadır.

Çizelge 6.3. TSAP algoritması ile ranking sonucun değerlendirilmesi

TSAP (N)	Domaine Özel Veri Kaynağı Eklenden	Domaine Özel Veri Kaynağı Eklendikten Sonra
N=5	0,257	0,306
N=10	0,203	0,242
N=20	0,149	0,175
N=30	0,103	0,133

TSAP algoritmasına göre, domaine özgü veri kaynaklarının eklenmesi sonuçlarda ilgililik (relevance) açısından bir iyileşme sağlamaktadır.

Genel arama motorları için standart sapma (ilgililik durumu 1,00 ve ilgisizlik durumu 0,00 olarak alınmıştır) ve precision değerleri Çizelge 6.4.’de verilmiştir.

Çizelge 6.4. Deneylere ilişkin standart sapma ve precision değerleri

Veri Kaynağı	Standart Sapma	Precision
SemiRank	0.4068	0,80
Google	0,4493	0,72
Bing	0,4773	0,65
Yandex	0.4818	0,63

N=30 sorgu için, SemiRank yaklaşımının diğer veri kaynaklarına göre bir precision artışı sağladığı gözlemlenmektedir.

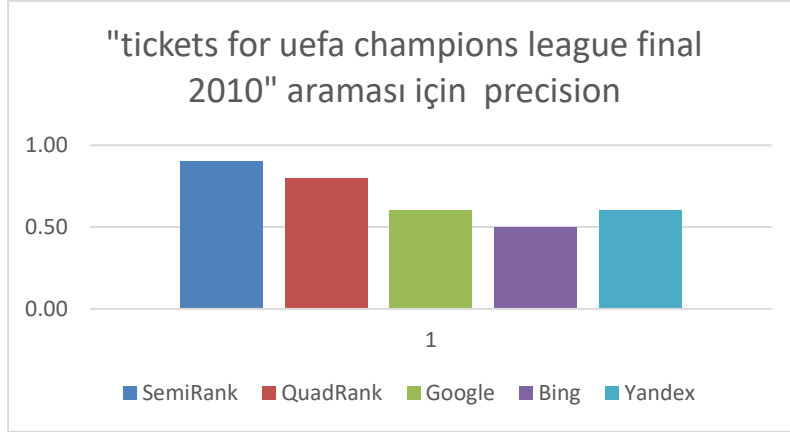
6.4. SemiRank yaklaşımının diğer yöntemlerle kıyaslaması

“tickets for uefa champions league final 2010” araması için Google, Bing, Yandex ve SemiRank tarafından üretilen sonuç kümelerinin ortalama değerlendirilmesini içeren bir TSAP formu Çizelge 6.4’de verilmiştir.

Çizelge 6.5. Örnek bir TSAP değerlendirme formu

Sorgu: tickets for uefa champions league final 2010												
Arama Motoru	1	2	3	4	5	6	7	8	9	10	R	
SemiRank	R	R	R	R	R	R	R	R	-	R		9
QuadRank	R	R	R	R	R	R	-	R	R	-		8
Google	R	-	-	R	R	-	-	R	R	R		6
Bing	R	R	-	R	-	-	R	-	R	-		5
Yandex	R	R	-	R	-	R	-	R	R	-		6

Bu değerlendirme için precision değeri Şekil 6,4’te gösterilmiştir.



Şekil 6.4. Örnek bir sorgu için precision değerleri

Çizelge 6.6. TSAP algoritması ile ranking sonucun değerlendirilmesi

TSAP (N)	SemiRank Domain ve Orijinal Rank verilerinin birleştirilmesi			QuadRank
	(k=0,3)	(k=0,5)	(k=0,7)	
N=5	0,307	0,326	0,321	0,293
N=10	0,234	0,262	0,227	0,221
N=15	0,187	0,189	0,187	0,187

TSAP algoritmasına göre, domaine özgü veri kaynaklarının eklenmesi sonuçlarda ilgililik (relevance) açısından bir iyileşme sağlamaktadır. Domain katsayısını artırmak genel arama motorlarından gelen özgün sonuçların gözden kaçırılmasına neden olduğundan belli bir seviyeden sonra (k=0,7) ilgililik değerini ters yönde etkilemektedir.

6.4.1. Sorguların çalışma zamanlarının kıyaslanması

Çizelge 6.7. İlk 100 sonuç için çalışma süreleri kıyaslanması

Sorgu	QuadRank	SemiRank
Tokyo hotel	0,032	0,032
Free fly player	0,028	0,029
markov chains	0,022	0,026
voice over ip	0,024	0,025
Waterfall	0,023	0,022

SemiRank algoritmasında yer alan her en büyük işlemci yoğun iş olan her bir ASK için TF/IDF (Terim Frekansı / Ters Dokuman Frekansı) hesabı yerine SemiRank algoritması Naive Bayes ile domain benzerliği hesaplamaktadır. Kıyaslanan sonuçlar, SemiRank algoritmasının çalışma zamanına belirgin fazladan bir yük getirmediğini göstermektedir.

Vijaya ve arkadaşları (Vijaya ve diğerleri, 2016) ASK'ların rank değerini hesaplamak için YSA'dan yararlanmaktadır. Adı geçen çalışmada Google, Bing ve Yahoo! Arama motorları kullanılırken bu tez çalışmasında farklı olarak Yahoo! yerine Yandex arama motoru (ülkemizdeki kullanım oranının yüksekliği nedeni ile) tercih edilmiştir. İlgili çalışmanın QuadRank ile çalışma zamanı kıyaslamaları göz önüne alındığında birbirine yakın çalışma zamanları elde edildiği görülmektedir. Bu tez çalışmasında önerilen rank algoritması da QuadRank ile benzer çalışma zamanlarına sahiptir (Çizelge 6.7).

Başarım açısından yapılan kıyaslamalarda QuadRank (Akritidis ve diğerleri, 2011) çalışmasında yer alan 10 sorgudan 8 tanesi Vijaya ve arkadaşları tarafından başarımın ölçülmesi için kullanılmıştır. Bu 8 sonuçta da ortalama precision değerleri açıkça yer almasa da grafiklerden %70-80 bandında dağıldı gözlemlenmektedir. SemiRank algoritmasının ortalama değerleri de 0,766 civarlarında bulunmuştur (Çizelge 6.8.) .

Çizelge 6.8. ilk 10 sonuç için deneylere ait precision ve standart sapma

Veri Kaynağı	Standart Sapma	Precision
SemiRank	0,4301	0,7666
Google	0,4363	0,7458
Bing	0,4766	0,6541
Yandex	0,4752	0,6583

7. SONUÇLAR VE DEĞERLENDİRMELER

Bu bölümde sonuçlar değerlendirilmiş ve gelecekte yapılabilecek çalışmalar sıralanmıştır.

7.1. Sonuçlar

Bu doktora tezinde, yeni bir Otomatik meta arama motoru mimarisi önerilmiştir. Önerilen mimaride sadece Web arama motorları veya Web veri kaynakları el ile eklenmekte, bunun dışında kalan sonuçların yer aldığı bölgelerin tespit edilmesinden, sonuçların birleştirilmesi, sıralanıp gösterilmesine kadar geçen işlemlerin otomatik olarak yapılabileceği gösterilmiştir.

- ASK'nın Sonuç Web sayfalarında otomatik olarak bulunması için içerik yoğunluk faktörü, düzenli ve düzensiz ASK'ların ayrıştırılarak DOM ağacı indirgenmesi ve sonrasında öge frekansına dayalı bir yöntemle arama sonuçlarının otomatik olarak çıkarılabileceği ve birleştirilebileceği gösterilmiştir.
- Zayıf öğrenmeli bir arama sorgusundan domain öğrenme algoritmasının otomatik arama meta motoru bileşeni şeklinde kullanılması sağlanmıştır.
- Otomatik öğrenme sayesinde domaine özgü düşey arama motorlarının sorguya dâhil edilmesi veya sorgudan otomatik olarak çıkartılmasının sonuçların ilgililik değerini artırdığı TSAP ölçümleri ile gösterilmiştir.
- Geliştirilen SemiRank algoritması QuadRank ile kıyaslandığında, çalışma zamanı kabul edilebilir, başarımı nispeten yüksek bir sıralama algoritması olduğu ortaya konulmuştur.

Alanında ilk kez önerilen bu çalışmanın değerlendirilmesi farklı açılardan yapılmış olup, yapılan çalışmaların ülkemizde ve dünyada meta-arama motoru çalışmalarının yanı sıra Web veri kaynaklarının sorgulanması konusunda da daha ileriye taşınmasına katkılar sağlayacağı değerlendirilmektedir.

7.2. Gelecekte Yapılabilecek Çalışmalar

Son yıllarda özellikle JSON kullanan REST API yaklaşımlarının yaygınlaşmış olması nedeniyle, HTML DOM benzerliği yerine JSON benzerliği ile sonuç çıkarımı üstünde çalışma yapılabilecek alanlardan biridir. Ayrıca Sonuç çıkarımı sırasında kelimelerin

semantik benzerliđini de bir  zellik (feature) olarak eřleřtirmeye d hil etmenin sonu lar  st ndeki etkileri incelenebilir.

Arama s re lerinde bireylerin hakları ve meta-arama motorlarının hukuki konumu da farklı bir konu olarak incelenebilir.

KAYNAKLAR

- Adele E., D. H. ve Dreilinger. (1997). SAVVYSEARCH: A metasearch engine that learns which search engines to query. *AI Magazine*, 18(2), 19.
- Akritidis, L., Katsaros, D. ve Bozanis, P. (2011). Effective rank aggregation for metasearching. *Journal of Systems and Software*, 84(1), 130-143.
- Anagnostopoulos I., Anagnostopoluos C., Psoroulas I., Loumos V., K. E. (2003). *An Hybrid Meta Search for Precise Image Search in Web*, 25th Int Conf on Inform Technol Interfaces (ITI), 375-381.
- Baeza-Yates, R., Ribeiro-Neto, B. ve Others. (1999). *Modern information retrieval* (C. 463). ACM press New York, 481.
- Beitzel, S. M., Jensen, E. C., Frieder, O., Grossman, D., Lewis, D. D., Chowdhury, A. ve Kolcz, A. (2005). *Automatic web query classification using labeled and unlabeled training data*. SIGIR 2005 - Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (581–582).
- Benitez, A. B., Beigi, M. ve Chang, S.-F. (1998). A Content-Based Image Meta-Search Engine using Relevance Feedback. *IEEE Internet Computing*, 2(4), 59-69.
- Bergman, M. K. (2001). The deep web: Surfacing hidden value. *Journal of Electronic Publishing*, 7(1), 1–36.
- Berners-Lee, T., Cailliau, R., Groff, J.-F. ve Pollermann, B. (1992). World-wide web: The information universe. *Internet Research*, 2(1), 52–58.
- Bharat, K. (1998). A technique for measuring the relative size and overlap of public Web search engines. *Computer Networks*, 30(1), 379-388.
- Brambilla, M., Ceri, S., Daniel, F., Di Giovanni, M., Mauri, A. ve Ramponi, G. (2018). *Iterative Knowledge Extraction from Social Networks*, Companion Proceedings of the The Web Conference, 1359-1364.
- Branch, A. I. (2012). Search Result Merging and Ranking Strategies in Meta-Search Engines: A Survey. *International Journal of Computer Science Issues*, 9, 239-251.
- Braslavski, P., Alshanski, G. ve Shishkin, A. (2004). *ProThes: Thesaurus-based meta-search engine for a specific application domain*. Proceedings of the 13th International World Wide Web Conference on Alternate Track, Papers and Posters, 222-223..

- Braslavski, P. ve Shishkin, A. (2004). *3 in 1 : Meta-Search , Thesaurus , and GUI for Focused Web Information Retrieval*, Digital Libraries: Advanced Methods And Technologies, Digital Collections. Proceedings of the 6th National Russian Research Conference. 2004, 135-140.
- Brin, S. ve Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine BT - Computer Networks and ISDN Systems. *Computer Networks and ISDN Systems*, 30 (1/7), 107–117.
- Broder, A. (2002). *A taxonomy of web search*. In ACM Sigir forum 2002 , 36(2), 3-10.
- Brown, I. (2013). *Research handbook on governance of the internet. Research Handbook on Governance of the Internet*, Edward Elgar Publishing, 121.
- Chau, M., Chen, H., Qin, J., Zhou, Y., Qin, Y., Sung, W. K. ve McDonald, D. (2002). *Comparison of two approaches to building a vertical search tool: A case study in the nanotechnology domain*. Proceedings of the ACM International Conference on Digital Libraries, 135-144.
- Chen, H., Fan, H., Chau, M. ve Zeng, D. (2003). Testing a Cancer Meta Spider. *International Journal of Human Computer Studies*, 59(5), 755-776.
- Crescenzi, V., Mecca, G. ve Merialdo, P. (2001). *RoadRunner: Towards automatic data extraction from large web sites*. VLDB 2001 - Proceedings of 27th International Conference on Very Large Data Bases, 109-118.
- Diaz, E. D., De, A. ve Raghavan, V. V. (2004). *On Selective Result Merging in a Metasearch Enviroment*. Workshop on Web-based Support Systems, 52–59.
- Ding, C. H. ve Buyya, R. (2004). Guided Google: A meta search engine and its implementation using the Google distributed Web services. *International Journal of Computers and Applications*, 26(3), 1-7.
- Dorn, J. ve Naz, T. (2008). *Structuring Meta-search Research by Design Patterns* , Technology Conference, 6–17.
- dos Santos, C. N. ve Milidiú, R. L. (2012). Named entity recognition. *SpringerBriefs in Computer Science*, 51–58.
- Dragut, E. C., Fang, F., Yu, C. ve Meng, W. (2009). *Deriving customized integrated Web query interfaces*. Proceedings - 2009 IEEE/WIC/ACM International Conference on Web Intelligence, 685-688.
- Dragut, E. C., Kabisch, T., Yu, C. ve Leser, U. (2009). *A hierarchical approach to model web query interfaces for web source integration*. Proceedings of the VLDB Endowment, 2(1), 325-336.

- Dreilinger, D. E. (1996). *Description and evaluation of a meta-searchagent*, Colorado State Üniversitesi Master Tezi, Fort Collins.
- Dwork, C., Kumar, R., Naor, M. ve Sivakumar, D. (2001). *Rank aggregation methods for the Web*. Proceedings of the 10th international conference on World Wide Web - WWW'01 ACM, 613–622.
- Fayzrakhmanov, R. R., Sallinger, E., Spencer, B., Furche, T. ve Gottlob, G. (2018). *Browserless Web Data Extraction: Challenges and Opportunities*. Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18, 1095-1104.
- Gauch, S., Wang, G. ve Gomez, M. (1996). ProFusion: Intelligent fusion from multiple, distributed search engines. *Journal of Universal Computer Science*, 2(9), 637-649.
- Gleim, L. C., Rezaul Karim, M., Zimmermann, L., Kohlbacher, O., Stenzhorn, H., Decker, S. ve Beyan, O. (2018). *Schema extraction for privacy preserving processing of sensitive data*. CEUR Workshop Proceedings, 1(39), 48.
- Gollapudi, S., Jeong, S., Ntoulas, A. ve Paparizos, S. (2011). *Efficient query rewrite for structured web queries*. International Conference on Information and Knowledge Management, Proceedings, 2417-2420.
- Gözüdeli, Y. (2009). Web'den Satmak için SEO: Kocakarı Masalı mı, Sihirli Değnek mi? *Digital Age*, 4(9), 21-24.
- Gozudeli, Y., Karacan, H., Yildiz, O., Baker, M. R., Minnet, A., Kalender, M., Akcayol, M. A. (2015). *A New Method Based on Tree Simplification and Schema Matching for Automatic Web Result Extraction and Matching*. Proceedings of the International MultiConference of Engineers and Computer Scientists, 1, 2–6.
- Gozudeli, Y., Yildiz, O., Karacan, H., Baker, M. R., Minnet, A., Kalender, M., ... Akcayol, M. A. (2014). *Extraction of Automatic Search Result Records Using Content Density Algorithm Based on Node Similarity*. International Conference on Data Mining, Internet Computing, and Big Data, 69–75.
- Gulli, A. ve Signorini, A. (2005). Building an open source meta-search engine. *Special interest tracks and posters of the 14th international conference on World Wide Web - WWW '05*, 1004.
- Guo, J., Xu, G., Cheng, X. ve Li, H. (2009). Named entity recognition in query. *Proceedings of the 32nd international ACM*, 267–274.
- Hamilton, N. (2003). *The mechanics of a deep net metasearch engine*. Proc. of the 12th International World Wide Web Conference, 3–4.
- He, H., Meng, W., Lu, Y., Yu, C. ve Wu, Z. (2007). *Towards deeper understanding of the search interfaces of the deep Web*, 10(2), 133-155.

- He, H., Meng, W., Yu, C. ve Wu, Z. (2003). *WISE-integrator: An automatic integrator of web search interfaces for E-commerce*. Proceedings - 29th International Conference on Very Large Data Bases, VLDB 2003, 357-368.
- He, H., Meng, W., Yu, C. ve Wu, Z. (2004). *Automatic extraction of web search interfaces for interface schema integration*. Thirteenth International World Wide Web Conference Proceedings, 414-415.
- Huang, C. K., Chien, L. F. ve Oyang, Y. J. (2003). Relevant term suggestion in interactive web search based on contextual information in query session logs. *Journal of the American Society for Information Science and Technology (JASIST)*, 54(7), 638-649.
- İnternet: Agarwal, C. A Review and Assessment of Land-Use Change Models Dynamics of Space, Time, and Human Choice. *Cipec*. URL: <http://www.webcitation.org/query?url=http%3A%2F%2Fhero.geog.psu.edu%2Farchives%2FAgarwalEtALInPress.pdf&date=2014-03-17>, Son Erişim Tarihi: 17.03.2014.
- İnternet: Goodwin, D. 60+ Mind-Blowing Search Engine Optimization Stats - SEO 101. URL: <https://www.searchenginejournal.com/seo-101/seo-statistics/#close>, Son Erişim Tarihi: 29.12.2017.
- İnternet: Underwood, L. A Brief History of Search Engines. URL: http://webreference.com/authoring/search_history/index-2.html, Son Erişim Tarihi: 18.08.2004.
- İnternet: Best search engine in the world. URL: <https://www.webdevelopersnotes.com/best-search-engine>, Son Erişim Tarihi: 07.06.2015.
- İnternet: Different Engines, Different Results: Web Searchers Not Always Finding What They're Looking for Online. URL: <http://www.infospaceinc.com/onlineprod/Overlap-DifferentEnginesDifferentResults.pdf>, Son Erişim Tarihi: 03.09.2012.
- İnternet: Dogpile.com. URL: <https://www.dogpile.com/>, Son Erişim Tarihi: 09.08.2016.
- İnternet: Berners, T. Internet Live Stats - Internet Usage & Social Media Statistics. URL: <https://www.internetlivestats.com>, Son Erişim Tarihi: 18.02.2015.
- İnternet: Mahathi S., P. M. ve Singitham. Tadpole: A Meta search engine Evaluation of Meta Search ranking strategies. URL: <http://www.stanford.edu/~pavan/tadpole.html>, Son Erişim Tarihi: 01.04.2013.
- İnternet: Sulvian D., Nielsen NetRatings: Search engine ratings. URL: <https://www.searchenginewatch.com/showPage.html?page=2156451>, Son Erişim Tarihi: 11.28.2019.

- İnternet: Gill, P. The Best Search Engines of 2019. (2019). URL: <https://www.lifewire.com/best-search-engines-2483352>, Son Erişim Tarihi: 06.09.2019.
- Jain, A. ve Pennacchiotti, M. (2010). *Open Entity Extraction from Web Search Query Logs*. Proceedings of the 23rd International Conference on Computational Linguistics, 510–518.
- Jones, K. S. (1971). Automatic keyword classification for information retrieval, Archon Books, 127.
- Kak, S. (1999). Better Web searches and prediction with instantaneously trained neural networks. *IEEE Intelligent Systems and Their Applications*, 6, 78-81.
- Kang, I. H. ve Kim, G. C. (2003). Query type classification for web document retrieval. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, (SPEC. ISS.), 64–71.
- Kumar, N. (2017). Meta Search Engine with Semantic Analysis and Query Processing. *International Journal of Computational Intelligence Research*, 13(8), 2005-2013.
- Laender, A. H. F., Ribeiro-Neto, B. a., da Silva, A. S. ve Teixeira, J. S. (2002). *A brief survey of web data extraction tools*. ACM SIGMOD Record, 31(2), 84.
- Langville, A. N. ve Meyer, C. D. (2011). *Google's PageRank and beyond: The science of search engine rankings*. *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton university press.
- Lawrence, S. (1998). Inquirus, the NECI meta search engine. *Computer Networks*, 30(1-7), 95-105.
- Lewandowski, D., Wahlig, H. ve Meyer-Bautor, G. (2006). The freshness of web search engine databases. *Journal of Information Science*, 32(2), 131-148.
- Li, L., Zhang, W., Shang, Y. ve Liu, Y. (2004). *LinbaCrawler: A linkagebased meta search engine*. Inter Conf on Intelligent Agents, Web Technology and Internet Commerce-IAWTIC'2004, 125-137.
- Liebel, U., Kindler, B. ve Pepperkok, R. (2004). "Harvester": A fast meta search engine of human protein resources. *Bioinformatics*, 20(12), 1962-1963.
- Linner, R. (2008). The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed our Culture [Book Review]. *IEEE Technology and Society Magazine*, 27(2), 61–64.

- Liu, D., Xu, X. ve Long, Y. (2017). *On member search engine selection using artificial neural network in meta search engine*. Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science, 865-868.
- Lu, Y., Meng, W., Shu, L., Yu, C. ve Liu, K. L. (2005). *Evaluation of result merging strategies for metasearch engines*. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 53-66.
- Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), 39-41.
- Mohamed, K. A.-E.-F. (2005). *Merging multiple search results approach for meta-search engines*. ProQuest, Doctoral Tezi, Information Sciences-University of Pittsburg, Pitsbörg.
- Nilsson, K. (2003). A meta search approach to locating and classifying reading material for learners of Nordic languages. *Unpublished thesis, Uppsala University*. http://numerus.lingfil.uu.se/exarb/arch/2003_nilsson.pdf adresinden erişildi.
- Noura, M., Gyrard, A., Heil, S. ve Gaedke, M. (2018). *Concept extraction from the web of things knowledge bases*. Proceedings of the International Conferences on WWW/Internet 2018 and Applied Computing 2018.
- Oztekin, B. U., Karypis, G. ve Kumar, V. (2002). *Expert agreement and content based reranking in a meta search environment using Mearf*, 11th international conference on World Wide Web , 333-344.
- P. V. S. Srinivas and A. Govardhan. (2011). A Survey on the Performance Evaluation of Various Meta Search Engines. *IJCSI International Journal of Computer Science Issues*, 8(3), 359.
- Page, L., Brin, S., Motwani, R. ve Winograd, T. (1998). A Ranking for Every Page on the Web. *World Wide Web Internet And Web Information Systems*, 54(1999–66), 1–17.
- Pandey, K. K. (2018). *Internet Search Engine : Performance Evaluating the Google , Yahoo and Bing Web Search Engine based on their Searching Capabilities*, NCDAMLS, 65-70.
- Paşca, M. (2007). *Weakly-supervised discovery of named entities using web search queries*. International Conference on Information and Knowledge Management, Proceedings 683–690.
- Pinkerton, B. (2000). *WebCrawler: Finding what people want*, CS And Engineering Depth University of Washington, Seattle.

- Rahm, E. ve Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4), 334-350.
- Renda, M. E. ve Straccia, U. (2003). *Web Metasearch: Rank vs. Score Based Rank Aggregation Methods*. Proceedings of the 2003 ACM symposium on Applied computing - SAC '03, 841-846.
- Risvik, K. M., Aasheim, Y. ve Lidal, M. (2003). *Multi-Tier Architecture for Web Search Engines*. LA-WEB, 3, 132.
- Sander-Beuermann, W. ve Schomburg, M. (1998). *Internet Information Retrieval: The Further Development of Meta-searchengine Technology*. Proceedings of the Internet Summit, Internet Society, 22-24.
- Selberg, E. ve Etzioni, O. (1995). *Multi-Service Search and Comparison Using the MetaCrawler*. World Wide Web Conference.
- Selberg, E. ve Etzioni, O. (1997). The MetaCrawler architecture for resource aggregation on the Web. *IEEE Expert-Intelligent Systems and their Applications*, 12(1), 11-14.
- Shen, D., Pan, R., Sun, J.-T., Pan, J. J., Wu, K., Yin, J. ve Yang, Q. (2005). Q 2 C@UST: Our Winning Solution to Query Classification in KDDCUP 2005. *ACM SIGKDD Explorations Newsletter*, 7(2), 100-110.
- Shen, D., Sun, J. T., Yang, Q. ve Chen, Z. (2006). *Building bridges for web query classification*. Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 131-138.
- Steiner, T. ve Mirea, S. (2012). *Seki@home, or crowdsourcing an open knowledge graph*. CEUR Workshop Proceedings, 7-12.
- Tallack, D. (1997). *New York sights: visualizing old and new New York*, Berg Publishers, 42-58.
- Thurrow, S. (2003). *Search engine visibility*. New Riders, 11-17.
- Uichin Lee, Z. L. ve Cho, J. (2005). Automatic Identification of user goals in Web search. *14th international conference on World Wide Web*, (1), 391-400.
- Vijaya, P., Raju, G. ve Ray, S. K. (2016). Artificial neural network-based merging score for Meta search engine. *Journal of Central South University*, 23(10), 2604-2615.
- Wu, W., Yu, C., Doan, A. ve Meng, W. (2004). *An interactive clustering-based approach to integrating source query interfaces on the deep web*. Proceedings of the ACM SIGMOD International Conference on Management of Data, 95-106.

- Wu, Z., Meng, W., Yu, C. ve Li, Z. (2001). *Towards a Highly-Scalable and Effective Metasearch Engine*. Proceedings of the 10th International Conference on World Wide Web, 386-395.
- Yadav, S. ve Prakash Sangwan, O. (2011). Neural Network based Approach for Predicting user Satisfaction with Search Engine. *International Journal of Computer Applications*, 18(5), 16–21.
- Yumusak, S., Dogdu, E., Kodaz, H., Kamilaris, A. ve Vandenbussche, P. Y. (2017). SpEnD: Linked data SPARQL endpoints discovery using search engines. *IEICE Transactions on Information and Systems*, E100D(4), 758–767.
- Zhang, Z., He, B. ve Chang, K. C. C. (2004). *Understanding web query interfaces: Best-effort parsing with hidden syntax*. Proceedings of the ACM SIGMOD International Conference on Management of Data, 107-118.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı : GÖZÜDELİ, Yaşar
 Uyuğu : T.C.
 Doğum tarihi ve yeri : 01.01.1978 / Kahramanmaraş
 Medeni hâli : Evli
 Telefon : 0 (312) 4737639
 E-mail : yagozudeli@gmail.com



Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Doktora	Gazi Üniversitesi / Bilgisayar Mühendisliği	Devam ediyor
Yüksek lisans	Gazi Üniversitesi / Bilgisayar Mühendisliği	2007
Lisans	İstanbul Üniversitesi/ Bilgisayar Bilimleri Mühendisliği	2002
Lise	Atatürk Anadolu Öğretmen Lisesi	1998

İş Deneyimi

Yıl	Yer	Görev
2002-2006	Maliye Bakanlığı	Mühendis

Yabancı Dil

İngilizce, Fransızca

Yayınlar

Gozudeli, Y., Karacan, H., Yildiz, O., Baker, M. R., Minnet, A., Kalender, M., Akcayol, M. A. (2015). *A New Method Based on Tree Simplification and Schema Matching for Automatic Web Result Extraction and Matching*. Proceedings of the International MultiConference of Engineers and Computer Scientists içinde (C. I, 2–6).

- Gozudeli, Y., Yildiz, O., Karacan, H., Baker, M. R., Minnet, A., Kalender, M., Akcayol, M. A. (2014). *Extraction of Automatic Search Result Records Using Content Density Algorithm Based on Node Similarity*. International Conference on Data Mining, Internet Computing, and Big Data içinde (69–75).
- Gözüdeli, Y., Akcayol, M. A. (2010). XML Veritabanı için Tavlama Benzetimi ile Sorgu Optimizasyonu. *Bilişim Teknolojileri Dergisi*, 1(1).
- Gözüdeli, Y. (2009). Yazılımcılar için SQL Server 2008 ve Veritabanı programlama. Seçkin Yayıncılık.
- Gözüdeli, Y., Akcayol, M. A. (2007). Genetik algoritma ile web sayfası düzeninin gerçek zamanlı optimizasyonu. Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 22(2).
- Gözüdeli, Y. (2003). Veritabanı programlama 1, Acar Yayıncılık.

Hobiler

Yüzmek

DİZİN

A

anlamsal sorgu analizi · xii
 arama motorları · xii, 1, 2, 3, 4, 5,
 6, 7, 9, 10, 11, 14, 15, 16, 17,
 18, 19, 20, 21, 23, 24, 25, 26,
 30, 31, 32, 33, 34, 35, 39, 41,
 45, 47, 51, 52, 54, 55, 57, 59,
 65, 67, 69, 70, 75, 76, 79
 Arama Motoru Optimizasyonu ·
 xv, 2
 arama sonuç kayıtları · xii, 33, 47

B

Bilgi çıkarımı · 7

D

Düzenli ASK · 46
 Düzensiz ASK · xviii, 42, 43, 46,
 48, 58

E

Etiket İşleme · 47, 57

G

Google · xviii, 1, 5, 9, 10, 13, 14,
 15, 16, 17, 18, 19, 20, 35, 43,
 45, 46, 47, 73, 74, 75, 76, 81,
 82, 85

H

Helios · xv, xviii, xv, 5, 6, 23, 24,
 54

K

k en yakın komşuluk · xii

M

Makine Öğrenmesi · 35
 Meta arama motorları · 2, 6, 14,
 17, 20, 21, 23, 32, 33, 34, 54

O

Ontoloji · 35

Ö

Örümcek · 1, 17
 Öznitelik Analizi · xv, 27, 30
 Öznitelik Çıkarımı · 26

R

Rank · xvi, 2, 4, 5, 32, 58

S

Sarmaç · 41, 46
 Semantik analiz · 40
 Sorgu İşleyici · 1, 11
 SRR · xv, 33, 66, 67

Ş

Şema Eşleme · xvi, 48, 69

T

TadPole · xv, xviii, xv, 24, 25, 55
 TreeMap · xv, xviii, xv, 24, 25, 26

W

Web bilgi çıkarımı · xii, 5, 7
 Web sonuç sıralama · xii, 54



GAZİ GELECEKTİR..