

WEB TABANLI
KİŞİSEL BİLGİ HASATLAMA

Ömer ERAKMAN

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEMMUZ 2013

ANKARA

WEB TABANLI
KİŞİSEL BİLGİ HASATLAMA

Ömer ERAKMAN

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEMMUZ 2013

ANKARA

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Ömer ERAKMAN

Ömer ERAKMAN tarafından hazırlanan “WEB TABANLI KİŞİSEL BİLGİ HASATLAMA” adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Şeref SAĞIROĞLU

.....

Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. O. Ayhan ERDEM

.....

Bilgisayar Mühendisliği Anabilim Dalı, (Teknoloji Fakültesi) G.Ü.

Prof. Dr. Şeref SAĞIROĞLU

.....

Bilgisayar Mühendisliği Anabilim Dalı, G.Ü.

Yrd. Doç. Dr. Hacer KARACAN

.....

Bilgisayar Mühendisliği Anabilim Dalı, G.Ü.

Tez Savunma Tarihi: 24/07/2013

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Şeref SAĞIROĞLU

.....

Fen Bilimleri Enstitüsü Müdürü

WEB TABANLI KİŞİSEL BİLGİ HASATLAMA**(Yüksek Lisans Tezi)****Ömer ERAKMAN****GAZİ ÜNİVERSİTESİ****FEN BİLİMLERİ ENSTİTÜSÜ****Temmuz 2013****ÖZET**

Bu tez çalışmasında, kişisel bilgi hasatlama işlemleri yapan çalışmalar ile yazılım uygulamaları incelenmiş, web tabanlı kişisel bilgi hasatlama yapan iki uygulama yazılımı geliştirilmiştir. Bu yazılımlara Internet Falcon (IF) NER (isimlendirilmiş varlık tanıma) ve IF Semantik (IFS) adları verilmiştir. IF NER’de doğal dil işleme teknikleri kullanılarak kişisel bilgilerin hasatlanması yapılabilirken, IFS ile bir semantik veritabanı kullanılarak milyonlarca varlık hakkında bilgiler hasatlanabilmektedir. Her uygulamanın kendine göre üstünlükleri ve kısıtları bulunmaktadır. Geliştirilen IF NER, web sayfaları üzerinde çalıştığından hasatlama süresi uzundur ve hasatlama sonucu elde edilen bilgiler geliştirilen IFS’ye göre daha az güvenilirirdir. IFS ise sadece semantik veritabanı üzerinde sorgulamalar yapabildiğinden, bunun dışındaki dokümanlar üzerinde hasatlama işlemini desteklememektedir. IFS’de hasatlama sonucu elde edilen bilgiler güvenilirirdir ve hasatlama süresi IF NER’e göre oldukça kısadır. Sahip olunan veritabanı içeriği kadar hasatlama yapılabilmesi ise geliştirilen bu yazılımın kısıtıdır.

Bilim Kodu : 902.1.014**Anahtar Kelimeler : Web bilgi hasatlama, kişi arama, kişi gruplama, bilgi çıkarma, semantik web****Sayfa Adedi : 121****Tez Yöneticisi : Prof. Dr. Şeref SAĞIROĞLU**

WEB BASED PERSONAL INFORMATION HARVESTING**(M.Sc. Thesis)****Ömer ERAKMAN****GAZİ UNIVERSITY****GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES****July 2013****ABSTRACT**

In this study, studies about personal information harvesting surveyed and two web based software has been developed for harvesting personal information. These software are named that Internet Falcon (IF) NER (named entity recognition) and IF Semantic (IFS). IF NER uses natural language processing techniques to harvest personal information. On the other hand, IFS can harvest about millions of entities on certain database. Each developed software has advantages and constraints. The developed IF NER can work on web pages but harvesting process takes longer time. The information can be obtained in end of harvesting process. IF NER is less reliable than IFS. IFS can query only on semantic database, it cannot harvesting other documents. Information achieved at the end of harvesting is reliable and harvesting process time is shorter than IF NER but IFS can harvest only the content of the semantic database is a constraint.

Science Code	: 902.1.014
Key Words	: Web information harvesting, personal searching, person grouping, information extraction, semantic web
Page Number	: 121
Supervisor	: Prof. Dr. Şeref SAĞIROĞLU

TEŞEKKÜR

Çalışmalarım boyunca yardım ve katkılarıyla beni yönlendiren saygıdeğer danışman Hocam Prof. Dr. Şeref SAĞIROĞLU'na, tecrübelerinden faydalandığım kıymetli hocalarım Yrd. Doç. Dr. Hacer KARACAN'a ve Doç. Dr. Hasan Ş. BİLGE'ye, manevi destekleriyle beni hiçbir zaman yalnız bırakmayan çok değerli kardeşlerim Salih ERAKMAN ve Lütfiye ERAKMAN'a teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR.....	xiii
1. GİRİŞ	1
2. BİLGİ HASATLAMA.....	9
2.1. Hasatlama Mimarileri.....	9
2.2. Kullanılan Yöntemler ve Karşılaşılan Problemler	15
2.2.1. Hasatlama öncesi doğru kaynağın tespiti	15
2.2.2. Bilgi çıkarma yöntemleri	24
2.3. Kullanılan Araçlar	42
2.3.1. GATE.....	42
2.3.2. Arama motorları	43
2.3.3. POS	43
2.3.4. Google Web 1T 5-gram	44
2.4. Genel Değerlendirme	44
3. GELİŞTİRİLEN UYGULAMA: INTERNET FALCON.....	45
3.1. IF NER	46
3.1.1. Hasatlama süreci.....	49

Sayfa

3.1.2. Mimarisi.....	51
3.1.3. Kullanılan teknolojiler	58
3.1.4. Kullanılan yöntemler	61
3.1.5. Yapılan testler	68
3.2. IFS	82
3.2.1. Hasatlama süreci	85
3.2.2. Mimarisi.....	87
3.2.3. Kullanılan teknolojiler	88
3.2.4. Kullanılan yöntemler	90
3.2.5. Yapılan testler	91
3.3. IF NER ve IFS Karşılaştırma	93
3.3.1. IF NER genel değerlendirme	93
3.3.2. IFS genel değerlendirme.....	94
3.3.3. Değerlendirme	95
4. SONUÇLAR VE TARTIŞMA	97
4.1. IF NER	97
4.2. IFS	100
4.3. Gelecekteki Çalışmalar ve Öneriler	102
KAYNAKLAR	104
EKLER.....	111
EK-1. Dilden bağımsız isimlendirilmiş varlık tanıma	112
EK-2. IAF verisi.....	114
EK-3. Wikipedia veri yapısı	119
ÖZGEÇMİŞ	121

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. İlgilenilmeyen sayfaların tespit sonucu	17
Çizelge 2.2. BE ile standart bir arama motoru karşılaştırılması	26
Çizelge 2.3. WebHawk bilgi çıkarma sonuçları.....	28
Çizelge 2.4. Ülke hasatlama kuralı	33
Çizelge 2.5. Hastalık belirtisini tespit etmek için hazırlanan bir CN.....	33
Çizelge 2.6. Sure-fire kuralları örnekleri	34
Çizelge 2.7. Tweet’ler üzerinde POS işaretleme performansı	36
Çizelge 2.8. T-CHUNK’ın OpenNLP’ye göre performansı	36
Çizelge 2.9. T-NER performansı	37
Çizelge 2.10. Tek dilli çıkarma performansı.....	39
Çizelge 2.11. Çapraz dilli çıkarma performansı.....	39
Çizelge 2.12. Çevirilerden çıkarma performansı	40
Çizelge 2.13. TEXTRUNNER ile KNOWITALL karşılaştırılması	41
Çizelge 3.1. ANNIE Jape kural örneği.....	60
Çizelge 3.2. “PersonData” hasatlama XSL	66
Çizelge 3.3. Wikipedia’dan “PersonData” bilgisi hasatlama.....	66
Çizelge 3.4. “VCard” hasatlama XSL.....	67
Çizelge 3.5. Wikipedia’dan “VCard” bilgisi hasatlama.....	67
Çizelge 3.6. Farklı meslek test verisi içeriği	69
Çizelge 3.7. Farklı meslek test verisi: Şeref Sağıroğlu	70
Çizelge 3.8. Farklı meslek test verisi: Antonio Albanese	71
Çizelge 3.9. Farklı meslek test verisi: Erhan Karaesmen.....	72

Çizelge	Sayfa
Çizelge 3.10. Farklı meslek test verisi: Jessica Fox.....	72
Çizelge 3.11. Farklı meslek test verisi: Peter Dourmashkin	73
Çizelge 3.12. IF NER farklı meslek test verisi için gruplandırma başarısı.....	74
Çizelge 3.13. IF NER farklı meslek test verisi ile hasatlama başarısı	74
Çizelge 3.14. Öğretim üyesi test verisi içeriği	75
Çizelge 3.15. Öğretim üyesi test verisi: Şeref Sağıroğlu	76
Çizelge 3.16. Öğretim üyesi test verisi: M. Ali Akcayol	76
Çizelge 3.17. Öğretim üyesi test verisi: Hasan Şakir Bilge	77
Çizelge 3.18. Öğretim üyesi test verisi: Suat Özdemir	77
Çizelge 3.19. Öğretim üyesi test verisi: Hacer Karacan	78
Çizelge 3.20. Öğretim üyesi test verisi: Mehmet Şimşek	78
Çizelge 3.21. Öğretim üyesi test verisi: Murat Hacıömeroğlu.....	79
Çizelge 3.22. Öğretim üyesi test verisi: Oktay Yıldız	79
Çizelge 3.23. IF NER öğretim üyesi test verisi için gruplandırma başarısı.....	80
Çizelge 3.24. IF NER öğretim üyesi test verisi hasatlama başarısı	80
Çizelge 3.25. IF NER Türkçe öğretim üyesi test verisi üzerinde gruplandırma	81
Çizelge 3.26. IF NER IAF test verisi hasatlama performansı.....	82
Çizelge 3.27. Bir varlığa ait tüm niteliklerin elde edileceği MQL sorgusu	90
Çizelge 3.28. IFS’de kullanılan MQL sorgu şablonları	91
Çizelge 3.29. Topic API ile özet bilginin sorgulanması	91
Çizelge 3.30. IFS Antonio Albanese verisi.....	92
Çizelge 3.31. IFS Jessica Fox verisi	92
Çizelge 3.32. IF NER – Semantik karşılaştırma	95
Çizelge 4.1. IF NER farklı meslek ve öğretim üyesi test verileri karşılaştırma.....	99

Çizelge	Sayfa
Çizelge 4.2. IF NER gruplandırma performansı	99
Çizelge 4.3. Bilgi hasatlama çalışmaları karşılaştırma (IAF verisi ile)	100
Çizelge 4.4. IF NER İngilizce ve Türkçe veriye göre performansları	100

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Harvest mimarisi.....	10
Şekil 2.2. IEPAD mimarisi	11
Şekil 2.3. Genel Web hasatlama aşamaları	12
Şekil 2.4. Odaklanmış Web hasatlamada sayfaların tespiti	13
Şekil 2.5. Odaklanmış Web hasatlama.....	13
Şekil 2.6. Ürün Modelleme Ontolojisi	14
Şekil 2.7. Bilgi çıkarım işlemi	31
Şekil 3.1. Geliştirilen IF NER ara yüzü	47
Şekil 3.2. a) IF NER gruplandırma ekranı, b) IF NER hasatlama sonucu ekranı	48
Şekil 3.3. IF NER hasatlama süreci	49
Şekil 3.4. Geliştirilen IF NER akış diyagramı	50
Şekil 3.5. IF NER mimarisi.....	51
Şekil 3.6. IF NER model sınıf diyagramı.....	56
Şekil 3.7. IF NER servis sınıf diyagramı	57
Şekil 3.8. IFS ara yüzü	82
Şekil 3.9. IFS için dil desteği	83
Şekil 3.10. IFS hakkında bilgi hasatlanacak varlığın seçimi	83
Şekil 3.11. IFS varlık hakkında kısa bilgi ve resim bilgisi hasatlama	84
Şekil 3.12. IFS bilgi hasatlama	84
Şekil 3.13. IFS akış diyagramı	86
Şekil 3.14. IFS mimarisi	88
Şekil 3.15. IFS hasatlama süreci ve kullanılan teknolojiler	88

SİMGELER VE KISALTMALAR

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

Simgeler	Açıklama
idf	Ters doküman frekansı
p	Hassaslık katsayısı
r	Geri çağırma katsayısı
tf	Terim frekansı
tf-idf	Terim frekansı - ters doküman frekansı

Kısaltmalar	Açıklama
AJAX	Asenkron JavaScript ve XML
ANNIE	Bilgi Çıkarma Sistemi
API	Uygulama Geliştirme Ara Yüzü
BE	Komşuluk Esasına Göre Çalışan Arama Motoru
CN	Kavram Düğümü
FTP	Dosya Transfer Protokolü
GATE	Metin Mühendisliği İçin Genel Bir Mimari
GWT	Google Web Uygulama Geliştirme Aracı
HAC	Hiyerarşik Aglomeratif Gruplama
HTML	Zengin Metin İşaret Dili
HTML5	5. HTML Sürümü
IAF	İnternet Address Finder Test Verisi
IE	Bilgi Çıkarma
IEPAD	Information Extraction Based on Pattern Discovery
IF	İnternet Doğanı
IFS	İnternet Doğanı Semantik
IR	Bilgi Getirme

Kısaltmalar	Açıklama
JSON	JavaScript Nesne Gösterim Biçimi
KNN	K En Yakın Komşular Algoritması
LCD	Sıvı Kristal Ekran
LIC	Bağlantı Bilgisi Gruplama
LOC	Mekan
LR	GATE’te Dil Kaynakları
SQL	Web Ara Veri Sorgulama Dili
MUC	Mesaj Anlama Konferansı
MVC	Model Görünüm ve Kontrol Tasarım Örüntüsü
NER	İsimlendirilmiş Varlık Tanıma
NFRs	Fonksiyonel Olmayan Gereksinimler
NLP	Doğal Dil İşleme
OIE	Açık Bilgi Çıkarma
ORG	Organizasyon
OWL	Web Ontoloji Dili
PERS	Kişi
PMO	Ürün Modelleme Ontolojisi
POS	Sözdizimsel İşaretleme
PR	Web Sayfasının Önemi Gösteren Katsayı
PRG	GATE’te İşleme Kaynakları
QDEX	Sorgu Tespit ve Çıkarma
RDF	Kaynak Tanımlama Çerçevesi
RDFS	Kaynak Tanımlama Çerçevesi Şeması
SSN	Sosyal Güvenlik Numarası
SVM	Destek Vektör Makinesi
SWOP	Anlamsal Ağ Tabanlı Açık Mühendislik Platformu
SW-PConf	Semantik Web Ürün Konfigürasyonu
UIS	Her Yerde Bilgi Servisi
UMLS	Birleşik Medikal Dil Sistemi

Kısaltmalar	Açıklama
URL	Tekdüzen Kaynak Bulucu
VR	GATE’te Görsel Kaynaklar
VSM	Vektör Uzay Modeli
WAF	Kelime Etkinleştirme Güçleri
WePS2	Webdeki Varlıklar Hakkında Bilgi Toplanması 2. Çalıştayı
WWW	Dünya Çapında Web
XML	Genişletilebilir İşaretleme Dili
XSL	Genişletilebilir Biçimlendirme Dili Ailesi
XSLT	Genişletilebilir Biçimlendirme Dili Dönüşümleri

1. GİRİŞ

Hasatlama, konu özel olarak kaynak sistemler üzerinden bilgileri farklı yöntemleri bir arada kullanarak toplama ve çıkarma işlemidir [1]. Bir varlık hakkında bilgiler hasatlanırken öncelikle doğru varlık hakkında dokümanların bulunduğundan emin olunmalıdır. Bu nedenle bilgi çıkarma ya da hasatlama işlemi genellikle ana iki aşamada yapılmaktadır [1]. Bunlardan ilki varlık ile ilgili doğru dokümanların tespit edilmesi, ikincisi ise bu dokümanlar içerisinde doğru bilgilerin çıkartılması işlemleridir.

Web üzerinde çalışırken karşılaşılan sorunlardan birisi de içeriğin kalitesidir. Web gürültülü, düşük kaliteli ve güvenilmez veriler ile doludur [2]. Bilgi çıkarma çalışmaları genellikle yüksek kalitede veri içeren kaynaklar ile yapılır. Fakat bilgi çıkarma işini Web verisi üzerinden yapmak istediğimizde içerik kalitesi açısından büyük sorunlar ortaya çıkmaktadır.

Web sayfalarının farklı şekillerde oluşturulması da ayrı bir zorluk oluşturmaktadır. HTML (zengin metin işaret dili) sayfaları ne tamamen düz metin ne de tamamen yapılandırılmış bir veri yapısına sahiptir [2]. Yapısı tam olarak belirli olmayan dokümanlar üzerinde işlem yapmak performans kaybına ve işlemin kalitesinin düşmesine neden olur.

Bazı Web geliştiricileri arama motoru sorgusundaki sıralarını değiştirmeye çalışır. Bu şekildeki sayfalara arama motoru istek dışı sayfaları denir [2]. Bu şekilde değiştirilmiş sayfalar asıl bulunmak istenen bilgiler ile alakalı olmadığı halde ilk sıralarda listelenir.

Web hızla gelişirken buna katkı sağlayan kişilerin sayısı da hızla artmaktadır. Bunların bir kısmı profesyonel olmayan ve kendi formatlarında Web sayfaları hazırlayan kişilerdir. Bu sayfaların çoğu (örneğin forumlar, bloglar, vb.) yazım hataları içermektedir. Bu karmaşık ve bozuk yapıya sahip büyük veri topluluğundan kullanıcılara aradıklarını bulma olanağı vermek oldukça zordur. Bu hizmeti sağlayan

sistemlere arama motoru denir. Kullanıcılar genellikle arama motorlarında ilk 10 sayfaya bakmaktadır [3]. Bu nedenle arama motorlarının yüksek hassaslık değerine sahip sonuçlar üretmesi gerekmektedir.

Google çalışmasında yüksek hassasiyetteki sonuçları üretmek için iki önemli özellik vardır [3]. Bunlardan bir tanesi PR'dir (Web sayfasının önemini gösteren katsayı). PR'de akademik alıntı literatürü Web için uygulanmıştır. Web sayfasına verilen alıntılar ve geri bağlantılar hesaplanmıştır. Yüksek sayıda geri bağlantı içeren sayfaların daha önemli ve içeriklerinin daha kaliteli olduğu kabul edilmiştir. Fakat Google sadece bu bağlantıları eşit ağırlıkla saymamış aynı zamanda bunları normalize de etmiştir. PR değeri Eş. 1.1'deki gibi hesaplanmıştır. A sayfasına geri bağlantı veren $T1...Tn$ sayfaları vardır. d , 0 ile 1 arasında değerler alan sönümleme faktörüdür. Genel olarak bu değer 0,85 olarak alınmıştır. $C(T1)$, $T1$ sayfasından çıkan bağlantı sayısını göstermektedir. $PR(T1)$ ise $T1$ sayfasının PR değeridir.

$$PR(A) = (1 - d) + d \left(\frac{PR(T1)}{C(T1)} + \dots + \frac{PR(Tn)}{C(Tn)} \right) \quad (1.1)$$

Google'ın kaliteli sonuç dönmesini sağlayan bir diğer özelliği ise bağlantı metinlerini ele alış biçimidir [3]. Bağlantı metinleri sayfa hakkında bilgi vermekte ve metin olmayan bağlantıları da kullanıcıya sunmaktadır.

Arama motorlarından istediğimizi bulmak bazen küçük adımları arka arkaya gerçekleştirmek ile olur. Örneğin bir profesörün telefon numarası bulunmak istendiğinde, kişiler önce profesörün Web sayfasına gitmekte daha sonra buradan telefon numarasına erişmektedir. Genellikle doğrudan arama motorlarını kullanarak telefon numarasını anahtar kelimeler ile aramamaktadırlar, buna yönlendirme denmektedir [4]. Yaygın arama motorları semantik sorgular yapamamaktadır. Fakat genellikle kullanıcılar semantik özellikleri ya da ilişkileri aramaktadır. Anahtar tabanlı arama tam olarak ihtiyacı karşılamamaktadır.

Araştırmacılar hem doğru varlığın tespiti için hem de tespit edilen dokümanlar içerisinden doğru bilgilerin hasatlanması için birçok yöntem geliştirmişlerdir [1, 5-7].

Bu çalışma kapsamında hasatlama işlemi kaynak sistem olarak çoğunlukla İnternet üzerinde yayınlanan sayfalar incelenmiştir. Hasatlaması yapılacak olan varlık ise kişi olarak belirlenmiştir. Araştırmacılar çalışmalarında genellikle kişi tespit etme ya da bilgi hasatlama şeklinde çalışmalar yapmışlardır [1,5]. Araştırmacılar farklı kişi tespit etme yöntemleri kullanmışlardır ve aynı şekilde bilgi hasatlaması için de farklı yöntemler ortaya koymuşlardır [1, 5-7].

Yapılan araştırmalarda sorgulamalar genellikle tek bir arama motoru üzerinden yapılmıştır [5,8]. Fakat arama motorları sonuçları birbirinden farklılık göstermektedir. Tek bir arama motorundan yapılan aramalarda bazı sonuçlar elde edilemeyebilir. Bu nedenle birden fazla arama motorunu bir arada kullanmak daha fazla dokümana erişimi sağlayacaktır.

Web'deki dokümanlar semantik bir yapıda tutulabilirse, arama motorları da buna uygun olarak çalışabilirler. İnternet üzerindeki dokümanların sayısı ve her geçen gün arttığı düşünüldüğünde semantik yapıya geçmek oldukça zor olacaktır. Bütün ontolojileri kapsayan bir çalışma henüz yapılamamıştır. Bu nedenle bazı sistemler kendi içerisinde ontolojik yapılar oluşturmakta ve bunu kullanmaktadır. Mevcut dokümanların semantik yapıda olmayışı, bu dokümanların semantik ilişkilere çeviren otomatik sistemlere ihtiyaç vardır. Bu sayede varlık arama işlemleri yapılabilir.

Web verisi hızlı bir şekilde sürekli artmaktadır. Bu şekilde artan veriyi işlemek birçok faydalı konuda kullanılmasını sağlamaktadır. Fakat Web verisi genel yöntemlerle kolay bir şekilde işlenemez. Bunun nedenlerinden birisi Web verisi veritabanlarında olduğu gibi belirli yapıda değildir [9].

Web üzerinde milyonlarca kullanıcının kullandığı milyarlarca Web sayfası bulunmaktadır [10]. Bu bilgi topluluğunun her geçen gün biraz daha büyümesi, kullanıcıların bilgiyi organize etmesini ve bilgiye erişmesini güçleştirmektedir. Semantik Web'in amaçlarından bir tanesi de bilgiyi makinelerin okuyabileceği bir formatta saklamak ve sunmaktır. Mevcut bilgi yönetim sisteminin bazı sorunları bulunmaktadır [10]. Bu sorunlar aşağıda başlıklar halinde verilmiş ve kısaca

açıklanmıştır.

Anahtar kelime ile arama: Aramak istediğimiz konu ile ilgisiz sonuçlar çıkabilmektedir. Mevcut Web yapısı anahtar kelime ile aramaya uygundur. Bu durumda istediğimiz sorguları semantik ilişkiler şeklinde elde etmemizi engeller.

Bilgi çıkarma ihtiyacı: Kişilerin Web üzerinden sörf yaparken, bilgi çıkarmaları için metinleri okumaları gerekmektedir. Bunun nedeni arama motorlarının ve Web sağlayıcılarının bilgi çıkarmak için gerekli işlemleri yapmamalarıdır. Bu nedenle mevcut Web yapısı bilgi çıkarma işlemine uygun değildir.

Verinin yönetimi: Zayıf olarak yapılandırılmış bilgi içeriği arttıkça yönetimi de oldukça zorlaşmaktadır. Mevcut Web yapısında verinin yönetimi düz metin olduğundan dolayı zordur.

Bu sorunların üstesinden semantik Web ile gelmek mümkündür. Tüm Web'i semantik bir yapıya taşımak çok zor bir iş olarak görülse de semantik yapıların kullanıldığı birçok çalışma yapılmıştır.

Ding tarafından, çoklu ajan kullanılarak, anlamsal ağlarla mekanik ürün yapılandırması yapmak amacıyla bir çalışma yapılmıştır [11]. SW-PConf (semantik Web ürün konfigürasyonu) isimli uygulamada, gösterim katmanı, ontoloji katmanı, model katmanı ve hizmet katmanından oluşan 4 katmanlı bir mimari kullanılmıştır [11]. Bu mimaride, ihtiyaç bilgisi ve konfigürasyon bilgisini göstermek amacıyla, müşteri ihtiyaç ontolojisi ve ürün konfigürasyon ontolojisi olmak üzere 2 adet ontoloji oluşturulmuştur. İki ontoloji arasında bir eşleme ile, müşteri ihtiyaç bilgisinin ürün konfigürasyon bilgisine otomatik olarak dönüşümü sağlanmıştır. SW-PConf uygulaması, müşterinin isteklerini doğal diller kullanarak bildirmesini ve müşteriye en uygun ürün konfigürasyon sonucunu döndürmeyi hedeflemektedir [11].

SW-PConf uygulaması 4 hiyerarşik katmandan oluşmaktadır [11].

Gösterim Katmanı: Ürün konfigürasyonu alanındaki anlamlandırmaları göstermek amacıyla OWL (Web ontoloji dili) kullanılmıştır. Nesneler, özellikler ve bunlar arasındaki ilişkiler OWL ile tanımlanmıştır.

Ontoloji Katmanı: Ürün yapılandırılmasının amacı, müşteriler ve ürün yapılandırmasını gerçekleştirenler tarafından farklı şekilde tanımlanan ürünler için bireylerin ihtiyaçlarını karşılamaktır. Bu katmanda, müşteri ihtiyaç ontolojisi ve ürün konfigürasyon ontolojisi oluşturularak, bilginin bilgisayarlar tarafından da anlaşılabilir olması ve üst katmanlar için bilgi dönüşümü ile zeki çıkarım yapılması amaçlanmıştır.

Model Katmanı: Anlamsal bilginin soyutlaştırıldığı ontoloji katmanının bir örneği (instance) olarak tanımlanabilen model katmanında, bilgi anlamlarının özel alanlara izdüşümü tanımlanır. Ontoloji katmanındaki müşteri ihtiyaç ontolojisi ile ürün konfigürasyon ontolojisine karşılık gelen alan modelleri, hizmet katmanındaki mantıksal işlemler için bir temel oluşturur.

Hizmet Katmanı: Bu katman, yazım ajanı, görüntüleme ajanı, çeviri ajanı ve konfigürasyon çözüm ajanı olmak üzere 4 farklı ajandan oluşur. Bu ajanlar, kişisel konfigürasyon hizmeti sağlamak amacıyla; ihtiyacın anlaşılması ve anlamlandırılması, müşteri ihtiyaç modeli ile ürün konfigürasyon modelinin oluşturulması, konfigürasyon çözümü ve sonucun müşteriye sunumu için bir birleriyle etkileşimli olarak çalışırlar.

a) *Ontoloji Tanımı:* Ontoloji, varlığın doğasını tanımlamak amacıyla kullanılmaktadır. Ontolojinin özü, ilgi alanındaki sonlu sayıda varlığı ve bunlar arasındaki ilişkileri tanımlamaktır. Aynı ürün için, müşteri ihtiyaçlarını doğal dillerle belirttiği fonksiyonel istekler ile tasarımcıların şartnameye dayalı oluşturdukları ürün konfigürasyonları arasında, iletişim engellerinden kaynaklanan anlamsal uyumsuzlukları ortadan kaldırmak için ontoloji kullanımı ideal bir çözümdür.

b) *Yazım ve görüntüleme ajanları:* Bu ajanlar, uygulamanın kullanıcı ara yüzleridir.

Yazım ajanının temel fonksiyonu, doğal dillerle belirtilen ihtiyaçları çözümlemek ve ontolojiye bağlı olarak anlamlandırarak çeviri ajanına sunmaktır. Doğal dil işleme bu ajanın sorumluluğundadır. Görüntüleme ajanı ise konfigürasyon sonucunu, ürün ağacı ve 3 boyutlu grafikler şeklinde kullanıcıya gösterir.

- c) *Çeviri Ajanı*: İhtiyaç ve konfigürasyon ontolojileri arasında anlamsal bir korelasyon mevcuttur. Bu korelasyon, ontolojiler arasında eşleme (mapping) kuralları oluşturularak gösterilir. Çeviri ajanı, müşteri ihtiyaç bilgisini ürün konfigürasyon bilgisine eşlemek suretiyle ürün konfigürasyon modelini otomatik olarak elde etmeyi amaçlamaktadır. Bu eşleme, müşteri ihtiyaç ontolojisini ürün konfigürasyon ontolojisine kural tabanlı olarak dönüştürülmesi şeklinde gerçekleştirilir.
- d) *Konfigürasyon Çözüm Ajanı*: Çeviri ajanından alınan ürün konfigürasyon modeli ve ürün bileşenleri ile ilgili kısıtları kullanarak, önceden tanımlı bileşen kütüphanesinden müşteri ihtiyaçlarını karşılayacak en uygun bileşenleri seçer [11].

Her yerde bilgisayar (ubiquitous computing), makinelerin insan hayatının her alanında bulunmasına ve insan hayatına girmesine denir [12]. Bu düzeyde makinelerin ve genel olarak işlevsel olmayan ihtiyaçların (NFR'ler) değişmesi çok sıklıkla olmaktadır ve bu makinelerin birbiri ile olan ilişkileri çok önemlidir. Finke ve arkadaşları, bu ihtiyaca bağlı olarak modellemeyi ontolojiler kullanarak yapmayı uygun görmüşlerdir [12].

Bu makalede uluslararası bir havaalanında bilgilendirme servisi sistemi tanımlanmıştır. Bu bilgilendirme sistemine birçok farklı cihaz ile (örneğin büyük LCD (sıvı kristal ekran), tabletler, dizüstü bilgisayarlar, cep telefonları ile) erişilebilmektedir. Her cihaz kendi kapasitesine; boyut, görsel ve ses özelliklerine göre farklılık göstermektedir. Burada amaç, bilgi sisteminden bilgi talep eden cihazın özellikleri ve kapasitesi algılanarak, gönderilecek bilginin içeriğinin ve hacminin buna göre belirlenmesidir.

Farklı özellikteki cihazları modelleyebilmek için, UIS'de (Her yerde bilgi servisi) bir

bölümü verilen ontoloji, OWL ve RDF (kaynak tanımlama çerçevesi) kullanılarak oluşturulmuştur.

Mevcut durumda İnternet üzerinden bir varlık araması yapılmak istenildiğinde kullanıcılar arama motorlarını kullanmaktadır. Arama motorlarının sağladığı ara yüzleri kullanarak giriş yapılır ve aramada girilen verilere göre tamamlanır. Metin tabanlı ara yüzler düşünüldüğünde kullanıcı bir kişi ile ilgili bilgileri arama yapmak istediğinde buraya kişinin adı ve soyadını yazarak arama yapar. Arama motoru sonuç olarak “ad soyad” ikilisi, “ad” ya da “soyad”ı içeren dokümanları sonuç olarak getirir. Ama burada aranan kişinin gerçekten istenilen kişi olup olmadığının doğruluğu hakkında bir işlem gerçekleşmez. Kullanıcı istediği kişiye ait dokümanları gözü ile seçmeli ve bu dokümanlar içerisinde istediği bilgileri yine aynı yöntemle tespit eder. Mevcut İnternet yapısı ve buna bağlı olarak mevcut arama motorları varlık araması için uygun değildir.

Bu tez çalışmasının amacı öncelikle kişisel bilgilerin hasatlanmasıdır. İngilizce dokümanlar incelendiğinden, kişisel bilgiler hasatlanarak akademik personellerin uluslararası tanınırlığının ölçülmesi sağlanabilmektedir. Bir akademik personel hakkında İngilizce kaynaklarda bilgi varsa, bu personel hakkında hasatlama yapmak mümkün olmalıdır. Hakkında daha fazla bilgi olan kişiler hakkında yapılan hasatlama çalışması daha başarılı olmalıdır.

Web üzerinden hasatlama işlemi ile herhangi bir kişi hakkında hasatlama yapmak mümkün olmalıdır. Bu nedenle farklı mesleklerde çalışan kişilerin, kişisel bilgilerinin de hasatlanması amaçlanmıştır.

Bu tez çalışmasında günümüzün popüler konularından semantik hasatlama yöntemleri ile geleneksel hasatlama tekniklerinin karşılaştırılması amaçlanmıştır.

Bu çalışmada, literatürde yapılan çalışmalara ve tekniklere yer verilmiştir. Geliştirilen 2 farklı uygulama hakkındaki detaylı bilgiler ve yapılan deneyler, çalışma kapsamındadır.

2. bölümde, literatürdeki hasatlama mimarilerine yer verilmiştir. Genel bir hasatlama mimarisi ve süreci belirlenmiştir. Hasatlama süreçlerindeki doğru kaynağın tespiti ile ilgili tekniklere ve çalışmalara yer verilmiştir. Öncelikle performansı artırmak için hasatlamada kullanılmayacak sayfaların elenmesi hakkındaki çalışmalardan daha sonra ise sayfaların kişilere göre gruplandırılması için kullanılan algoritma ve tekniklerden bahsedilmiştir. Aynı isme sahip kişilerin olması durumları için hasatlama öncesi sayfaların benzerliklerine bakarak, sayfaların kişilere göre gruplandırması önemlidir. Dokümanların işlenmesi sırasında tüm dokümanın işlenmesi yerine, dokümanların farklı formatlarda gösterilmesine yer verilmiştir. Farklı formatlarda gösterilen ya da başka bir formata dönüştürülmeden hazırlanan dokümanlar üzerinde yapılan bilgi çıkarma tekniklerine bu bölümde yer verilmiştir. Bilgi çıkarma çalışmalarında kullanılan bazı araçlardan bahsedilmiştir.

3. bölümde, bu tez çalışmasında geliştirilen 2 farklı yazılım uygulaması tanıtılmış ve geliştirilen yazılımların testlerinin yapılmasına yönelik çalışmalara yer verilmiştir. Geliştirilen uygulamanın ilki Internet Falcon (IF) NER'dir (isimlendirilmiş varlık tanıma). Bu uygulamada geleneksel teknikler bir arada kullanılarak Web üzerinden kişisel bilgiler hasatlanmaktadır. Diğer uygulama ise IFS'dir (IF Semantik). Bu uygulamada ise semantik bir veritabanı kullanılarak milyonlarca varlık üzerinde hasatlama işlemi yapılabilmektedir. Bu bölümde her iki uygulama ile yapılan deneylerden ve birbiri ile yapılan karşılaştırmalardan bahsedilmiştir.

4. bölümde, geliştirilen iki uygulamanın üstünlükleri ve kısıtları verilmiştir. Yapılan testlerde uygulamalar değerlendirilmiştir. Geliştirilen uygulamalar, literatürdeki diğer çalışmalar ile karşılaştırılmıştır. Gelecekte yapılabilecek çalışmalara ışık tutmak için bazı önerilere yer verilmiştir.

2. BİLGİ HASATLAMA

İnternet üzerinde yayımlanan kişisel bilgiler pek çok güvenlik açığına neden olmaktadır. Günümüzde bazı kişisel bilgiler kullanılarak banka hesapları, e-posta hesapları ya da buna benzer başka kaynaklara erişmek mümkün olabilmektedir. Bu nedenle bilgi hasatlama çalışmaları önem kazanmıştır.

Sosyal güvenlik numarası (SSN), değişik alanlarda kimlik doğrulaması için kullanılmaktadır. Bu nedenle kişiye ait bu özel numaranın gizliliği de çok önemlidir. Acquisti ve Gross halka açık bilgileri kullanarak, SSN'nin tahmin edilebileceğini göstermişlerdir [13]. Hipotezlerine göre Amerika Birleşik Devletleri'nde aynı lokasyonda ve yakın tarihlerde doğan kişilerin, SSN'leri da benzerdir. Tahmin işlemi 2 aşamada yapılmıştır. Öncelikle herkese açık olan ölüm dosyaları işlenmiştir. Bu dosyalarda kişilerin doğum yeri, doğum tarihi ve SSN numaralarını bulmak mümkündür. Önerdikleri hipoteze göre aynı yerde ve yakın tarihte doğan iki kişiden birinin SSN'si biliniyorsa diğerini tahmin etmek mümkün olabilir.

Aşağıda alt başlıklarda bilgi hasatlamada kullanılan teknikler, teknolojiler, yapılar, mimariler ile karşılaşılan problemler açıklanmıştır.

2.1. Hasatlama Mimarileri

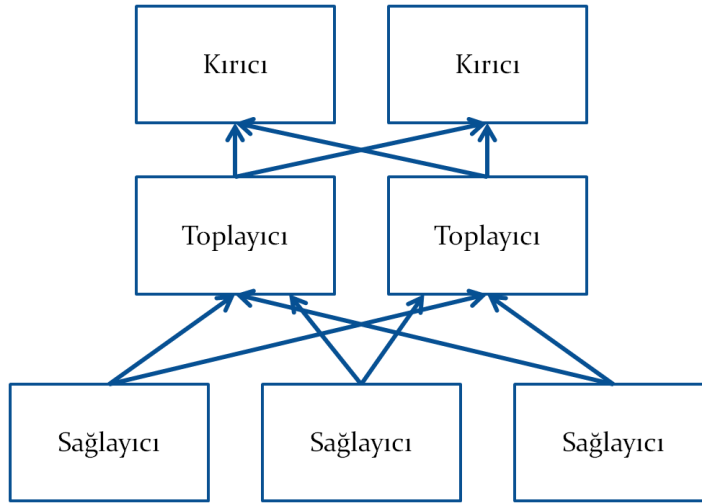
Hasatlama işlemi için uygulanan farklı mimariler vardır. Bu mimariler bazı çalışmalarda farklılık gösterse de temelde benzer katmanlardan oluşmaktadır. Katmanlı yapıya sahip hasatlama çalışmalarında alt katmanın ürettiği çıktıları, üst katmanlar girdi olarak kullanmakta ve sonuçta bilgi elde edilmektedir.

Harvest çalışmasında üç katmanlı bir mimari kullanılmıştır [1].

- a) *Sağlayıcılar (Providers)*, konu ile ilgili verileri sağlarlar. Bunu dizinler ile performanslı olarak sunarlar.
- b) *Toplayıcılar (Gatherers)*, bir üst katmanda bulunmaktadır. Toplayıcılar

dizinlenmiş bir ya da birden fazla sağlayıcıdan verileri alır ve bunlar içerisinde bilgileri toplar.

- c) *Kırıcılar (Brokers)*, üçüncü katmanda bulunmaktadır. Kırıcılar toplanmış bilgiler üzerinde dizinleme ve sorgu yapma özelliklerine sahiptirler. Kırıcılar toplayıcılar ya da diğer kırıcılardan bilgi alabilmektedir. Dizinlerini sürekli güncellemektedirler. Sistemin en önemli kısmı olan kırıcı modülü 4 alt modülden oluşmaktadır; Kollektor, Sicil, Depolama Yöneticisi ve Dizin/ Arama Motoru, Sorgu Yöneticisi.
- *Sicil ve Depolama Yöneticisi*, Kırıcı içerisindeki nesnelerin listesinin yönetimi işini yapmaktadırlar. Bu sayede kırıcıda tekil bilgilerin tutulması sağlanır.
 - *Kollektor*, belirli periyotlar ile toplayıcılardan ya da diğer kırıcılardan güncel verileri istemektedir.
 - *Sorgu yöneticisi modülü*, nesneleri ağlara aktarmaktadır. Bir sorguyu kabul eder ve bunu arama motorunun anlayacağı bir şekle çevirerek ona iletir.

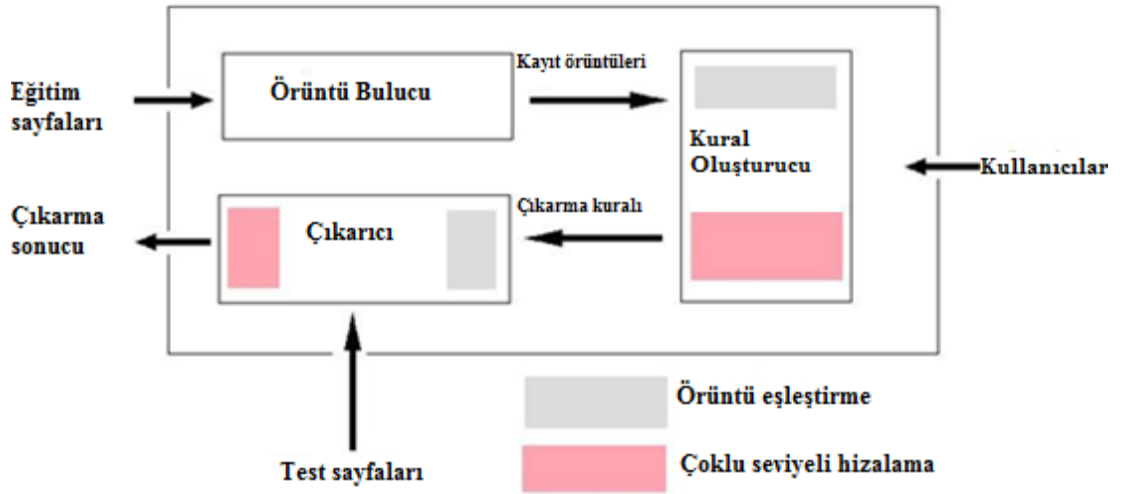


Şekil 2.1. Harvest mimarisi [1]

HTTP, FTP protokolleri ile çalışırken ağ yükü fazla olan işlemler yapılmaktadır. Dizinlere veri sağlayan bu protokoller ağ yoğunluğuna neden olmaktadır. Bu çalışmadaki sorun birkaç farklı şekilde ele alınmıştır. Sağlayıcılar, toplayıcılar için nesnelerin özetlerini sağlayarak, veri miktarını azaltmışlardır. Diğer bir yöntem ise toplayıcıların değişmiş verileri aralıklar ile tarayıp bir ön bellekte saklamasıdır.

Gütl ve diğerleri başka bir hasatlama projesi gerçekleştirmişlerdir [6]. Harvest çalışmasına benzer olarak sağlayıcı, toplayıcı ve kırıcı modülleri vardır [1]. Yazarlar bu projede kullanıcıların özel bir konuda arama yapmalarını ve arama sonuçlarının istenilen kalitede olmasını hedeflemiştir.

IEPAD (Information Extraction based on PAttern Discovery) hasatlayıcı mimarisi 3 ana kısımdan oluşmaktadır. Bunlar “Örüntü Bulucu”, “Kural Oluşturucu” ve “Çıkarıcı”dır [14].



Şekil 2.2. IEPAD mimarisi [14]

Şekil 2.2’de bloklarda verilen hususlar aşağıda açıklanmıştır.

Örüntü Bulucu (Pattern Discoverer): Girdi olarak Web sayfalarını alır ve bilgi çıkarması yapılacak olan veriler için potansiyel örüntüleri tespit eder.

Kural Oluşturucu (Rule Generator): “Pattern Viewer” adında bir ara yüz içerir. Bu ara yüz ile bulunan örüntüler görüntülenir. Kullanıcı buradan örüntüyü seçebilir. Kullanıcı aynı zamanda çıkarılan veriler için nitelik adlarını da atayabilir.

Çıkarıcı (Extractor): Seçilen örüntüye göre benzer sayfalardaki bilgileri çıkarır.

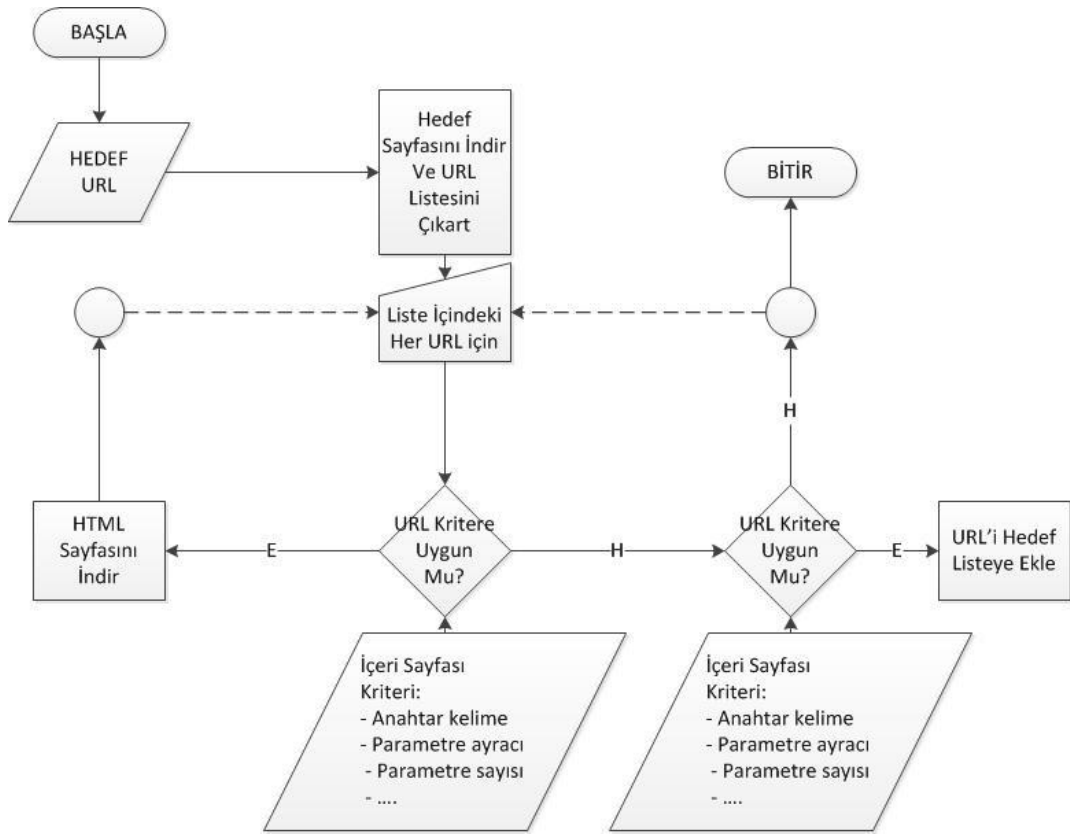
Bazı Web tabanlı hasatlayıcılar öncelikle doğru kaynağın tespit edilmesini

sağlamışlardır. Daha sonra bilgi çıkarıcılar ile bu kaynaklardan hasatlama işlemini gerçekleştirirler. Yapılan hasatlama işlemine göre makine öğrenmesi uygulanabilmekte ve elde edilen bilgiler kontrol edilmektedir. Sonuç olarak bilgi edinilmektedir.

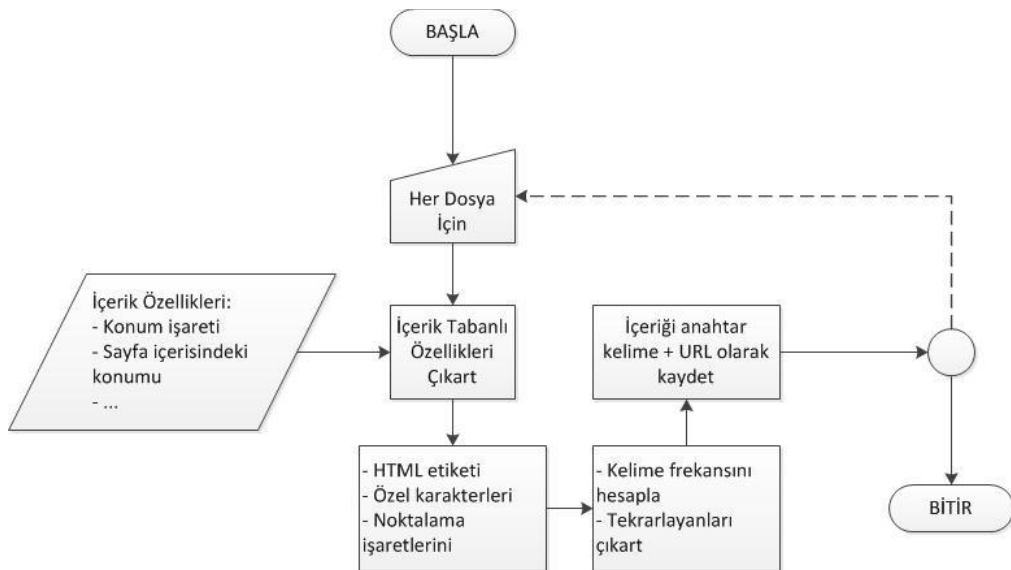


Şekil 2.3. Genel Web hasatlama aşamaları [15]

Odaklanmış Web hasatlama mimarisinde kullanıcı ya da bir sistemin verdiği hedef URL (tekdüzen kaynak bulucu)’den başlayarak sayfalar incelenmiştir [7]. Sayfalar arası bağlantı metinlerinin benzerliklerine bakılarak diğer sayfalar tespit edilmiştir [16-18]. Endonezya’daki bilimsel dizinler kullanılarak hasatlanacak sayfalar tespit edildikten sonra, anahtar kelimeler ve URL’ler olmak üzere bilgiler hasatlanmıştır. Şekil 2.4’te sayfaların tespit edilme akış diyagramı verilmiştir. Şekil 2.5’te ise Web hasatlama ile ilgili akış diyagramı verilmiştir.



Şekil 2.4. Odaklanmış Web hasatlamada sayfaların tespiti [7]



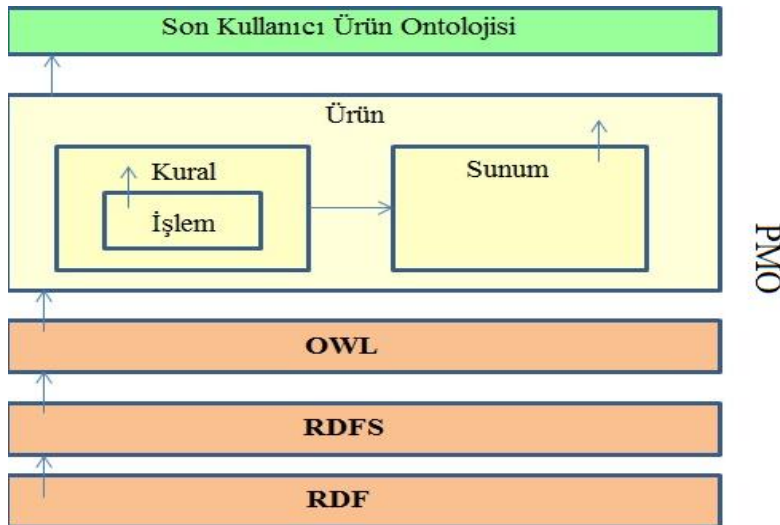
Şekil 2.5. Odaklanmış Web hasatlama [7]

Hakia, semantik arama yapan bir arama motorudur. Hakia semantik mimarisi

temelde 3 modülden oluşmaktadır [19]. Bu çalışmaya çevrimiçi olarak ulaşmak mümkündür [19].

a) *QDEX (Sorgu tespit ve çıkarma)*: Sayfa içeriğini bilgi bitleri olarak analiz etmek ve kaydetmek için kullanılan bir yöntemdir. Web verisi üzerinde semantik olarak hızlı bir şekilde arama yapabilmek için bu yöntem kullanılmaktadır. Bu yöntemde cümleler anlamlı bilgi dizileri halinde saklanmaktadır. QDEX algoritması dokümanın içerdiği bilgiler ile ilgili muhtemel sorguları çıkarmaktadır [19].

Avrupa SWOP (Anlamsal Ağ Tabanlı Açık Mühendislik Platformu) projesi kapsamında Bonsma ve arkadaşları tarafından yapılan başka bir çalışmada, anlamsal ağ yapısı kullanılarak bir ürünün tamamen geliştirilip konfigüre edilebileceği bir modelleme ontolojisi geliştirilmiştir. Çalışma, gerekli ve yeterli olmak koşulu ile tüm son kullanıcı ürün ontolojisini, ilişkili tüm son kullanıcı ürün sınıflarını, özelliklerini ve sayısal kısıtları ile ilişki yapısı, veri tipleri birimler ve ön tanımlı değerler yapısının modellenmesini içermektedir [20].



Şekil 2.6. Ürün Modelleme Ontolojisi [20]

Şekil 2.6'da ürün modelleme ontolojisi (PMO) verilmiştir. PMO; İnternet, WWW (dünya çapında Web), XML (genişletilebilir işaretleme dili), RDF, RDFS (kaynak

tanımlama çerçevesi şeması) ve OWL tanımlamalarından oluşmuş katmanlı protokol yapısı olarak düşünülebilir. Buradaki temel amaç, anlamsal Web tabanlı açık mühendislik projesi alt yapısının oluşturulmasıdır.

PMO kapsamında “operation.owl”, “rule.owl”, representation.owl” ve “product.owl” olmak üzere bir biri ile ilişkili 4 adet ontoloji oluşturulmuştur. PMO’yu bir modelleme paketi olarak kullanabilmek için bu ontolojileri projeye dahil etmek gerekmektedir.

Oluşturulacak tüm ürün ontolojisi, ürün sınıfları, ilişkiler, sınırlamalar, veri tipleri, birimler ve ön tanımlı değerler sonrasında gerçekleşmesi öngörülen süreçte ürünler arasındaki ilişkilerin iyi bir şekilde oluşturulmuş olması ve çıkarımların yani sorgulamaların çok karmaşık ürün modellerinde bile iyi çalışıyor olmasının garanti ediliyor olmasıdır.

Herhangi bir durum için yeni bir ontoloji geliştirmektense, PMO daha esnek, değiştirilip geliştirilebilir ve daha dağınık yapıda bir anlamsal ürün modelinin oluşturulması ve yazılım entegrasyonunun kolayca yapılması sağlanabilir.

Hasatlama çalışmaları mimarilerini incelediğimizde, katmanlı mimariler bulunmaktadır. Her katman yapması gereken işi yapar ve elde ettiği çıktıları diğer katmana iletir. Geliştirilen hasatlama çalışmaları için katmanlı hasatlama mimarilerinin uygun olduğu görülmüştür.

2.2. Kullanılan Yöntemler ve Karşılaşılan Problemler

Bu bölümde literatürdeki bilgi çıkarma yapan çalışmalarda kullanılan yöntemlerden ve karşılaşılan problemlerden bahsedilmiştir.

2.2.1. Hasatlama öncesi doğru kaynağın tespiti

Hasatlama çalışmalarında, bilgiyi çıkarmaya başlamadan önce hasatlama

çalışmalarında kullanılacak asıl verilerin tespiti yapılmalıdır. Örneğin kişisel bilgiler hasatlanacaksa, o kişiye ait olmayan sayfalar elenebilir. Bu bölümde hasatlama işleminden önce doğru kaynakların tespiti ve ilgisiz sayfaların elenmesi ile ilgili çalışmalara yer verilmiştir [5,21,22].

Arama motorlarını kullanarak kişisel bilgileri bulurken en önemli sorunlardan birisi aynı isme sahip başka kişilere ait bilgilerin de karışık olarak gelmesidir. Arama sonuçlarından doğru kaynakların elde edilmesi için farklı yöntemler kullanılmıştır [5,21,22].

Bilgi çıkarma, bilgi getirme ya da sorgu getirme çalışmalarında sıkça kullanılan terimlerden birisi tf-idf'dir (Term Frequency – Inverse Document Frequency). tf-idf, bir terimin kitaplıktaki ters orantısı ile dokümandaki göreceli frekansının belirlenmesidir [23]. Buradan yola çıkarak bu değer o terimin doküman ile ne kadar ilişkili olduğu bilgisini vermektedir. tf-idf hesaplaması farklı uygulamalarda bazı farklılıklar gösterebilmektedir fakat genel yöntem olarak Eş. 2.1 şeklinde hesaplanmaktadır.

$$w_d = f_{w,d} \times \log(|D|/f_{w,D}) \quad (2.1)$$

Burada w_d , w teriminin tf-idf değerini ifade etmektedir. w, d dokümanında geçen bir terim ve $d \in D$ olacak şekilde D kitaplığı ifade eder. $f_{w,d}$, w teriminin d dokümanında geçme sayısıdır. $f_{w,D}$ ise D kitaplığındaki w terimini içeren doküman sayısıdır.

Kaynaklar üzerinde hasatlama işlemini gerçekleştirmeden önce kaynakların belirlenmesi için aşağıdaki yöntemler açıklanmıştır.

İlgisiz sayfaların elenmesi

Sayfaları gruplandırmadan önce kişi ismi olmayan nesneleri içeren sayfaların elenmesi başarıyı artıracaktır. WebHawk uygulaması sayfaları gruplandırmadan önce

bir filtre kullanarak ilgisiz sayfaları elemiştir [5]. İlk 100 sayfanın %35'i bu şekilde elenmiştir. Aşağıdaki özelliklerin frekanslarına göre ağırlıklandırarak filtreleme işlemi yapılmıştır.

1. *Basit sözlüksel özellikler:* Başlıkları, ara kelimeleri ve gövde metinleri içerir. Ağırlıkları olarak kelimelerin frekansları kullanılmıştır.
2. *Biçimsel sözlüksel özellikler:* Koyu yazılan, bağlantılar içinde yazılan sözcükleri içerir. Ağırlıkları olarak kelimelerin frekansları kullanılmıştır.
3. *Sorgu-ilişkili sözlüksel özellikler:* Sorgu kriterinin komşu sözcüklerini içerir. Ağırlıkları olarak kelimelerin frekansları kullanılmıştır.
4. *Sözel özellikler:* Kişi isimleri, kurum ve mekan bilgisini içerir.
5. *Sorgu ilişkili sahiplik özellikleri:* Sahiplik belirten “s” ekinin frekansını içerir.
6. *Sorgu ilişkili kısaltma özellikleri:* A’dan Z’ye kadar harfler için “X” ya da “X.” terimlerinin frekansını içerir.

İlgisiz sayfaların tespitinde Çizelge 2.1’deki sonuçlar elde edilmiştir. w/1 sadece 1. özelliğin kullanıldığını, w/o ise o özellik dışındaki hepsi anlamına gelmektedir. Değerlendirme parametre olarak P , R ve $F_1 = 2pr/(p + r)$ kullanılmıştır. Burada P , uygun bilinen kayıtlara oranı, R ise uygun kayıtların tüm kayıtlara oranıdır.

Çizelge 2.1. İlgilenilmeyen sayfaların tespit sonucu [5]

Özellik Kümesi	P	R	F_1
w/ 1	0,740	0,751	0,745
Tüm Özellikler	0,803	0,774	0,788
w/o 1	0,712	0,738	0,725
w/o 2	0,751	0,802	0,776
w/o 3	0,778	0,773	0,776
w/o 4	0,746	0,775	0,760
w/o 5	0,783	0,777	0,780
w/o 6	0,785	0,775	0,780

Diğer bir çalışmada eleme işlemi daha detaylı yapılmıştır [22]. Sayfanın tamamen elenmesi yerine sayfa içerisindeki bazı metin ve sayılar elenmiştir. 4 kelimedenden daha az olan ve çizgiler ile ayrılmış ya da “” etiketleri ile işaretlenmiş metinler

temizlenmiştir. Bunun dışında noktalama işaretleri 4 haneden daha az olan sayılar da elenmiştir.

Sayfaları gruplandırma

Arama motorlarından kişi ismiyle arama yapıldığında, aynı isme sahip birden fazla kişi olabileceği için isim dışında farklı nitelikleri de kullanarak bir gruplandırma yapılması gerekmektedir. Kullanıcılar genellikle sorgulara yeni kriterler ekleyerek asıl aradığı kişiye ait bilgileri bulmaya çalışmaktadırlar [21]. WebHawk çalışmasında sayfaları gruplandırmak için bir aglomeratif gruplama algoritması kullanılmıştır [5]. Bu algoritmada ilk başta her sayfa farklı bir gruptadır. Daha sonra en benzer 2 grup bir grupta birleştirilmektedir. 2 grubun (c_1, c_2) benzerliği Eş. 2.2 şeklinde hesaplanmaktadır.

$$sim(c_1, c_2) = \frac{\sum_{i=1}^m \sum_{j=1}^n sim(p_i, p_j)}{m \times n} \quad (2.2)$$

Burada p_i ve p_j , c_1 ve c_2 gruplarındaki web sayfalarıdır. m , c_1 'deki sayfa sayısı, n ise c_2 'deki sayfa sayısıdır.

SemEval-2007'de WePS çalışmasında Web sayfalarının kişiye göre gruplandırma yapılmıştır [21]. Kullanılan en önemli değerlendirme parametrelerinden biri, değerlendirilecek dokümanların manuel olarak gruplara ayrılmasıdır. IR (bilgi getirme) çalışmalarında çokça kullanılan performans ve doğruluk ölçülerinden olan P (hassaslık) bu çalışmada da kullanılmıştır. Saflık (purity) değeri de hassaslık değeri ile ilişkili ayrı bir değerlendirme parametresi olarak kullanılmıştır. Burada C ele alınacak gruplardır. L ise manuel olarak oluşturulmuş kategorilerdir, n ise gruplandırılmış eleman sayısını göstermektedir.

$$Saflik = \sum_i \frac{|C_i|}{n} maks Hassaslik(C_i L_j) \quad (2.3)$$

$$Hassaslik(C_i L_j) = \frac{|C_i \cap L_j|}{|C_i|} \quad (2.4)$$

Değerlendirmede kullanılan bir diğer parametre ise ters saflık (inverse purity) değeridir. Bu değer her grupta maksimum R (geri çağırma) değerine odaklanmaktadır. Bu sayede her kategoride daha fazla elemanı olan ödüllendirilmektedir.

$$Ters\ Saflık = \sum_i \frac{|L_i|}{n} maks\ Hassaslık(L_i C_j) \quad (2.5)$$

Kelime gruplarının benzerliği sayfaları gruplandırırken ya da genellikle arama motorları için benzer sorgular üretilirken kullanılmaktadır. Kısa metinlerin benzerliğine bakarken kullanılan yöntemlerden birisi de sözlüksel olarak kelime gruplarının incelenmesidir [24]. Burada Q ve C kelime gruplarının benzerliklerine bakılırken 3 farklı işlem yapılır.

Tamamen eşleşme: Q ve C sözlüksel olarak tamamen birbiri ile aynı ise (Örneğin Q: “satılık kombili ev”, C: “satılık kombili ev”).

İfade: C, Q’nun alt metni ise (Örneğin Q: “satılık kombili ev”, C: “kombili ev”).

Alt küme: C’nin kelimeleri, Q’nun kelimelerinin alt kümesi ise (Örneğin Q: “satılık kombili ev”, C: “satılık ev”).

HAC (Hiyerarşik aglomeratif gruplama)

Liu ve arkadaşları çalışmalarında kişi ismine göre sayfaları gruplandırmak için HAC algoritmasını kullanmışlardır [22,25]. Bu algorithmada önce tüm sayfalar ayrı bir grupta toplanmıştır. Daha sonra her iterasyonda 2 grubun benzerliklerine bakarak tek bir grup oluşturulur. Bu şekilde aşağıdan-yukarıya bir teknik ile gruplamalar yapılmıştır. Çalışmada WePS2 (Web üzerindeki varlıklar hakkında bilgi toplanması 2. çalıştay) veri kümesi kullanılmıştır. Bu kümede 30 kişi ismi ve her kişi için 150 doküman vardır. Toplamda 3444 doküman vardır. Geliştirilen sistemin değerlendirilmesinde B-Cubed, ve Purity-tabanlı puan kullanılmıştır [21,26].

B-Cubed değerlendirmesi için hassaslık ve saflık değerlerini ortalama tutmak hedeflenmiştir [26]. Dokümanda her varlık için hassaslık ve saflık değerleri ayrı ayrı hesaplanmış ve sonuç değerleri üretilmiştir.

$$\text{Sonuç Hassaslık} = \sum_{i=1}^N w_i x \text{Hassaslık}_i \quad (2.6)$$

$$\text{Sonuç Saflık} = \sum_{i=1}^N w_i x \text{Saflık}_i \quad (2.7)$$

Eş. 2.6 ve Eş. 2.7’de, N dokümandaki varlık sayısı, w_i varlığa atanan ağırlık değeridir. Bu çalışmada tüm ağırlıklar eşit olacak şekilde $w_i = 1/N$ olarak hesaplanmıştır.

LIC (Bağlantı bilgisi gruplama) algoritması

Doküman gruplandırma algoritmaları genellikle sadece metin üzerinden işlemler yapabilmektedir. Bu algoritmalar video, müzik ya da resimler için yetersiz kalmaktadır. LIC algoritması KNN (K en yakın komşular algoritması) tabanlı bir algoritmadır, KNN’nin hızını ve doğruluğunu artırmayı hedeflemiştir [27]. Bu algoritma test verisi olarak Web sayfalarını kişilere göre gruplandırmamıştır. Bunun yerine bazı kategorilere göre gruplama yapılmıştır. Aynı algoritma Web sayfalarının kişilere göre gruplandırması amacıyla ya da kişiye göre gruplandırma işleminin bir parçası olarak kullanılabilir.

$$\varphi = \frac{L(d_i)}{O(d_i)} \quad (2.8)$$

Eş 2.8’de, $L(d_i)$ dokümanın uzunluğunu, $O(d_i)$ ise dokümanın dışarı çıkan bağlantı derecesidir. φ ise sınıflandırma derecesi olarak kullanılmıştır.

Eğer bu değer belirli bir değerden yüksek ise, o doküman belirlenen gruba dahil edilir.

$$w(t, d) = \frac{\sum_{i=1}^n q_i}{N(t)} \quad (2.9)$$

Eş. 2.9’da, $N(t)$, t anahtar kelimesini içeren bağlantı sayısını ifade eder. n sınıflandırılacak tüm dokümanlardaki bağlantı sayısını ifade eder. q_i ise ata (üst kategorisindeki) Web sayfalarının sınıflandırılacak web sayfasına olan ağırlığıdır.

$$sim(x, d) = \frac{x.d}{|x|.|d|} \quad (2.10)$$

Daha sonra dokümana benzer K tane bağlantı bilgisi seçilmiştir. Eş. 2.10’da $sim(x, d)$ bağlantı vektör bilgisinin benzerliğidir. d , eğitim kümesini, x ise Web sayfasını göstermektedir.

$$p(x|d) = \sum_{i=1}^{i=k} sim(x, d_i) y(d_i, c) \quad (2.11)$$

Eş. 2.11’de, $y(d_i, c)$, sınıflandırma fonksiyonudur. Eğer d_i , c kategorisinde ise 1, değilse 0 değerini alır. Bu şekilde tüm kategorilerin değerleri hesaplanır. Doküman değeri en yüksek olan kategoride sınıflandırılır.

LIC algoritmasında performansı artırmak amacıyla kurumsal Web siteleri ile İnternet kullanıcılarının Web sitelerini ayrı ayrı gruplandırmışlardır. Kurumsal Web siteleri genellikle belirli bir hiyerarşide olduğundan sayfalar arası bağlantılar da düzenlidir. Diğerlerinde ise kullanıcının isteğine kalmıştır.

Varlık – ilişki diyagramı

Varlık – ilişki diyagramı yönteminde öncelikle k adet Web sayfası alınmıştır [28]. Daha sonra bu sayfalar içerisinden bir IE (bilgi çıkarma) yazılımı kullanılarak kişiler, organizasyonlar, lokasyonlar, e-posta ve bağlantı bilgileri çıkarılmıştır. Bu varlıklar gruplandırılarak çıkarıldıkları Web sayfaları ile ilişkilendirilmiştir. Ayrıca bağlantılar arasında da alt alan bilgisi kullanılarak bir ilişki oluşturulmuştur. Bu sayede varlık-ilişki diyagramı hazırlanmıştır. Algoritma iki nesnenin (u ve v) özelliklerini

kullanarak $f(u, v)$ benzerliklerini hesaplamaktadır. Bu sayede iki nesnenin de aynı referansı işaret edip etmediği kararını vermektedir.

$$c(u, v) = \sum_{p \in P_{uv}} c(p) \quad (2.12)$$

Eş. 2.12’de, $c(u, v)$, iki nesne arasındaki bağlantı gücü hesaplanmaktadır. P_{uv} , iki nesne arasındaki tüm bağlantıları ifade etmektedir. $c(p)$ ise p bağlantısının sağladığı katkıyı ifade etmektedir. Algoritma SIGIR’05 ve WWW’05 verileri ile test edilmiştir. Kişi ismi ile sorgulama yapılmış ve ilk 100 sayfa ile işlenmiştir [29,30].

WAF (kelime etkinleştirme güçleri) tabanlı gruplama

WAF tabanlı gruplandırma çalışmasında öncelikle dokümanlar WAF ile matrisler ile ifade edilmiştir [31,32]. Bu değer 0 ile 1 arasında değer almaktadır. Eğer bu değer 0 ise, j. Kelime hiçbir zaman i. kelimedenden sonra gelmemektedir anlamı çıkmaktadır. Bu şekilde dokümanlar arasındaki benzerlikler hesaplanmıştır. Daha sonraki aşamada ise elde edilen veriler ile dokümanlar için VSM kullanılmıştır. Son olarak da HAC algoritması ile dokümanlar gruplandırılmıştır.

Ansiklopedi ile kişi tespiti

Baidu-PND, Baidu Ansiklopedisi’ni kullanarak kişi ismi tanıma işlemi yapılmıştır [33]. Bu yöntemde 2 aşama bulunmaktadır [33]. İlk aşamada Çince cümleler üzerinde kelime bölümlenmesi yapılır. Bunun için ICTCLAS kullanılmıştır [34]. Daha sonra üç alt aşama uygulanmıştır. İlk alt aşamada kişi özelliklerinin tespiti yapılmıştır. “”, “<< >>”, “[]” gibi özel noktalama işaretleri tespit edilmiştir. “Wei Bo Men” bunlara bir örnektir. İkinci alt aşamada İlişkili-Kümeler tespit edilmiştir. Bunun için Baidu Ansiklopedisi’ndeki ilişkili kişiler ile Web sayfalarındaki ilişkili kişiler kullanılmıştır. Üçüncü alt aşamada veriler üzerindeki gürültünün çıkarılması, noktalama işaretleri elenmesi gibi kaliteyi artırıcı işlemler yapılmıştır.

İkinci aşamada ise kişi ismi tanıma işlemi yapılmıştır. Öncelikle metin benzerliği göz

önünde bulundurulmuştur. Bunun için de kosinüs açısına bakılmıştır.

$$Sim(v_1, v_2) = \frac{\sum_{i=1}^n v_{1i} \cdot v_{2i}}{\sqrt{\sum_{i=1}^n v_{1i}^2} \sqrt{\sum_{i=1}^n v_{2i}^2}} \quad (2.13)$$

Eş. 2.13'te v , metin vektörüdür. v_{1i} ise v_1 vektöründe i . kelimenin tf-idf ağırlığıdır.

$$v = (tf_1 \cdot idf_1, tf_2 \cdot idf_2, \dots, tf_n \cdot idf_n)^T \quad (2.14)$$

D dokümanı, s ismini içermektedir. Bu yöntem ile Baidu'dan D_s dokümanı bulunur. Bu doküman $P(s) = \{p_1, p_2, p_3, \dots, p_n\}$ olası ilişkili isimleri içerir. Sim formülü ile p_i ile D dokümanı arasındaki benzerlik tespit edilir.

Bir sonraki aşamada bağlam benzerlikleri tespit edilir. Bir kişi ismi bir bağlamda farklı birine aitken, diğer bir bağlamda başka birine ait olabilir. Bu da Chr fonksiyonu ile hesaplanır. Sonuç ne kadar büyükse o bağlama daha yakındır.

$$Chr(p, D) = \frac{|C(p) \cap C(D)|}{|C(D)|} \quad (2.15)$$

Eş. 2.15'te $C(D)$, D dokümanındaki kelimelerin özellikleridir. $C(p)$ ise p varlığının özellikleridir.

Eğer birden fazla dokümandaki kişi isimleri aynı kişiye aitse, bu dokümandaki isimler ilişkili isimler arasında olabilir.

$$Grp(p, D) = \frac{|U(p) \cap U(D)|}{|U(D)|} \quad (2.16)$$

Burada $U(p)$, p varlığının ilişkili kişilerini içermektedir. $U(D)$ ise dokümandaki kişi isimleridir. Bu şekilde tutarlılık sağlanmış olur. Baidu-PND'nin performansı SimCon ile karşılaştırılmıştır. SimCon kişilerin bilgi birikimi ve Web sayfalarını ifade etmektedir.

Hasatlama işleminden önce ilgisiz sayfaların elenmesi ve kişilere göre gruplandırma işleminin önemi görülmüştür. Hasatlama işleminde kullanılacak dokümanların sayısı azaltılarak hasatlama performansı artırılabilir.

2.2.2. Bilgi çıkarma yöntemleri

Hasatlama çalışmalarında bilgi çıkarma teknikleri kullanılmaktadır. Örneğin Setooz çalışmasında arama motorlarında da kullanılan hatalı yazım önerilerini bilgi çıkarma yöntemini kullanarak geliştirilmiştir [35]. Öncelikle yazılan kelime 2 karakterden 5 karaktere kadar parçalara ayrılmıştır. Örneğin “October” kelimesi için; oc, ct, to, ob, be, er, oct, cto, tob, obe, ber, octo, ctob, tobe, ober, octob, ctobe ve tober oluşturulmuştur. Bu parçalar daha sonra sorgulamada kullanılmıştır. Bu parçaları içeren öneriler dokümanlar içerisinde çıkartılmış daha sonraki aşamada ise metin benzerliklerine göre öneriler değerlendirilmiştir.

Dokümanlar daha sonra üzerlerinde işlemler yapılmak üzere farklı yapılara ya da formatlara çevrilmektedir. Kullanılan yöntemlerden bazıları aşağıda açıklanmıştır.

i. Vektör uzay modeli

Vektör uzay modelinin uygulandığı bir çalışmada web sayfalarından bilgiler çıkarılmıştır [22]. Bu modele göre her doküman aşağıda maddeler halinde verilen 7 parçadan oluşan bir vektör ile ifade edilmiştir.

- a) *Web sayfası başlığı*: Web sayfasının başlığı tek kelimelere bölünüp özellik uzayına eklenmiştir.
- b) *Web sayfasının URL'i*: Sunucu bilgisini de içeren URL'lerden bir sözlük yardımı ile uzantılar ve anlamsız numaralar kaldırılmıştır.
- c) *Web sayfası meta-veri*: Sadece anahtar kelimeler ve açıklama alanı kullanılmıştır.
- d) *Özet bilgi*: Sorgu eğilimli parçalar Yahoo! Arama motoru tarafından sağlanmaktadır. Bu parçalar genellikle metnin özet bilgisini içermektedir.
- e) *Bağlam penceresi*: Sorguyu içeren pencerenin elemanlarını ifade etmektedir.

- f) *Bağlam cümlesi*: Sorguyu içeren cümleyi ifade etmektedir.
- g) *Kelime torbası*: Dokümanı ifade eden tüm kelimeler özellik kümesine dahil edilmiştir.

Her bir s parçası için tf-idf ağırlığı hesaplanmıştır. Eş. 2.17’de tf_s , s’in frekansını ifade etmektedir. N_d toplam doküman sayısı, df_s ise s’in bulunduğu doküman sayısıdır.

$$W'_s = \log(tf_s + 1) \times \log\left(\frac{N_d}{df_s}\right) \quad (2.17)$$

ii. WAF

Guo ve arkadaşları çalışmalarında kelimeler arasındaki ilişkiyi belirlemeye çalışmışlardır [32]. Dokümandaki i ve j kelimeleri için f_i ve f_j frekanslarını ve f_{ij} birliktelik frekansını hesaplamışlardır. i kelimesinin j kelimesi üzerindeki gücü hesaplamak için $waf_{ij} = (f_{ij}/f_i)(f_{ij}/f_j)/d_{ij}^2$ formülünü kullanmışlardır. Burada d_{ij} , i kelimesinin j kelimesinin birlikteliklerinde önünde olduğundaki durumlarda ortalama uzaklığını ifade eder. Bu şekilde i’den j’ye olan waf_{ij} hesaplanmış olur. Burada waf_{ij} sıfır ise i kelimesi hiçbir zaman L kelimelik kullanıcı deneyiminden daha yakın olarak j kelimesinden daha önce gelmemektedir. waf_{ij} 1 ise o zaman da i kelimesi ile j kelimesi ardışık kelimeler anlamını kazanmaktadır. Bir sözlük için her kelime çifti ile bir asimetrik matris hazırlanmıştır $WAF = \{waf_{ij}\}$. Bu matriste i. satırdaki sıfır olmayan elemanlar i. eleman için i’den çıkan bağlantıları göstermektedir. i. kolondaki sıfır olmayan elemanlar da diğer elemanlardan i’ye olan bağlantıları göstermektedir.

Bilgi çıkarmak için farklı yöntemler kullanılmaktadır. Kullanılan yöntemlerden bazıları aşağıda açıklanmıştır.

1. Özel arama motorları ile hasatlama

Bilgi çıkarma tekniklerinden birisi de özel arama motorlarıdır. Mevcut arama motorları ile anahtar kelime ile arama yapılabilir. Özel arama motorları, bazı sorulara veya özel sorgulara cevap verebilmektedir [36].

KnowItNow çalışmasında kendi arama motorunu BE'yi (komşuluk esasına göre çalışan arama motoru) kullanılarak NLP (doğal dil işleme) teknikleri ile mevcut çalışmalara oranla daha hızlı bilgi çıkarma işlemi yapmaktadır [36]. Mevcut arama motorları NLP işlemlerine uygun dokümanlar dönmemektedir. Bu nedenle Web üzerinden bilgi çıkarma işlemleri çok zaman almaktadır. BE'de bağlamlar hızlı bir şekilde elde edilebilmektedir. Örneğin “cities such as <NounPhrase>” şeklinde bir arama ile şehirler elde edilebilmektedir. BE bu şekilde sorguları desteklemektedir. Her sorgu için $O(k)$ kadar disk okuması gerçekleştirmektedir. Burada k değişken olarak tanımlanmayan terim sayısıdır. BE mimarisinde komşuluk esastır. Komşuluk dizinleri ve yapısı sayesinde sorgu zamanında büyük farklar oluşmaktadır. Standart arama motorlarında sorgu zamanı, BE'nin sorgu zamanından daha uzundur [36].

Çizelge 2.2'de BE'nin sorgu zamanındaki kazancı, k sabit terimler, B ise terimlerin bağlamını ifade etmektedir. K düşük bir sayı olduğundan sorgu zamanı azalmaktadır [36].

Çizelge 2.2. BE ile standart bir arama motoru karşılaştırılması [36]

	Sorgu Zamanı	Dizin Boşluğu
BE	$O(k)$	$O(N)$
Standart Arama Motoru	$O(k + B)$	$O(N)$

2. Düzenli ifadeler ile hasatlama

IEPAD çalışmasında Web sayfaları üzerinden örüntüleri bularak bilgi çıkarmayı hedeflemiştir [14]. Bilgi çıkarma çalışmalarından genel olarak düz metin içerisinde yapılmaya çalışılmıştır. Fakat Web sayfaları düşünüldüğünde tam olarak

düz metin bulunmamaktadır. Yarı yapılandırılmış metinler bulunmaktadır. Örneğin tablolar, numaralandırılmış, ya da listeler gibi veri yapıları bulunmaktadır. Bu nedenle bu yapıyı yarı yapılandırılmış olarak nitelendirmişlerdir. Her Web sayfasının bu yapısı farklılık göstermektedir. Bu nedenle yapıya özel bir teknik geliştirmek zor olacaktır. Burada iki soru ortaya çıkmaktadır. “Tek uzun bir HTML metni olan sayfayı nasıl anlamlı alt metinlere ayrılacaktır?”. İkinci soru ise “bu alt metinlerin nasıl nitelik ve veri elemanlarına ayrılacağıdır?”. Bu nedenle eğitim örneklerine ihtiyaç duymayan örüntü bulan bir algoritma geliştirilmiştir. Bu sayede çıkarıcının oluşturulması ve pratik bir şekilde yapılması sağlanmıştır. IEPAD’ın temel prensibi örüntüleri bulmak ve bu örüntüler ile bilgi çıkarımını yapmaktır. Burada örüntüden kasıt alfabedeki harflerden oluşan simgelerin tanımlandığı düzenli ifadelerin bir alt sınıfıdır.

Deneylerde Kushmerick’in çalışmasından Okra, IAF (Internet Address Finder test verisi), BigBook, ve Quote-Server veri setleri kullanılmıştır [37]. Ortalama sayfa boyutları 28 Kb ve 11 Kb’tır. Birçok Web sitesi (12/14) için IEPAD en iyi kuralı oluşturabilmiş, alma ve doğrulukta %100 başarı sağlamıştır. Sistem sadece 30 eğitim sayfası ile alternatif örüntüler üretebilmiştir. Bilgi getirme oranı tek bir eğitim sayfası ile ilk kural çalıştırılarak %96,28 olmuştur. İkinci kural da çalıştırıldığında bu oran %99,10 olmuştur. Eğitim eğrisi, makine öğrenmesine göre çok daha hızlı olmuştur. Bilgi çıkarma oranı sadece eğitim sayfalarında yüksek olmamıştır, aynı zamanda test sayfalarında da yüksek başarı elde edilmiştir. İki test sayfası için bu oran %98,65 olmuştur [37].

Hakia, genel amaçlı bir semantik arama motorudur [38]. Hakia’nın sunduğu arama sonuçları 3 kriteri yerine getirmektedir. Arama sonuçları, kütüphaneciler tarafından önerilmiş güvenilir Web sitelerinden gelmektedir, en güncel verileri içermektedir ve yapılan sorguya uygun kayıtlardır. Hakia aramaları, Web, haber ve blog aramaları şeklinde yapılabilmektedir. Bunun dışında güvenilir Web sayfaları ve resimler üzerinden de arama yapabilmektedir. Popüler aramalar için “Galleries” denilen farklı başlıklar altında arama sonuçları gruplandırılmaktadır. Örneğin Boston araması için şehir profili, otel kılavuzları gibi bilgiler ayrı gruplar halinde sunulmaktadır.

3. Doğal dil işleme teknikleri ile hasatlama

Web üzerinde dokümanlar genellikle doğal dil ile hazırlanmıştır. Bu nedenle bu dokümanları doğal dil işleme teknikleri ile incelemek mantıklı olacaktır. Fakat forumlar, tartışma grupları düşünüldüğünde buralarda dil ve gramer kurallarına uyulmadan yazılar yazılabilmektedir. İdeal bir ortam olmadığından doğal dil işleme teknikleri ile Web üzerinde hasatlama işlemi yapmak zorlayıcı bir süreçtir.

WebHawk çalışmasında cümle analizi yapılarak kişi ismi, unvanı, e-posta adresi, organizasyon ve telefon bilgileri çıkarılmıştır [5]. Örneğin şu cümle analiz edilerek, unvan bilgisi çıkarılmıştır [5].

“David Lee is a painter.”

Elde edilen sonuçlar çıkarılan bilginin tipine göre performans değişikliği göstermiştir. Hassaslık, geri çağırma ve F1 katsayısı değerleri hesaplanmıştır.

Çizelge 2.3. WebHawk bilgi çıkarma sonuçları [5]

Kategori	P (%)	R (%)	F1
İsim	93,0	92,3	0,926
Unvan	62,5	55,1	0,586
Organizasyon	22,1	33,0	0,265
E-Posta adresi	92,3	88,9	0,906
Telefon numarası	84,3	89,6	0,869

Bilgi Çıkarma (Information Extraction) isimli farklı bir çalışmada bilgi çıkarma teknikleri ve yaşanan zorluklardan bahsedilmektedir [39]. Bilgi çıkarma işlemi genellikle iki ana aşamadan oluşmaktadır. İlk başta sistem, dokümandan yerel metin analizi ile unsurları çıkartır. İkinci aşamada ise çıkarılan unsurları birleştirerek daha büyük ya da yeni unsurlar oluşturulur. Son adımda ise oluşturulan unsurlardan istenilen formatta bir çıktı oluşturulur. İlk unsurları çıkarırken dile ait kalıpları eşleştirerek bulunur. Doğal dili kelime dizileri olarak işlemek pratik olmadığından girdileri düzenleme yoluna gidilmiştir. Bu bileşenlerin ve ilişkilerin düzeyler

şeklinde belirlenmesi ile olmuştur. Bu süreç sözlüksel analiz (kalıpların bulunması, sözlükten bulunan kelimeler) ile başlamakta ve isim tanıma (isimlerin bulunması, tarih gibi özel yapıya sahip bileşenlerin bulunması) işlemi ile devam etmektedir. Bu sözdizimsel analiz ise devam etmektedir. Kalıp eşleştirilmesi kısmı metnin belirlenmiş düzenli ifadeler ile eşleştirilmesi ile olmaktadır. Eğer bir metin bir düzenli ifade ile eşleşirse bu metin o düzenli ifadeye bağlı olarak işaretlenir. Sözlüksel analiz kısmında metin cümlelere ve bileşenlere ayrılır. Her bileşenin sözlükten karşılığına bakılır. İsim tanıma bölümünde düzenli ifadeler kullanılarak tespit edilir. Örneğin kişi isimleri farklı şekillerde olabilir: “Mr. Herrington Smith”, “Fred Smith”, “Snippety Smith Jr.”, “Humble T. Hopp” gibi.

Sözdizimsel analiz bölümünde cümlenin dilbilgisi olarak yapısı incelenir. Bazı sistemler cümlenin tüm gramer yapısını analiz etmektedir fakat bir cümlenin çok farklı yapıları olabildiği için bu şekilde bir işlem çok karmaşık olmaktadır. Şekil 2.8’de verilen işlem adımlarında cümlenin tamamının analiz edilmesi yerine, belirli kalıpların analiz edilmesi yoluna gidilmiştir. Bunlar isim kalıpları (np) ve fiil kalıplarıdır (vp). Bu çalışmada bir cümlenin gramer olarak yapısı şu şekilde işaretlenir [39].

“[np entity: e1 Sam Schwartz] [vg retired] as [np entity: e2 executive vice president] of [np entity: e3 the famous hot dogmanufacturer], [np entity: e4 Hupplewhite Inc.] [np entity: e5 He] [vg will be succeeded] by [np entity: e6 Harry Himmelfarb].”

“Sam was president. He was succeeded by Harry.”

Senaryo kalıp eşleştirmesi bölümü, senaryoya bağlı olarak olayların ve ilişkilerin çıkarıldığı bölümdür. Örneğin: “person retires as position”. Coreference bölümünde zamirler ile belirli isim kalıpları arasındaki isimler analiz edilir. Örneğin “he” şeklinde geçen zamir ile ilgili, “person” varlığı bulunmalıdır. Çıkarım ve olay birleştirme bölümünde bir olaydan sonra başka bir olayın çıkarılması işi yapılır. Örneğin Sam başkandı, Harry onun varisiydi, cümlesinden Harry’nin başkan olacağı çıkarımı yapılır [39].

AeroDAML, web sayfalarından doğal dil işleme bilgi çıkarma tekniklerini kullanarak, DAML formatında bilgileri işaretleyen bir araçtır [40]. Bir dokümandaki tüm bilgileri makinenin anlayacağı bir formata çevirebilmek için, NLP teknikleri kullanmak çok zordur. Bu çalışmada AeroDAML'ın tasarımından bahsedilmiştir.

DAML formatında üretilcek dokümanlar için en az dört farklı araç vardır [40]. Bunlar aşağıdaki şekildedir:

Yarı- otomatik: Yarı otomatik araçlar, kelimeleri DAML sınıfları ve niteliklerine atar. Bu atama işine insanın kendisi karar vermektedir. Genellikle sürükle-bırak şeklinde ara yüzler ile sağlanmaktadır.

Otomatik- varsayılan ontolojiler: Otomatik araçlar kelimeleri sınıf ve niteliklere atarken NLP tekniklerini kullanır. Bu araçlar, önceden belirlenmiş varsayılan bağımsız ontolojileri kullanırlar.

Otomatik- özelleştirilmiş ontolojiler: Bağımsız ontolojiler yerine alana özel ontolojiler kullanırlar. Örneğin; UMLS (birleşik medikal dil sistemi).

Melez: Melez araçlarda ise insan faktörü ve NLP teknikleri beraber kullanılır.

AeroDAML'da melez yapı kullanılmıştır [40]. Teknik olmayan insanların otomatik yapıları, teknik insanların ise yarı otomatik yapıları tercih edeceği düşünülerek melez yapı tercih edilmiştir. Bu çalışmada anlamsal olarak açıklamalı web sayfaları üç farklı amaç için kullanılmıştır:

Bilgi getirme: Benzer sayfa ve dokümanların bulunması ve sıralandırılması.

Soru & Cevap: örneğin; Alaska'nın başkanı kimdir?

Karmaşık soru cevaplama: örneğin; Şu anda Cezayir'deki durum nedir?

Web tabanlı AeroDAML uygulamasında kullanıcı bir URL girer ve uygulama bu

web sayfasını DAML formatında geri döndürür.



Şekil 2.7. Bilgi çıkarım işlemi [39]

İstemci-sunucu yapısında ise kullanıcı bir dosya ismi girer, AeroDAML, DAML formatındaki dokümanı döner. AeroDAML aşağıdaki sınıfları tespit edebilmektedir [40]:

Özel isimler – örneğin: Japan instanceof nation.

Cins isimler – örneğin: gun instanceOf weapon.

Ortak referanslar – örneğin: Clinton eşittir Bill Clinton instanceOf person.

Ölçüm birimleri – örneğin: 22 inches instanceOf measure.

Para – örneğin: \$200 instanceOf money.

Mutlak tarih – örneğin: December 19, 1997 instanceOf absolute date.

Göreceli tarih – örneğin: last month instanceOf relative date.

Geçici ortak referanslar – örneğin: December 19, 1997 is event date then last month is November 1997.

Kuruluştan yere – örneğin: Tyrolean Airways to Austria.

Kişiden kuruluşa – örneğin: Bill Gates to Microsoft.

4. Sınıflandırıcı ile hasatlama

Ufuk çizgisi (skyline) yönteminde birçok parametre ile karar vermede kullanılmaktadır. Web dokümanlarını kişilere göre gruplandırma çalışmalarından biri de Kalashnikov ve arkadaşları tarafından yapılmıştır [41]. Öncelikle Web sayfalarından kişi ve organizasyon bilgileri çıkarılmıştır. Daha sonra ufuk çizgisi sınıflandırıcısı eğitilmiştir.

5. Kural tabanlı hasatlama

KNOWITALL çalışmasında kural tabanlı bilgi çıkarma yapılmıştır [42]. İşaretlenen cümleler üzerinde kurallar çalıştırılır. Eğer kural ile cümle birbiri ile uyuyorsa cümle içerisinden bilgiler hasatlanır. Örneğin Çizelge 2.4'teki kuralda veritabanında saklanan ülkeler “such as” metni ile tespit edilmiştir. Bu kural şu cümleden 3 tane

ülke bilgisini çıkaracaktır [42].

“European countries such as Great Britain, France, and Germany.”

Çizelge 2.4. Ülke hasatlama kuralı [42]

NP1 “such as” NPList2 & head (NP1) = “countries” & properNoun (head (each (NPList2))) => instanceOf (Country, head (each (NPList2))) keywords: “countries such as”

Web üzerindeki veriler yüksek oranda gürültü içermektedir. Bu nedenle KNOWITALL çalışmasında hasatlanan her bilgi için bir olasılık hesaplanmaktadır [42]. Bu olasılık hasatlanan bilginin web üzerindeki frekansına bakılarak yapılmaktadır. Yani bu bilgi için arama motorunun geri döndürdüğü sonuç sayısı kullanılmıştır.

CRYSTAL çalışmasında da bilgi çıkarmak için kurallar tanımlanmıştır. Bu çalışmanın diğerlerine göre farkı yeni kavram düğümleri ya da kuralları öğrenebiliyor olmasıdır [43]. Çizelge 2.5’teki CN (kavram düğümü) şu cümledeki “any episodes of nausea” bilgisini çıkarabilmektedir.

“The patient denies any episodes of nausea”

Çizelge 2.5. Hastalık belirtisini tespit etmek için hazırlanan bir CN [43]

CN type Sign or Symptom Subtype Absent Extract from Direct Object Active voice verb Subject constraints words include “PATIENT” head class <Patient or Disabled Group> Verb constraints words include “DENIES” Direct Object constraints head class <Sign or Symptom>

Kavram düğümleri test edilirken bir hata toleransı kullanılır. Kavram düğümünün hesaplanan hatası bu tolerans ile karşılaştırılır ve ona göre veritabanına kaydedilir. Hata toleransı 0,0 ile 0,4 arasında değer alan bir katsayıdır [43]. Hassaslık ve geri çağırma değerlerine göre, en uygun hata toleransının 0,2 ile 0,3 arasında olduğu gözlemlenmiştir.

6. NER ile hasatlama

İsimlendirilmiş varlık tanıma metin içerisindeki isimlerin ya da kelime gruplarının belirli isimlendirilmiş varlıklar ile eşleştirilmesidir [44].

Grover ve arkadaşları sorgu yapılacak listeler kullanmadan isimlendirilmiş varlık tanıma işlemini yapmışlardır [44]. Çalışmaları 5 adımda olmaktadır. Birinci adımda sure-fire gramer kurallarını işletmişlerdir. Burada aday ifadenin hangi kategoriye girdiği tespit edilmiştir. Sure-fire kurallarının çalışması için önce bir POS (sözdizimsel işaretleme) ve semantik işaretleyicisi çalıştırılmıştır.

Çizelge 2.6. Sure-fire kuralları örnekleri [44]

Kural	Atama	Örnek
Xxxx+ is? A? JJ*PROF	PERS (kişi)	Yuri Gromov, a former director
Xxxx+ is? a? JJ* REL	PERS	John White is beloved brother
Xxxx+ himself	PERS	White himself
Xxxx+, DD+,	PERS	White, 33,
shares in XXXX+	ORG (organizasyon)	shares in Trinity Motors
PROF of/at/with XXXX+	ORG	director of Trinity Motors
XXXX+ area	LOC (mekan)	Beribidjan area

“XXXX+” ilk harfi büyük kelimeyi, “DD” sayıyı, “PROF” uzmanlığı, “REL” akrabalığı, “JJ*” sıfır ya da daha fazla sıfatı, “PERS” kişiyi, “ORG” organizasyonu, “LOC” ise mekânı ifade etmektedir.

İkinci aşamada sistem önce dokümandaki tüm işaretlenmiş varlıkları bir araya getirir. Varlıkların sırasını bozmadan, isimlendirilmiş varlık ile farklı formlarda daha sonra

tekrar karşılaşılr ise aynı atama işlemleri gerçekleştirilir. Örneğin “Adam Kluver Ltd” bir organizasyon olarak işaretlenmiştir, bundan sonra "Kluver Ltd", "Adam Ltd" ve "Adam Kluver" da olası organizasyonlar olarak işaretlenir. Daha sonra bu bilgiler önceden eğitilmiş bir maksimum entropi modeline gönderilir [45]. Onun da kararına göre son işaretleme yapılır.

Üçüncü aşamada kurallar tekrar çalıştırılır ama bu sefer bağlam kısıtları daha az olarak işletilir. Önceki örnekte belirtilen “Adam Kluver” organizasyon olarak işaretlenmişti. Burada ise tekrar kişi olarak işaretlenecektir.

Dördüncü aşamada parçalı eşleştirme yapılmıştır. Örneğin “James White” önceden işaretlenmişse “White” kelimesi de aynı şekilde işaretlenir. Bu aşamada da yeni işaretlenenler maksimum entropi modelinden geçirilir.

Son aşamada ise dokümanın başlığındaki varlıklar doküman içerisindeki varlıklar ile eşleştirilir. Çünkü başlıklarda genelde ilk harf büyük olmakta ve az bir bilgi içermektedir. Yapılan deneyler MUC (mesaj anlama konferansı) - 7 haber dokümanları ile yapılmıştır.

Clark ve arkadaşları çalışmalarında tweetler (<http://twitter.com/> üzerinden gönderilen maksimum 140 karakter olan mesajlar) üzerinden isimlendirilmiş varlık tanıma çalışmasını yapmışlardır [46]. Tweet’lerdeki en büyük sorunlardan biri yüksek oranda gürültü içermesidir. Bu nedenle klasik doğal dil işleme teknikleri yetersiz kalmaktadır. Tweet’lerden isimlendirilmiş varlıkların çıkarılması da oldukça zordur. Öncelikle birçok farklı tipte (organizasyonlar, ürünler, firmalar ve daha birçoğu...) isimlendirilmiş varlık bulunmaktadır. Diğer bir zorluk ise 140 karakterlik sınırdır. Bu sınır içeriğin anlatılmasını kısıtlamaktadır.

Tweetler üzerinde sözlükte olmayan birçok kelime kullanılmaktadır. Bu da POS işaretleyicilerinin performansını düşürmektedir. Bu nedenle geliştirilen T-POS işaretleyicisinde 800 tweet manuel olarak işaretlenmiş ve eğitim kümesi olarak kullanılmıştır. Twitter’a özel yeni etiketler eklenmiştir. Örneğin, “retweets”,

“@usernames”, “#hashtags”, ve “urls”. Çizelge 2.7’ye göre Tweet’ler üzerinde POS işaretleme performansları, parantez içerisinde eğitim kümeleri verilmiştir. Stanford POS Tagger’a göre %41 oranında hata azaltılmıştır [46].

Çizelge 2.7. Tweet’ler üzerinde POS işaretleme performansı [46]

	Doğruluk	Hata Azaltma
Çoğunluk Alt Sınırı (NN)	0,189	-
Kelimenin En Sık Etiketi	0,760	-
Stanford POS Tagger	0,801	-
T-POS(PTB)	0,813	%6
T-POS(Twitter)	0,853	%26
T-POS(IRC + PTB)	0,869	%34
T-POS(IRC + Twitter)	0,870	%35
T-POS(PTB + Twitter)	0,873	%36
T-POS(PTB + IRC + Twitter)	0,883	%41

Sonraki işlemde ise isim, fiil ya da edat öbekleri gibi kelime grupları tespit edilmiştir. Bunun için de T-CHUNK geliştirilmiştir. T-CHUNK, Sha ve Pereira’nın çalışmasındaki “koşullu rastgele alanlar” yöntemini kullanmaktadır [47]. Bu çalışmada “koşullu rastgele alanlar” adında bir çatı geliştirilmiştir [47]. Bu çatı ile veri dizisini bölümlenmek ve işaretlemek mümkündür.

Çizelge 2.8’e göre T-CHUNK, OpenNLP’ye göre doğruluk ve hata azaltma değerlerinde daha iyi sonuçlar elde etmiştir.

Çizelge 2.8. T-CHUNK’ın OpenNLP’ye göre performansı [46]

	Doğruluk	Hata Azaltma
Çoğunluk Alt Sınırı (B-NP)	0,266	-
OpenNLP [48]	0,839	-
T-CHUNK(CoNLL)	0,854	%9
T-CHUNK(Twitter)	0,867	%17
T-CHUNK(CoNLL + Twitter)	0,875	%22

İsimlendirilmiş varlıkların tanınması için Freebase veritabanı kullanılmıştır [49]. Twitter’da bahsi geçen varlıkların %30’u Freebase veritabanında bulunmamaktadır. Bunlar genellikle ya yeni bir kelime olmaktadır ya da yazım yanlış içermektedir.

İşaretlenemeyen varlıklar Labeled LDA ile tespit edilmiştir [50]. Labeled LDA ile tipi belirlenememiş olan varlığın tipi üretilebilmektedir.

Freebase’de kullanılan 10 tip şu şekildedir [49].

“PERSON, GEO-LOCATION, COMPANY, PRODUCT, FACILITY, TV-SHOW, MOVIE, SPORTSTEAM, BAND, OTHER”.

Çizelge 2.9. T-NER performansı [46]

Sistem	P	R	F_1
COTRAIN-NER (10 tip)	0,55	0,33	0,41
T-NER(10 tip)	0,65	0,42	0,51
COTRAIN-NER (PLO)	0,57	0,42	0,49
T-NER(PLO)	0,73	0,49	0,59
Stanford NER (PLO)	0,30	0,27	0,29

Çizelge 2.9’da görüldüğü gibi T-NER, P, R ve F_1 değerlerinde diğer çalışmalardan daha iyi performans sergilemiştir.

Web üzerinde dokümanların çoğu İngilizcedir fakat bazı durumlarda dilden bağımsız olarak isimlendirilmiş varlık tanıma yapılabilir. Dilden bağımsız isimlendirilmiş varlık tanıma çalışmasında veri kümesi olarak CoNLL-2003 veri kümesini kullanmışlardır [51]. Bu veri kümesi İngilizce ve Almanca dillerine ait 8 doküman içermektedir. Her dil için bir eğitim dosyası, bir geliştirme dosyası, bir test dosyası ve büyük bir işaretlenmemiş veri dosyası içermektedir. İngilizce verisi MBT Tagger tarafından işaretlenmiştir [52]. Almanca verisinde ise bir ağaç işaretleyicisi kullanılmıştır. Yapılan çalışmanın performansı ise $F_{\beta=1}$ ile ölçülmüştür. Eş. 2.18’de P , hassaslık, R ise geri çağırma değerleridir. Çalışma performansına ait değerler EK-1’de verilmiştir.

$$F_{\beta} = \frac{(\beta^2 + 1) * P * R}{(\beta^2 * P + R)} \quad (2.18)$$

7. apraz dilli hasatlama

Saggion ve Szasz alıřmalarında İřpanyolca/ İngilizce dillerinde 3 farklı alanda bilgi ıkarma iřlemi yapmıřlardır [53]. Bu alanlar ve ıkarılan bilgiler ařağıdaki gibidir:

Havacılık kazaları: Uak, kaza nedeni, kazanın tarihi, varıř yeri, uuř numarası, kalkıř yeri, vb.

Demiryolu kazaları: Kaza nedeni, kazanın tarihi, varıř yeri, kalkıř yeri, yolcu sayısı, sağı kalan yolcu sayısı, vb.

Depremler: Etkilenen řehir, etkilenen lke, tarihi, merkez ss, kaza sayısı, vb.

Kaynak dilde elde edilen metin hedef dile Google eviri yardımı ile evrilmiřtir [54]. Metinler GATE (metin mhendisliğı iin genel bir mimari) yardımı ile analiz edilmiřtir [55]. Yapılan alıřmada 6 farklı deney yapılmıřtır.

İřpanyolca ile yapılan deneyler: İřpanyolca ıkarma sistemi orijinal İřpanyolca zetler zerinden eğıtilip, İřpanyolca zetler ıkarılmıřtır.

İngilizce ile yapılan deneyler: İngilizce ıkarma sistemi orijinal İngilizce zetler zerinden eğıtilip, İngilizce zetler ıkarılmıřtır.

İřpanyolcada apraz dilli ıkarma deneyleri: İřpanyolca ıkarma sistemi orijinal İřpanyolca zetler zerinden eğıtilip, İngilizceden İřpanyolcaya vrilen zetler zerinde uygulanmıřtır.

İngilizcede apraz dilli ıkarma deneyleri: İngilizce ıkarma sistemi orijinal İngilizce zetler zerinden eğıtilip, İřpanyolcadan İngilizceye vrilen zetler zerinde uygulanmıřtır.

İřpanyolcada eviri deneyleri: İřpanyolca ıkarma sistemi İřpanyolcaya vrilmiř zetler ile eğıtilip, vrilmiř zetler zerine uygulanmıřtır.

İngilizcede çeviri deneyleri: İngilizce çıkarma sistemi İngilizceye çevrilmiş özetler ile eğitilip, çevrilmiş özetler üzerine uygulanmıştır.

Yapılan deneylerde 3 farklı dilde tren, havacılık kazaları ve depremler ile ilgili haberler üzerinden bilgi çıkarma işlemi yapılmış ve elde edilen sonuçlardan P, R ve F ağırlıkları hesaplanmıştır [53].

Çizelge 2.10. Tek dilli çıkarma performansı [53]

Olay	P	R	F
İspanyolca Tren Kazası	0,47	0,41	0,44
İngilizce Tren Kazası	0,65	0,53	0,58
İspanyolca Havacılık Kazası	0,64	0,46	0,54
İngilizce Havacılık Kazası	0,68	0,63	0,66
İspanyolca Deprem	0,61	0,46	0,53
İngilizce Deprem	0,51	0,37	0,43

Tek dilli çıkarmada, tren ve havacılık kazalarında İngilizcede daha iyi performans elde edilmiştir. Depremlerde ise İspanyolcada daha başarılı sonuçlar elde edilmiştir. Bunun nedeni olarak İspanyolcada insan işaretlemesinin daha az olması görülmüştür.

Çizelge 2.11. Çapraz dilli çıkarma performansı [53]

Olay	P	R	F
İspanyolca Tren Kazası	0,87	0,60	0,71
İngilizce Tren Kazası	0,88	0,57	0,70
İspanyolca Havacılık Kazası	0,89	0,59	0,71
İngilizce Havacılık Kazası	0,80	0,56	0,66
İspanyolca Deprem	0,60	0,48	0,53
İngilizce Deprem	0,85	0,60	0,71

Çapraz dilli çıkarma, tek dilli çıkarmaya göre daha iyi bir performans sergilemiştir. Bunun nedeni ise; çevirilerde daha az insan işaretlemesi olmasındandır.

Çizelge 2.12. Çevirilerden çıkarma performansı [53]

Olay	P	R	F
İspanyolca Tren Kazası	0,70	0,57	0,63
İngilizce Tren Kazası	0,76	0,71	0,73
İspanyolca Havacılık Kazası	0,82	0,75	0,78
İngilizce Havacılık Kazası	0,70	0,59	0,64
İspanyolca Deprem	0,60	0,46	0,52
İngilizce Deprem	0,73	0,64	0,68

8. Diğer hasatlama teknikleri

Kör hasatlama tekniği, mümkün olduğu kadar bilgiyi hasatlamaya çalışır [8]. Kişisel bilgilerin doğruluğunu ile ilgilenmez. Bunun için en uygun araç arama motorudur. Hedefe yönelik hasatlama tekniğinde ise kullanıcılar “spam” mesajlara güvenir ve ortalama tekniğinin kullanıldığı sistemlere yönlendirilir [8].

Bilgi çıkarma çalışmada tf-idf ağırlığı yaygın olarak kullanılmaktadır. Bu çalışmalarda genellikle bu ağırlık belirli bir veri seti üzerinden hazırlanmaktadır. Gerçek Web ortamında ise çok daha büyük bir veri mevcuttur. Web’deki verilere daha yakın ağırlıklar hesaplanabilir. Chen ve arkadaşları geliştirdikleri bilgi çıkarma sisteminde “Google’s Web 1T 5-gram” kitaplığını kullanarak tf-idf değerinin daha uygun bir değere çekilmesini sağlayarak tutarsızlıkların üstesinden gelmişlerdir [56].

$$tf(v, d) = \frac{f(v, d)}{\max\{f(v', d), v' \in d\}} \quad (2.19)$$

Burada v , frekansı hesaplanacak isimlendirilmiş varlıktır (terim). d ise frekansı hesaplanacak olan terimin olduğu dokümandır. v' , d dokümanında en fazla geçen terimdir. Bu şekilde bir terimin tf değeri o terimin o dokümandaki frekansının, en fazla geçen terimin frekansına oranıdır.

$$V(w, S) = \log(tf_{w, S} + 1) \times \log(idf_w) \quad (2.20)$$

Eş. 2.20’de S , bir metnin alt dizgileridir. S ’in alt dizgilerinden biri olan w ’nin tf-idf

ağırlığı yukarıdaki Eş. 2.20 ile hesaplanmaktadır. $tf_{w,S}$, w 'nin S içerisindeki frekansdır. idf_w ise w 'yi içeren web dokümanlarının bölümünün tersidir. İki parçalı kelime grupları için Google'ın kitaplığı kullanılarak V' şu şekilde hesaplanmıştır.

$$V'(w, S) = \log(TF_{w,S} + 1) \times (\log(N/TF_{w,gc}) + \log(IDF_w)) \quad (2.21)$$

Eş. 2.21'de $TF_{w,gc}$, w 'nin Google kitaplığındaki frekansdır. N , kitaplıktaki tüm ikili parçaların frekanslarının toplamıdır.

TEXTRUNNER çalışmasında yeni bir bilgi çıkarma tekniği ortaya konulmuştur [57]. Bunun da adına OIE (açık bilgi çıkarma) denmiştir. Bu yöntemde 3 modül bulunmaktadır. Bunlardan ilki Kendinden Denetimli Öğrenicidir. Bu modülde verilen küçük bir kitaplık ile Öğrenici “güvenilir” ve “güvenilir değil” ayrımını yapabilen bir sınıflandırıcı vardır. İkinci modül ise Tek-Geçişli Çıkarıcı'dır. Bu modül tüm kitaplığın üzerinden bir kere geçerek ilişkileri çıkarmaktadır. Çıkarıcı, her cümle için bir ya da birden fazla kelime grubu adayı çıkarır. Bunlar sınıflandırıcıdan geçirilerek güvenilir olup olmadıkları tespit edilir. Son modül ise Artıklık-Tabanlı Değerlendirici'dir. Bu modül, KNOWITALL çalışmasında verilen benzer bir olasılık hesabı yapmaktadır [42]. Fiil, isim gibi grupları normalize ederek grupları birleştirebilmektedir. Örneğin “was developed by” grubu, “was originally developed by” grubunun normalize edilmiş halidir. Bu şekilde farklı cümlelerde normalize halleri aynı olanların sayısı olasılık hesabında kullanılmıştır.

TEXTRUNNER çalışması 9 milyon Web sayfasını içeren bir veri seti kullanılarak, KNOWITALL çalışması ile karşılaştırılmıştır. Sonuçta KNOWITALL'dan daha düşük hata oranına sahip olduğu ve yakın sayıda doğru çıkarma yaptığı tespit edilmiştir.

Çizelge 2.13. TEXTRUNNER ile KNOWITALL karşılaştırılması [57]

	Ortalama Hata Oranı	Doğru Çıkarma Sayısı
TEXTRUNNER	%12	11476
KNOWITALL	%18	11631

2.3. Kullanılan Araçlar

Bilgi çıkarma, metin işleme, doğal dil işleme gibi konularla ilgili birçok araç bulunmaktadır.

2.3.1. GATE

GATE (General Architecture for Text Engineering) ilk olarak 1996 yılında Sheffield Üniversitesi'nde geliştirilmiştir [55]. GATE, dil işleme problemlerine çözüm getirmek için hazırlanmış bir mimaridir. Bu mimaride çokça geçen 3 terim aşağıda kısaca açıklanmıştır.

LR (Dil kaynakları): Bu terim sadece veri içeren kaynaklar için kullanılmaktadır. Örneğin sözlük, kitaplık ya da ontoloji bunlardan birkaçıdır. Bazı dil kaynakları bir yazılım ile beraber gelebilmektedir. Örneğin WordNet bir yazılım ile beraber kullanılsa da sonuçta asıl amaç veriye erişmektir. Bu nedenle dil kaynağı olarak ele alınmıştır.

PRG (İşleme kaynakları): Bu terim ise program niteliğinde ya da algoritmik bir değer taşıyan kaynaklar için kullanılmaktadır. Örneğin üreticiler, çeviriciler ya da ayrıştırıcı bunlardan birkaçıdır.

VR (Görsel kaynaklar): Bir GUI ile bileşenlerin görüntülenmesi ve değiştirilmesini sağlayan kaynaklardır.

GATE'in GATE Developer, GATE Embedded, ANNIE (bilgi çıkarma sistemi) gibi birçok modülü ve eklentisi bulunmaktadır. Çok geniş bir mimari olduğundan, burada sadece tez çalışmasında kullanılan modüllerden bahsedilmektedir.

GATE mimarisinde algoritmalar ve veriler birbirinden ayrılmıştır [58]. Bu da farklı uzmanlık alanına sahip kişilerin rahatça beraber çalışmalarına olanak sağlamaktadır. GATE Developer, geliştirilen kullanıcı ara yüzüdür. Bu ara yüz ile LR, PRG ve VR

üzerinde işlemler yapılabilmektedir.

GATE, bunun dışında başka bir yazılım ile rahatlıkla entegre olabilecek GATE Embedded modülünü de sunmaktadır. Bu API (uygulama geliştirme ara yüzü) ile GATE Developer’ın sunduğu tüm servislere erişmek mümkündür [58].

GATE genel bir mimari olduğundan farklı dil işleme yeteneklerini sunmaktadır. Sağladığı eklentiler ve eklenti geliştirme özelliği ile GATE’e ek özellikler katmak mümkündür. Versiyon 7’de sunulan bazı özellikler şu şekildedir:

- ANNIE eklentisi ile doküman içerisindeki kişi ismi, tarih, yer adı, organizasyon adı gibi varlıklar işaretlenebilmektedir.
- “Verb Group Chunker” eklentisi ile fiil grupları tespit edilebilmektedir.
- “Noun Phrase Chunker” eklentisi ile isim öbekleri tespit edilebilmektedir.

2.3.2. Arama motorları

Arama motorları, veritabanları sistemlerinde Web sayfalarına ait bilgileri saklamaktadırlar. Girilen bir sorguya göre arama algoritmalarını kullanarak sorgu ile ilişkili sonuçları listelemektedirler. Birçok arama motoru dinleyeceği Web sitelerinin kaydedilmesi için servisler sunmaktadır. Bu servislere kaydedilen sayfalar arama motorunca belirli zamanlarda taranarak dizinlenmektedir [59].

Bazı bilgi hasatlama çalışmalarında, üzerinde çalışılacak dokümanların tespitinde ya da bilgi çıkarma işlemi aşamasında arama motoru kütüphaneleri kullanılmıştır. Popüler arama motorlarının çoğunun uygulamalar üzerinden sorgu yapmaya olanak sağlayan kütüphaneleri vardır.

2.3.3. POS

POS, kelimeleri 8 ayrı sınıfta belirler [60]. Bunlar isim, fiil, ortaç, bent, zamir, edat, zarf ve bağlaçtır. Tagger ise otomatik olarak işaretlemek anlamına gelmektedir. POS

işaretleyicisinin bölümleri aşağıdaki gibidir [60].

1. *Parçalama*: Girdi olan metin daha sonraki işlemler için parçalara ayrılır. Bu işlem noktalama işaretleri, kelime-benzeri gruplar gibi sınırlar ile yapılır.
2. *Belirsizlik sorgusu*: Bir sözlük ve tahmin edici kullanır. Sözlükte olmayan parçalar tahmin edici tarafından tespit edilir ve olası işaretlemeyi yapar.
3. *Belirsizlik çözümü ve belirsizliği giderme*: Bu işaretlemede son aşamadır. Belirsizlik iki kısımda olabilir:
 - a) Kelimeye ait bilginin kendisinde, örneğin fiil yerine isim olarak işaretlenmiş olabilir.
 - b) Kelime/ işaret dizisinde ait bilgide, örneğin eğer önceki kelime bağlaç ya da bent ise, model isim ya da fiil analizi yapılabilir.

2.3.4. Google Web 1T 5-gram

Google Web 1T 5-gram, Google tarafından n-parçalı İngilizce kelime gruplarını ve bu kelime gruplarının 2006 yılında web sayfalarından alınmış 1 trilyon kelime grubundaki frekanslarını içermektedir [61]. Bu veri setinin varlık tanıma, bilgi çıkarma, ses tanıma gibi birçok çalışmada kullanılacağı öngörülmüştür.

2.4. Genel Değerlendirme

Bilgi çıkarmak için önceki başlıklarda bahsedilen yöntemler bulunmaktadır. Web verisi üzerinde çalışırken aynı isme sahip kişilere ait sayfaların birbirinden ayrılması için sayfaların gruplandırılması önemlidir. Web üzerindeki herhangi bir doküman üzerinde çıkarma işlemi yapmak için cümle analizi genel bir çözüm üretebilmektedir. Cümle analiz edilirken POS araçları cümleyi anlamlandırmak için yardımcı olacaktır. Web'deki dokümanlara erişmek ve hızlı arama sonuçları elde edebilmek için arama motorları kullanılabilir.

3. GELİŞTİRİLEN UYGULAMA: INTERNET FALCON

Bu tez çalışmasında 2. Bölüm’de bahsedilen sorunlara çözüm getirmek amacıyla 2 farklı uygulama geliştirilmiştir. Bu uygulamaların adları IF NER ve IFS’dir.

Günümüzde kişisel bilgi güvenliğinin önemi gittikçe artmaktadır. Doğum tarihi, kimlik numarası, meslek gibi bilgiler, bankacılıkta, telefon ve İnternet üzerinden yapılan diğer işlemlerde kullanılmaktadır. Web üzerinden herkesin kolayca erişebileceği bu bilgiler o kişiler açısından güvenlik açığına neden olmaktadır. Bu yüzden bu bilgilerin otomatik olarak tespit edilmesi önem kazanmaktadır.

Web üzerinde popüler arama motorlarını kullanarak semantik aramalar yapmak mümkün değildir. Web sürekli geliştiğinden semantik yapılara geçebilmek için otomatik olarak bilgilerin çıkarılması ve bir ontolojiye aktarılması gerekmektedir.

Mevcut arama motorları ile bu bilgileri bulmak zor bir iştir. Arama sonuçları ve sayfalar tek tek incelenerek bu bilgiler çıkarabilir. IF NER uygulaması bu işlemi kullanıcıdan bir kere geri bildirim alarak yapmaktadır. Kullanıcı hangi kişiden bahsettiğini seçerek o kişiye ait kişisel bilgilerin tespit edilmesine yardımcı olur. Hiç geri bildirim almadan tüm bilgiler de çıkarılabilir fakat performans açısından resim ve özet bilgi gruplar ile beraber gösterilir. Bu şekilde kullanıcı gerçekte hangi kişiyle ilgili hasatlama yapılması istediğini seçebilir.

Popüler arama motorlarında kişisel bilgilerin arandığı durumda Web üzerinde en çok bilgisi bulunanlar daha çok sayıda ve daha üst konumlarda çıkmaktadır. Hangi sayfanın hangi kişiden bahsettiğini anlamak kullanıcıya kalmaktadır. Bu nedenle Web sayfalarının kişilere göre gruplandırılması konusunda birçok çalışma yapılmıştır. IF NER’de kişisel bilgileri hasatlamadan önce sayfaları gruplandırmaktadır. Asıl amacı hasatlama işlemi olduğundan az bilgi içeren ya da hiç bilgi içermeyen sayfalar gruplandırma işlemi sırasında elenmektedir. Bir Web sayfasının en benzer olduğu grup ile olan benzerliği, minimum benzerlik değerinden daha az ise bu sayfa elenmektedir. Bu sayede hasatlanan bilgilerin kalitesi veya

doğruluğu artmaktadır.

Bu tez çalışması kapsamında geliştirilen ve IF olarak bölümlendirilen çalışmalar, 2 farklı uygulamadan oluşmuştur. İki uygulamada da kullanılan teknolojiler, yöntemler ve ara yüzler farklılık göstermektedir. Her iki uygulamanın da avantajları ve dezavantajları bulunmaktadır. Bu tez kapsamında geliştirilen uygulamalar aşağıda alt başlıklarda detaylı açıklanmıştır.

3.1. IF NER

IF NER, Web tabanlı bir uygulamadır. Sunulan ara yüz ile kişilere ait sayfalar aranabilmektedir. Temel hedef olarak kişilere ait sayfalar aranabilir fakat bunun dışında herhangi bir konuda da arama yapmak mümkündür. IF NER’de sunulan ara yüz ile kişi ismini parametre olarak alır daha sonra arama motoru entegrasyonunu kullanarak arama motoru arama sonucunu alır. Elde edilen Web sayfaları aynı isme sahip kişilere göre gruplandırılır. Her adaş için farklı bir grup oluşturulur. Kullanıcı bu gruplardan birisini seçerek hasatlama işlemini başlatabilir.

Web verisi çok gürültülüdür ve belirli düzgün bir yapıya sahip değildir [2]. Bu nedenle sayfalar arası gruplandırmayı yaparken belirli bir benzerlik katsayısı kullanmak çok zordur. Bu nedenle IF belirli bir gruplandırma katsayısı (genellikle 1,0) ile gruplandırmaya başlar, daha sonra belirlenen en az gruplandırma katsayısına kadar (genellikle 0,1) belirli bir oranda gruplandırma katsayısı azaltılır. Bu şekilde benzerlik hesaplamalarına göre web sayfaları çok benzer olmasa bile gruplandırma yapılabilmektedir. Web verisindeki tutarsızlık ve bozukluklara rağmen uygun bir şekilde gruplandırma yapılabilir.

IF NER, sadece İngilizce dokümanlar üzerinde işlem yapmaktadır. Diğer dillerde performans düşebilir. Bu nedenle arama motoru sorgularında dil olarak İngilizce olacak şekilde yapılandırılmıştır. Bu sayede farklı ülkelerden (örneğin Türkiye) kişilerin de hakkındaki İngilizce sayfalardan bilgi çıkarılabilmektedir.

IF NER, sonuçta resim, meslek, unvan ve doğum tarihi gibi bilgileri gruplanan dokümanlardan çıkarır.

INTERNET FALCON NER

Hasatlamak İçin Kişi Adı Yazınız

Ara Tümünü Hasatla Grupları Temizle

URL Sonuçlar

Temizle Kaldır

Yapılandırma

Arama Sayısı 10

Gruplandırma Benzerliği 1.0

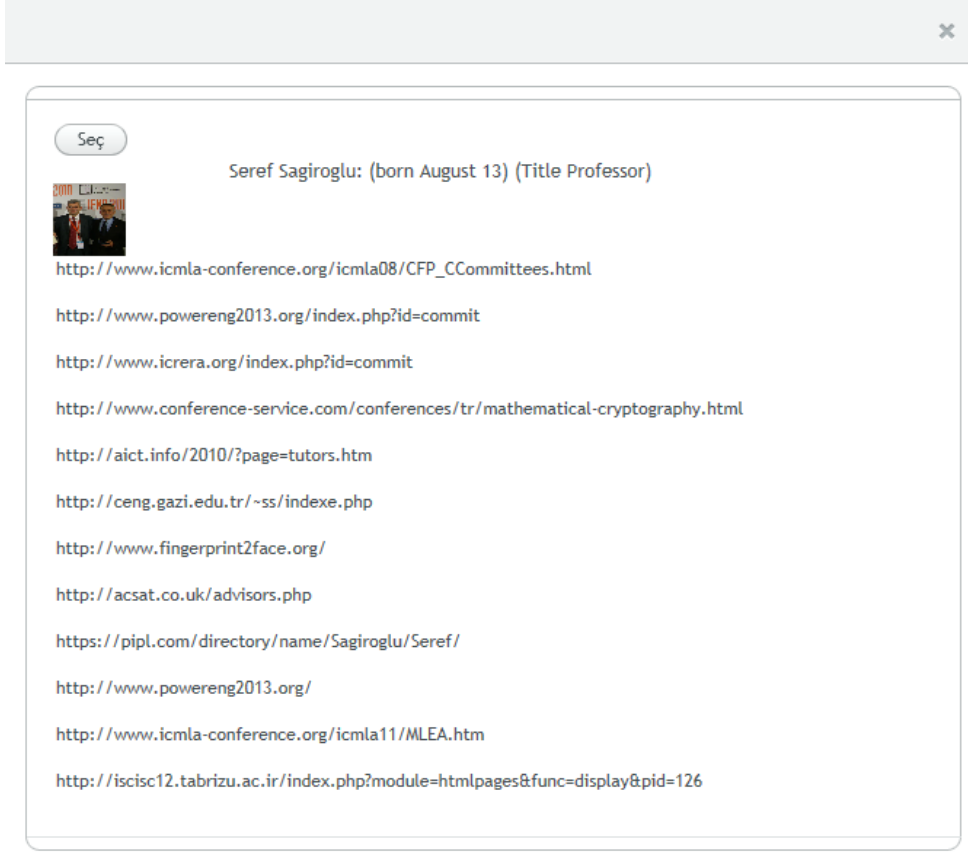
Min. Benzerlik 0.4

Test Verisi Seç

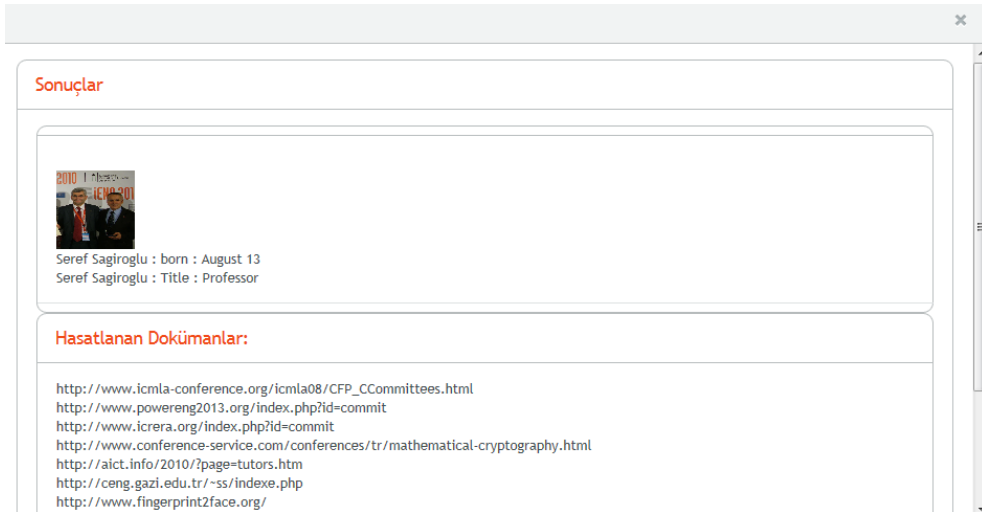
IAF Test Verisi

Şekil 3.1. Geliştirilen IF NER ara yüzü

Sonraki aşamada tespit edilen sayfalar gruplandırılır.



a)



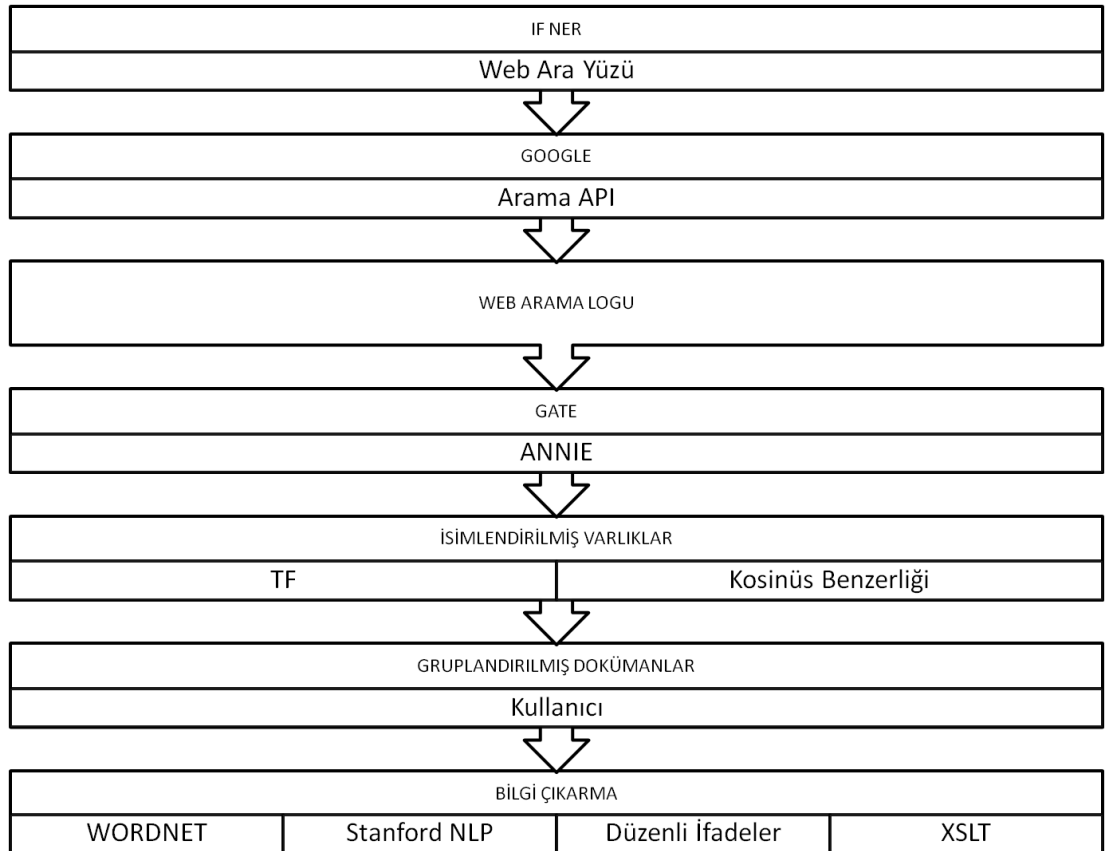
b)

Şekil 3.2. a) IF NER gruplandırma ekranı, b) IF NER hasatlama sonucu ekranı

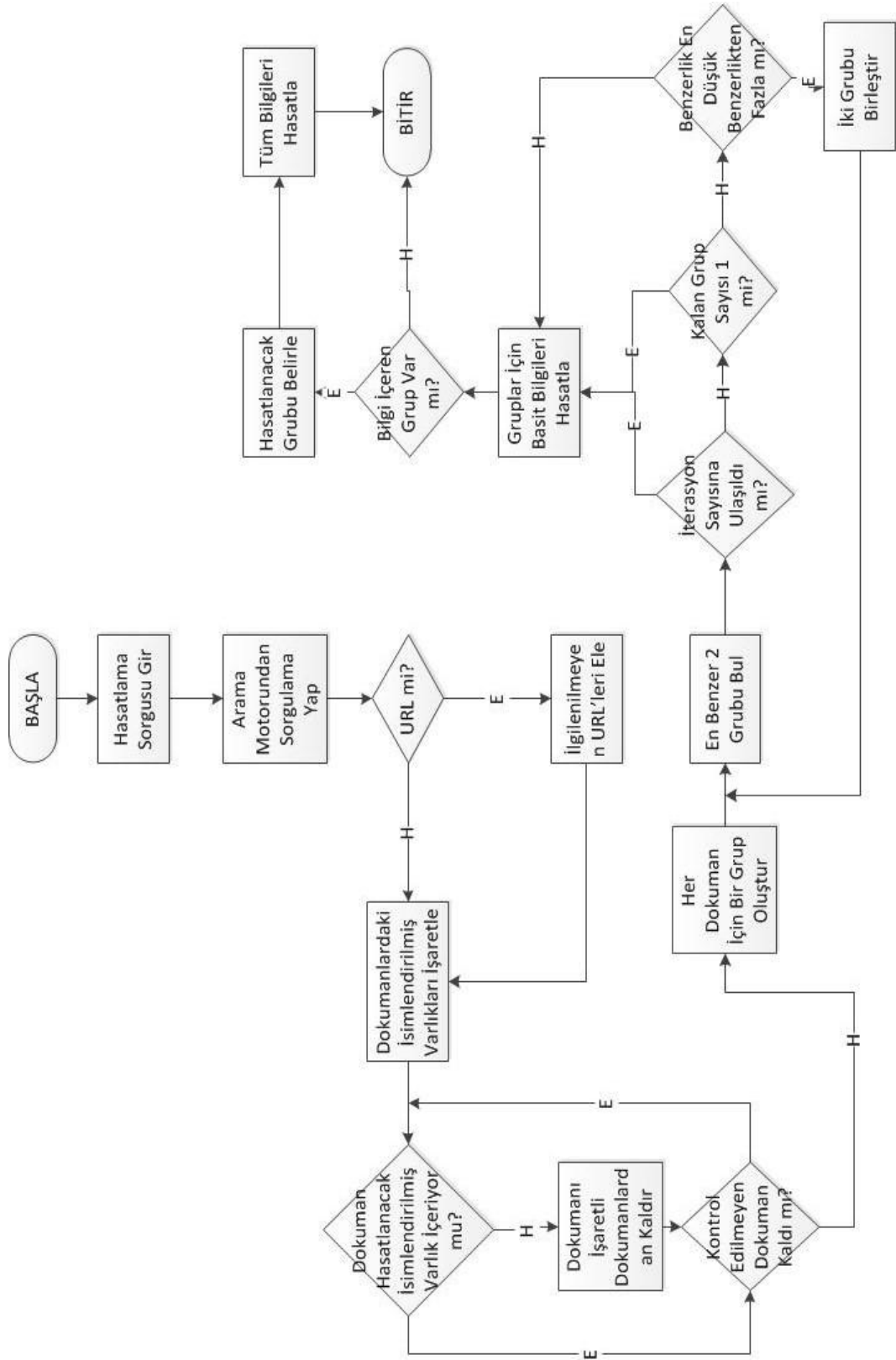
Son aşamada ise gruplandırılan sayfalardan hangi grup seçilirse, o gruptan bazı bilgiler çıkarılır ve kullanıcıya sunulur.

3.1.1. Hasatlama süreci

IF hasatlama sürecinde 2 kere kullanıcı etkileşimi olmaktadır. Bu etkileşimlerden ilki, kullanıcının hakkında hasatlama yapmak istediği kişinin ismini yazması ve hasatlama işlemini başlatmasıdır. Bir sonraki etkileşim ise IF'in adaşları farklı gruplar halinde kullanıcıya sunması ve kullanıcının hangi adaş hakkında hasatlama işlemi yapmak istediğini seçmesidir. Bu iki etkileşim dışında hasatlama süreci tamamen otomize edilmiştir. Sürecin arka tarafında farklı modüller birbiri ile iletişime geçmekte ve hasatlama işlemi tamamlanmaktadır.



Şekil 3.3. IF NER hasatlama süreci

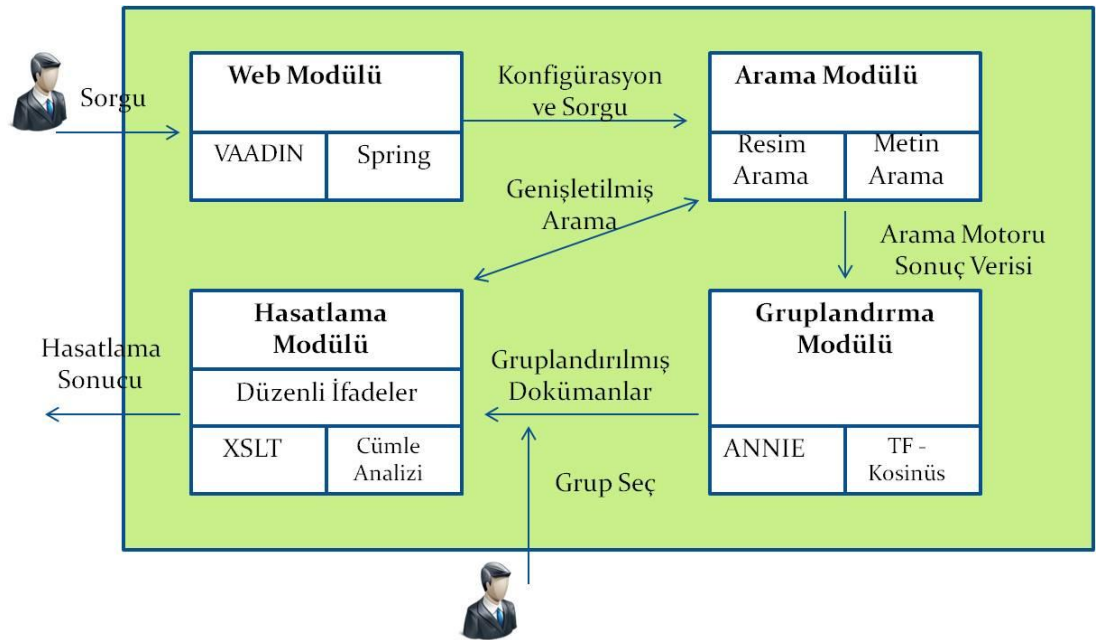


Şekil 3.4. Geliştirilen IF NER akış diyagramı

IF NER hasatlama süreci detayı için Şekil 3.4'e bakıldığında, önce kullanıcı hasatlama sorgusunu girmektedir. Daha sonra bu sorgu arama motoruna gönderilir. Daha sonra elde edilen sonuçlar URL mi şeklinde kontrol edilir. Bunun nedeni IF NER'in arama motoru sorgu sonucu dışında dokümanları da hasatlama alt yapısı olmasındandır. Arama motorundan dönen URL sonuçları içerisinde ilgisizler elenir. IF NER, sadece HTML dosyalarını işlediği için pdf, xls gibi farklı uzantıdaki URL bilgilerini eler. Kalan URL bilgileri ANNIE'ye gönderilir ve bu dokümanlar içerisindeki isimlendirilmiş varlıklar tespit edilir. Eğer dokümanlar içerisinde hasatlanacak tipte (meslek, tarih vb.) isimlendirilmiş varlıklar içermiyorsa bu dokümanlar da elenir. Daha sonra her doküman bir grup olacak şekilde grup benzerliklerine bakılmaya başlanır. Benzer dokümanlar birleştirilir ve kullanıcıya hangi grup için hasatlama yapacağı sorulur. Hasatlanacak grup belirlendikten sonra asıl hasatlama işlemi gerçekleştirilir ve sonuçlar sunulur.

3.1.2. Mimarisi

IF Web tabanlı bir uygulamadır ve bir tarayıcı yardımı ile çalışmaktadır. Toplamda 4 ana modülden oluşmaktadır. Bu modüller devam eden satırlarda kısaca açıklanmıştır.



Şekil 3.5. IF NER mimarisi

Web modülü

Web modülünde kullanıcıya bir ara yüz sunulmuştur. Kullanıcı sistem ile bu modül üzerinden iletişime geçmektedir. Web modülü kullanıcıya sistemi kullanması ve bazı yapılandırma işleri için bir takım alanlar sunar. Ana ekran 2 kısımdan oluşmaktadır.

İlk kısımda kullanıcının sorgu yapabilmesi için bir metin alanı vardır. Bu alana düz metin (genellikle kişi ismi) girerek arama yapabilmektedir. Arama sonuçlarını URL listesi olarak görebilmekte ve bu sonuçlar üzerinden hasatlama işlemini başlatabilmektedir. Kullanıcı isterse bu listeye ek olarak yeni URL'ler ekleyebilir ya da bazılarını çıkarabilir. Hasatlama işleminin ilk evresi sonra erdiğinde bir pencerede gruplar ve gruplara ait özet bilgiler kullanıcıya gösterilir ve hasatlama işleminin yapılacağı grup seçilir. Sonuç olarak hasatlama sonuçları birinci kısımda gösterilir.

Ana ekranın ikinci kısmında ise yapılandırma kesimi bulunmaktadır. Web verisinin düzensizliğinden dolayı bazı parametreler değiştirilerek daha doğru sonuçlar elde edilebilmektedir. Bu parametreler şu şekildedir:

Arama Sayısı: Arama motorlarını kullanarak girilen sorguya göre getirilecek sonuç sayısıdır.

İlk Gruplandırma Değeri: Gruplandırma işlemine başlandığında başlangıç değeri olarak gerekli benzerlik değeridir. Genellikle 1,0 değerini alır.

En Az Gruplandırma Değeri: Gruplandırma değeri ilk gruplandırma değerinden başlayarak bu gruplandırma değerine kadar yavaş yavaş azaltılır. Bu değerden daha az bir değer bulduysa gruplandırma işlemi sonlandırılır.

Test Verisi: Önceden test edilmiş verilerden birisi seçilir. Bir test verisi kişi adı ve o kişi adı ile ilgili arama motoru sonuçlarını içerir.

Bu modül tamamen baştan geliştirilmiştir. Ara yüz teknolojisi olarak Vaadin

kullanılmıştır [62]. Vaadin, sunucu tabanlı bir çatıdır. Arka tarafta GWT (Google Web uygulama geliştirme aracı) çatısını kullanmaktadır [63].

Arama modülü

Arama modülünde Google entegrasyonu yapılmıştır. Google Custom Search kütüphanesi kullanılarak yazılımsal olarak girilen sorgu ile ilgili sonuçlar elde edilmektedir. Bunun için Java programlama dili istemci kütüphanesi kullanılmıştır. Bu şekilde yazılımsal olarak girilen bir sorguya ait arama motoru sonuçları elde edilebilmektedir. IF'nin mimarisi birden fazla arama motorunu destekleyecek şekilde geliştirilmiştir. Fakat bu aşamada sadece Google arama motoru kullanılmıştır.

Bu modül ile IF çalışmasında Google Custom Search API kullanılarak 2 farklı arama (resim ve metin tabanlı aramalar) yapılmıştır. Bunlardan ilki metin tabanlı aramadır. Hasatlanacak sayfaların tespitinde bu yöntem kullanılmıştır. Eğer ilk sayfalardan kişiye ait fotoğraf bilgisi tespit edilememiş ise bu sefer resim tabanlı arama işlemi gerçekleştirilir. Hazır bir arama kütüphanesi kullanılmış olmasına rağmen uygulama ihtiyaçlarına göre bu modül baştan geliştirilmiştir.

Gruplandırma modülü

Gruplandırma modülünde GATE çatısı kullanılmıştır. Sayfaları gruplandırmak ve sayfalar içerisinden bilgi çıkarmak için önce sayfalardaki isimlendirilmiş varlıklar çıkarılmaktadır. Bu işlem GATE'in bir eklentisi olan ANNIE ile yapılmaktadır [64]. IF çalışmasında dokümanları gruplandırabilmek için dokümanlar VSM ile ifade edilmiştir. Vektörlerin boyutunun küçük tutulması için doküman içerisinde geçen anahtar kelimeler ya da tüm kelimeler kullanılmamıştır. ANNIE ile arama modülünden elde edilen Web sayfaları GATE doküman formatına çevrilmiştir. Bu format GATE çatısı ile tüm operasyonların yapılmasına olanak sağlamaktadır. Dokümanlar GATE formatına çevrildikten sonra ANNIE eklentisi bu dokümanlar üzerinde çalıştırılır. Bu işlemden sonra her doküman için isimlendirilmiş varlıklar tespit edilmiş olur.

Elde edilen isimlendirilmiş varlıklar ile bir VSM oluşturulur. Bu sayede her doküman isimlendirilmiş varlıklar kullanılarak bir vektör ile ifade edilmiştir. Elde edilen VSM ile dokümanlar benzerliklerine göre gruplandırılmıştır. VSM ve benzerlik için kosinüs benzerliği kullanılmıştır. Bu kesim IF uygulaması içerisinde geliştirilmiştir. Web verisindeki zorluklardan dolayı Web modülünde anlatılan parametreler gruplandırma modülü içerisinde kullanılmaktadır.

Gruplandırma yapıldıktan sonra kullanıcının doğru kişiyi seçebilmesi için basitleştirilmiş bir hasatlama işlemi yapılmıştır. Bu işlemde gruplar içerisinde meslek ve resim bilgisi çıkarılmaya çalışılmıştır.

Gruplandırma modülünün ürettiği çıktı kullanıcıya sunulur. Kullanıcı bu çıktıdan bir grubu seçer ve bu seçim hasatlama modülüne girdi olarak iletilir.

Hasatlama modülü

Hasatlama modülünde, gruplandırılmış dokümanlar farklı teknikler bir arada kullanılarak bilgiler çıkarılır.

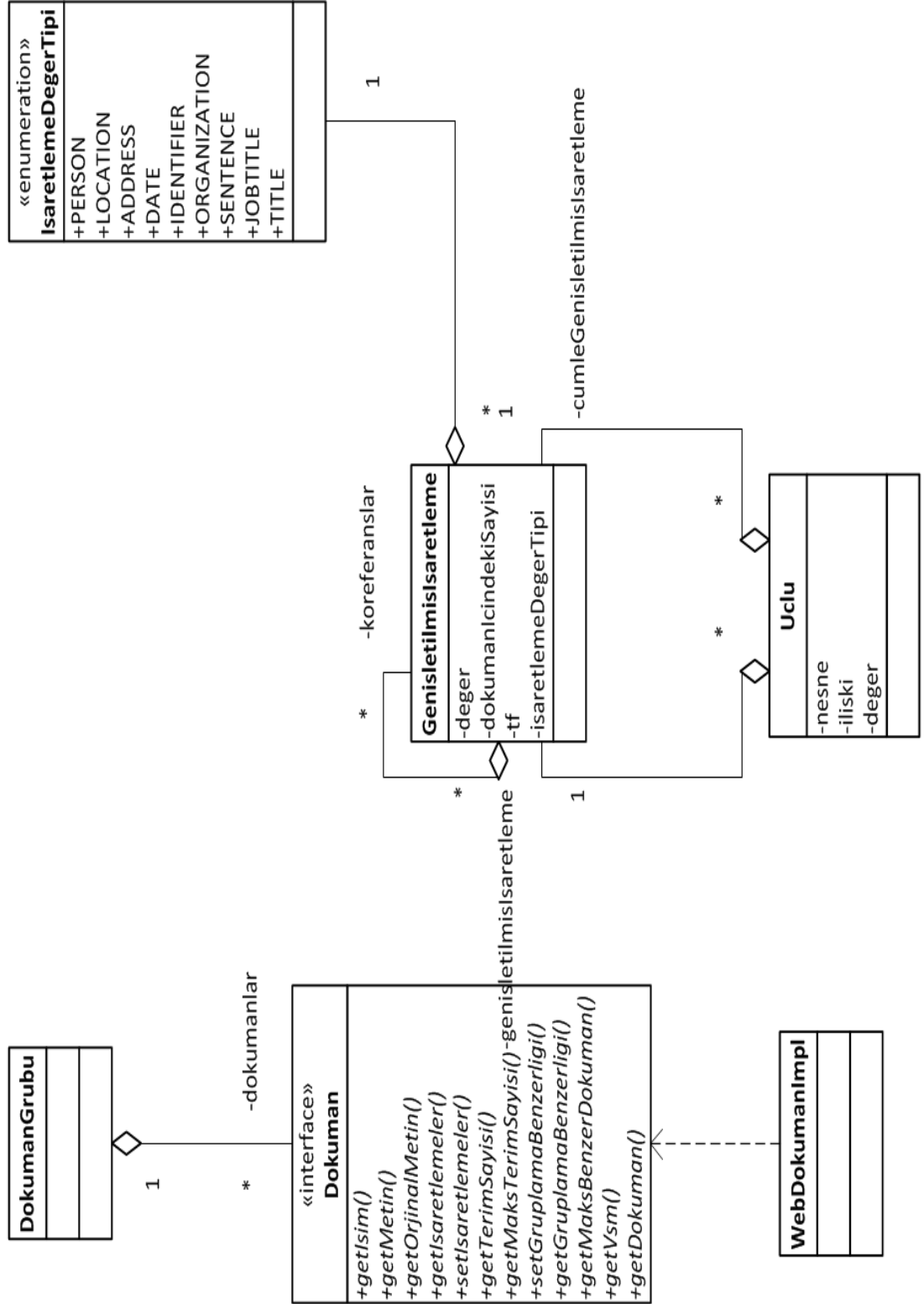
Kullanılan yöntemlerden birisi cümle analizidir. Bu yöntemde doküman içerisinde çıkarılması hedeflenen isimlendirilmiş varlıkların geçtiği cümleler analiz edilir. Cümle iki farklı şekilde analiz edilmiştir. İlkinde WordNet ve Stanford NLP'yi kullanarak cümlenin yüklemi tespit edilmiş ve işlenmiştir. Örneğin cümlenin yüklemi “birth” ise cümle içerisinde geçen tarih bilgisi doğum tarihi bilgisini ifade etmektedir. Diğerinde ise cümlenin yüklemi analiz edilmeden cümle içerisinde kişi ismi ve çıkarılması hedeflenen isimlendirilmiş varlık varsa bu bilgi o kişiye aittir varsayımı yapılmaktadır.

Kullanılan diğer bir yöntem ise XSL (genişletilebilir biçimlendirme dili ailesi) dönüşümüdür. Web üzerinde bazı sayfalar belirli formatlara sahiptir. Örneğin Wikipedia sayfaları için bir düzenli ifade ve XSL dönüşümü ile kişisel bilgiler çıkarılmıştır. Öncelikle sayfa içerisinde bilgi içeren kısım düzenli ifade ile tespit

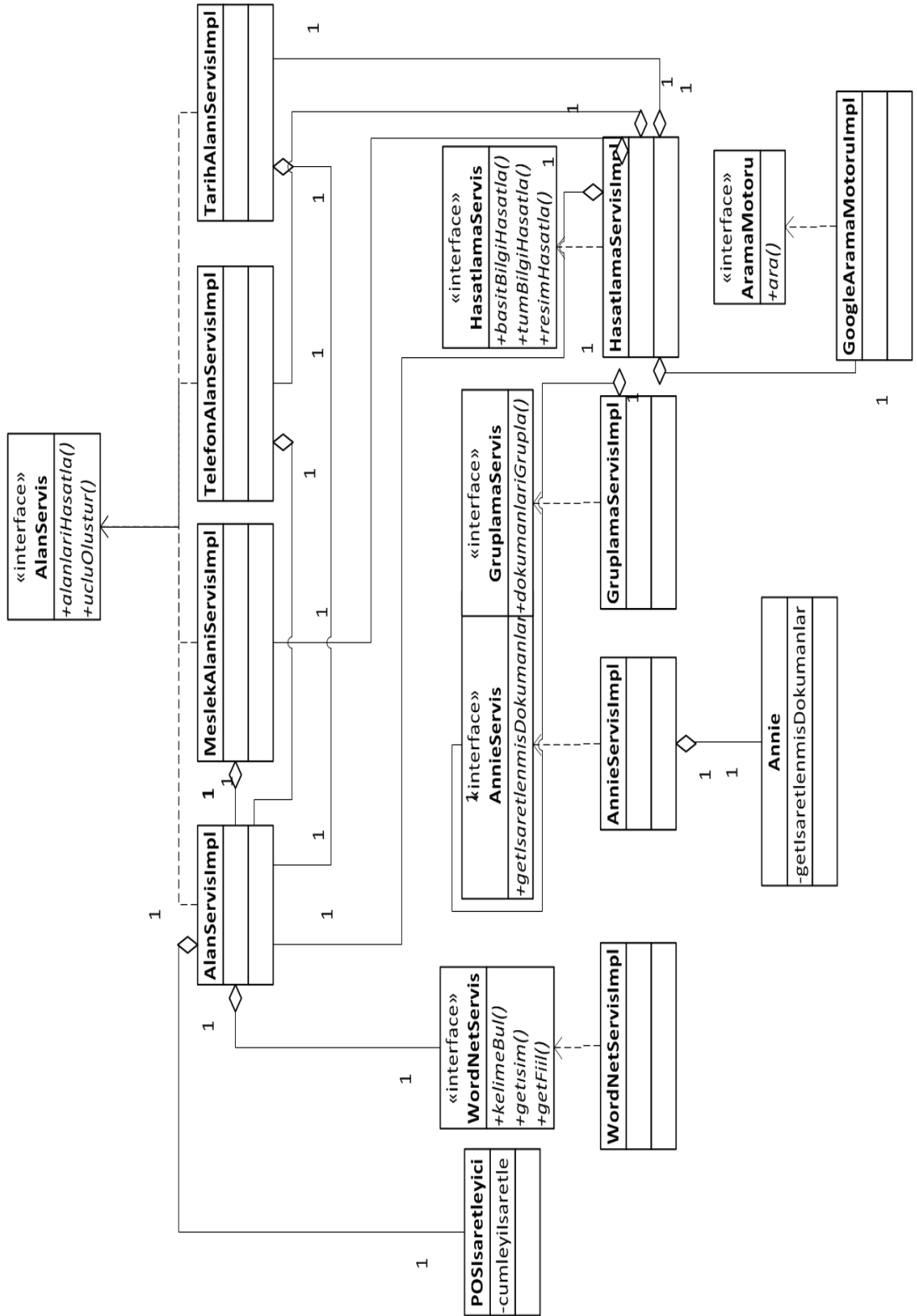
edilir. Örneğin bu bir HTML table ise, "<table id=\"persondata\">" şeklinde bir metni bulan düzenli ifade ile tespit edilmektedir. Bu kesim tespit edildikten sonra tablonun kapanma "</table>" yeri bulunarak bilgi içeren kısım ayrılmaktadır. Bundan sonra XSL dönüşümü ile HTML tablo şeklindeki bilgi düz ve IF'nin işleyebileceği bir formata çevrilmektedir.

IF NER, düz metin dışında, resim bilgisi de hasatlayabilmektedir. Resim bilgisi hasatlamak için genişletilmiş arama yöntemi kullanılmıştır. Gruplandırılan dokümanlar içerisinde belirlenen kriterlere uygun bir resim bilgisi bulunursa bu resim hasatlanmaktadır. Eğer bir grup içinde resim bulunamaz ise, meslek ve doğum tarihi bilgileri ile mevcut arama sorgusu genişletilir. Örneğin "Kişi ismi OR meslek OR Doğum tarihi" şeklinde yeni bir sorgu üretilerek arama modülünden resim araması yapılması istenir. Tespit edilen resimler tekrar kontrolden geçirilir ve resim bilgisi bulmaya çalışılır.

IF NER model tasarımında ANNIE'nin tüm işaretleyicilerini kullanmamak için "IsaretleyiciDegerTipi" numaralandırması tanımlanmış ve sadece ilgilenilen işaretlemelerin yapılması sağlanmıştır. Elde edilen tf (terim frekansı), işaretlemenin doküman içerisinde geçme sayısı gibi bilgilerin tutulması için "GenişletilmişIsaretleme" sınıfı oluşturulmuştur. Hasatlama sonucu elde edilen semantik bilgiler "Uclu" sınıfı tipinde elde edilmiştir. Arama motorundan elde edilen dokümanlar ve bu dokümanların benzerliklerinden faydalanarak elde edilen doküman grupları da Şekil 3.6'da verilmiştir.



Şekil 3.6. IF NER model sınıf diyagramı



Şekil 3.7. IF NER servis sınıf diyagramı

IF NER'in servis yapısına göre sınıf tasarımı daha basittir. Hasatlama işleminde asıl

karmaşık olan süreçtir. “AlanServis” herhangi bir alanı hasatlayacak olan servisin gerçekleştirmesi gereken bir ara yüzdür. Meslek, tarih ve telefon bilgileri için farklı servis gerçekleştirmeleri mevcuttur. Bunun dışında diğer alanlar için “AlanServisImpl” kullanılmaktadır. Bu servis, alan özel işlemlerim gerek olmadığı bilgileri hasatlamak için kullanılmıştır. Örneğin “TarihAlaniServisImpl” sınıfı sadece doğum tarihi hasatlamak için cümle analizi yapmaktadır. Bu nedenle de ona özel bir geliştirme yapılmıştır. Cümle analizinde kullanılan WordNet işlevlerini “WordNetServisImpl” servisi gerçekleştirmektedir. Hasatlama ve gruplama işlemlerinin gerçekleştirildiği farklı servis tanımlarında yapılmıştır. Google arama motoru ile ilgili sınıf “AramaMotoru” ara yüzünü gerçekleştirdiği için, bu arayüzü gerçekleştiren farklı arama motoru geliştirmeleri de yapılabilir. Örneğin Bing arama motorunun istemci yazılımını kullanan ve “AramaMotoru” ara yüzünü gerçekleştiren bir servis geliştirilerek uygulamaya 2 arama motoru ile çalışma özelliği kolayca kazandırılabilir.

3.1.3. Kullanılan teknolojiler

IF NER’de farklı teknolojiler bir arada kullanılarak hasatlama işlemi gerçekleştirilebilmektedir. IF NER’de kullanılan teknolojiler aşağıdaki başlıklarda açıklanmıştır.

Spring Framework

Günümüzde Web uygulamaları için çokça çatı bulunmaktadır. Spring Framework, günümüzde büyük çaplı birçok kurumsal yazılımda kullanılmaktadır [65].

Google Custom Search

Google Custom Search herhangi bir Web sitesinin kendi içerisinde özelleştirilmiş bir arama motoru olmasını sağlamaktadır [66]. IF NER ise Custom Search’ün bir istemci kütüphanesini kullanmaktadır. Bu istemci kütüphane her aramada 10 adet sonuç dönmektedir. Her arama sonucunun bir indis değeri vardır. Daha fazla sonuç

istendiğinde ise başlangıç indis değeri set edilerek sonraki 10 sonuç elde edilebilir. Örneğin 30 sonuç listelenmek isteniyorsa, ilk aramada başlangıç indisi 1 olmalıdır. Bir sonraki aramada 11, sonuncuda ise 21 olmalıdır. Bu şekilde 3 arama yaparak ilk 30 sonuç elde edilebilir.

IF NER’de yapılan aramalarda varsayılan arama dili olarak İngilizce kullanılmıştır. Bunu Google’ın gelişmiş arama seçeneklerinden dil olarak İngilizceyi seçip arama yapmaya benzetebiliriz.

GATE

IF NER’de GATE’in ara yüzü yerine GATE Embedded kullanılmıştır [67]. GATE Embedded, nesneye yönelik, Java programlama dili ile geliştirilmiş açık kaynak kodlu bir çatıdır. Bu çatı kullanılarak GATE’in sağladığı dil işleme işlevlerini bir uygulama içerisinde kullanmak mümkündür. Bu çatı yazılım geliştiriciler için hazırlanmıştır.

Gruplanacak ve hasatlanacak URL’ler tespit edildikten sonra GATE Embedded ile ANNIE eklentisi çalıştırılır. ANNIE, öncelikle URL’lerden kendi formatında birer doküman oluşturur ve şu işaretlemeleri yapar: “PERSON, LOCATION, ADDRESS, DATE, IDENTIFIER, ORGANIZATION, SENTENCE, JOBTITLE, TITLE”. Bunlar sırasıyla kişi, mekân, adres (URL, eposta, vs.), tarih, kimlik bilgisi, organizasyon, cümle, meslek ve unvandır. ANNIE bunun dışında yüzde (%), tesis adı gibi farklı işaretlemeler de yapmaktadır fakat IF çalışmasında performans kısıtlarından dolayı sadece yukarıdaki işaretlemeler kullanılmıştır. ANNIE bu işaretlemeleri yaparken Jape kurallarını kullanır. Jape (Java Annotations Pattern Engine) dokümanlardaki işaretlemeler üzerindeki düzenli ifadelerin tanınmasını sağlar [67]. Bu kurallar iki kısımdan oluşmaktadır. Öncelikle kuralın aktif olması gereken koşulların sağlanması gerekir. Daha sonra kural aktif olur ve bir işlem yapılır. Örneğin Çizelge 3.1’deki kuralda “30 pounds” yapısındaki metinler para olarak işaretlenmektedir. İlk kısımda metin bir sayı içermeli, daha sonra bir boşluk olmalı ve ondan sonra da para birimlerinden birisi olmalıdır. Bu koşulları sağlayan

bir metin parçası para olarak işaretlenmektedir. Bu şekilde dokümanlar içerisinde işaretlemeler yapılmaktadır.

Çizelge 3.1. ANNIE Jape kural örneği [64]

```
Rule: Money1
// e.g. 30 pounds
(
  (AMOUNT_NUMBER)
  (SpaceToken.kind == space)?
  ({Lookup.majorType == currency_unit})
)
:money -->
:money.Number = {kind = "money", rule = "Money1"}
```

Kişi, tarih, meslek gibi diğer işaretlemeler için de kural tanımları ANNIE içerisinde mevcuttur. ANNIE bu işlemleri İngilizce metinler üzerinde yapabilmektedir. Bu nedenle arama sonuçlarında farklı dillerde sonuçlar geldiği zaman bu dokümanlar için performansı düşmektedir.

WordNet

WordNet, İngilizce isim, sıfat, fiil ve edatları içeren sözlüksel bir veritabanıdır [68]. Ayrıca eşanlamlı ve zıt anlamlı gibi bilgileri de içerir. Dil işleme çalışmalarında sıklıkla kullanılmaktadır. IF’de kullanılan çok önemli bir özelliği daha vardır. Verilen bir fillin kök halini bulabilmesidir. Bu sayede bazı semantik ilişkiler çıkarılabilmektedir. Örneğin doğum tarihi bilgisine bu şekilde ulaşılabilir. Bir cümlede geçen fiilin kök hali WordNet ile bulunup, daha sonra doğum ile ilgili fiil kökleri ile karşılaştırılırsa (give birth, bear, birth), o cümledeki tarih bilgisinin doğum tarihi olduğu anlaşılabilir.

Stanford POS Tagger

Bir POS işaretleyicisi cümledeki isim, fiil, sıfat gibi grupları tespit eder. Stanford POS işaretleyicisi bir cümledeki bu alanları tespit etmektedir [69]. IF’de ANNIE’nin sağladığı cümleler içerisinde WordNet ile beraber semantik ilişkileri çıkarmak için

kullanılmıştır. Genellikle verilen bir cümlemin yüklemine bulmak için kullanılmıştır. Cümle analizi ile doğum tarihi belirleme işlemi bu şekilde yapılmıştır. Tespit edilen tarih bilgisinin doğum tarihi olduğu cümlemin yüklemi ile belirlenmektedir.

Düzenli ifadeler

Düzenli ifadeler bir metin içerisinde belirli bir metni seçmek için kullanılır [70]. Örneğin “\b20\d\d[0-9]\d\d\b[9\d\b” düzenli ifadesi ile 1000-2099 arasındaki yıllar tespit edilebilmektedir. Düzenli ifadeler alan bilgisinden bağımsızdır. Herhangi bir alan için farklı düzenli ifadeler yazılabilmektedir.

XSLT (Genişletilebilir biçimlendirme dili dönüşümleri)

XSLT, XML dokümanlarından başka XML dokümanlarının oluşturulması için kullanılan bir dildir [71]. XSLT, XSL’in bir parçasıdır. XSL, XML’in formatlanması için gerekli olan sözlüğü içerir. XSL, XML’in nasıl biçimlendirildiği ile ilgilenir, XSLT ise nasıl transfer edildiği ile ilgilenir. Hasatlama tekniklerinden biri olarak kullanılan XSLT, Wikipedia üzerinde hasatlamada kullanılmıştır.

3.1.4. Kullanılan yöntemler

IF, farklı yöntemleri bir arada kullanarak sayfaları gruplandırır ve bu sayfalar içerisinde kişisel bilgileri hasatlar. Bu çalışmada kullanılan yöntemler aşağıda alt başlıklarda açıklanmıştır.

İlgisiz sayfaların elenmesi

IF hasatlama işlemi yapmayacağı dokümanlara farklı zamanlarda karar verebilmektedir. Gruplama aşamasından önce URL bilgisinden bazı sayfalar elenebilmektedir. Bunlar üzerinde işlem yapamadığı sayfalardır. IF sadece HTML sayfaları üzerinden işlem yaptığından “.pdf”, “.doc”, “.docx”, “.xls” ve “.xlsx” uzantılarına sahip URL’ler ilk aşamada elenir. Bu dokümanlar üzerinde herhangi bir

işlem yapılmaz. Bunun nedeni kullanılan test verilerinde sadece HTML sayfalarının hasatlanacak bilgileri içermesidir. Diğer bir sayfa eleme yöntemi ise, GATE dokümanları oluşturulduktan ve ANNIE ile isimlendirilmiş varlıklar tespit edildikten sonra yeterli bilgi içermeyen sayfaların elenmesidir. Sayfalar gruplandırıldıktan sonra özet bilgi olarak meslek, unvan ve tarih bilgileri gösterilmektedir. Eğer bu türlerden herhangi bir isimlendirilmiş varlık içermiyorsa o doküman da elenir. Bu tip dokümanlar az bilgi içeriyor olarak kabul edilir ve hasatlama performansı artırılmaya çalışılır.

Tf katsayısı

IF çalışmasında dokümanlar içerisindeki tüm terimlerin/ kelimelerin frekansları kullanılmamıştır. Bunun yerine ANNIE ile çıkarılan isimlendirilmiş varlıkların frekanslarına bir anlam yüklenmiştir. Bu isimlendirilmiş varlıkların çıkarılması uzun süren bir işlemdir fakat bu varlıklar hem sayfaların gruplandırma işleminin hızını artırmakta hem de hasatlama işleminde kullanılmaktadır. tf, Eş. 2.19’da verilen ifade kullanılarak hesaplanmaktadır.

Kosinüs benzerliği

Benzer dokümanların gruplandırılması için kosinüs benzerliğinden faydalanılmıştır. Öncelikle her dokümandaki terimlerin tf değerleri kullanılarak birer VSM oluşturulmuştur. VSM olarak ifade edilen her doküman farklı bir grupta olacak şekilde belirlenmiştir. Daha sonra en benzer iki grup tek bir grupta olacak şekilde birleştirilmiştir. Benzerliğin belirlenmesi için kosinüs benzerliğinden yararlanılmıştır. Kosinüs benzerliği Eş. 2.13’te verilen ifade kullanılarak hesaplanmıştır. Terimlerin dokümanlar içerisindeki öneminden çok dokümanların benzerlikleri ile ilgilenildiğinden, formülde tf-idf katsayıları yerine tf katsayıları kullanılmıştır. Burada tamamen aynı olan terimler kullanılmamıştır çünkü Web verisinde aynı terimler farklı şekillerde yazılmış olabilir ya da web verisi yazım hataları içerebilir. Bu yüzden terimin türüne göre belirli bir benzerlik hesabı yapılmaktadır. Terim benzerlikleri için terimin türüne göre 4 farklı hesaplama

yapılmıştır. Bunlar adres (URL ve e-posta), tarih ve diğerleridir.

Adres: Adres benzerliğinde öncelikle terimin e-posta mı yoksa URL mi olduğu kontrol edilir.

E-Posta: E-posta benzerliğinde domain bilgisi ya da isim bilgisi diğer e-posta tarafından içeriliyorsa bu iki e-posta benzerdir. Örneğin “omererakman@gmail.com” e-posta bilgisi ile “omererakman@yahoo.com” e-postaları benzerdir çünkü “omererakman” isim bilgisi iki e-postada da mevcuttur.

URL: URL benzerliğinde öncelikle popüler mi kontrolü yapılır. Eğer bir URL popülerler listesinde ise benzer değildir. Bunlar: “facebook”, “twitter”, “flicker”, “linkedin”, “plus.google”, “wikipedia” ve “slideshare” dir. Eğer URL bu listeden birini içeriyorsa gruplandırma işlemi sırasında kullanılmayacaktır. Bunun nedeni popüler URL bilgileri hemen her sayfada olabilir. Bu yüzden gruplandırma performansını kötü etkileyebilir. Eğer URL bilgileri popüler olanlardan değilse bu sefer sözlüksel olarak benzerliklerine bakılır. URL’lerden “http”, “.com” gibi hemen her URL’de olan metinler kaldırılır, daha sonra “.” karakterine göre parçalara ayrılır ve bu parçaların benzerliklerine bakılır. Benzer bir parça bulunduğu URL’ler benzer kabul edilir.

Tarih: Web verisinde tarih bilgileri aynı tarihi göstermelerine rağmen çok farklı formatlarda yazılabilmektedir. Bu nedenle iki tarih bilgisinin benzer/ aynı olduğuna karar vermek zor bir iştir. Öncelikle iki tarih metni tamamen aynı ise o zaman benzer kabul edilmiştir. Aynı değil ise tarih bilgilerinden yıl parçaları düzenli ifadeler ile çıkarılmıştır. Yıl bilgisi aynı olan tarihler benzer kabul edilmiştir.

Yıl bilgisini çıkarmak için kullanılan düzenli ifade “\b20\d\d[1[0-9]\d\d\b|9\d\b” dir. Bu düzenli ifade ile 1000-2099 yılları arasındaki yıl bilgileri çıkarılabilmektedir. Ayrıca “98” gibi 2 basamaklı formatta yazılmış yıllar da tespit edilebilmektedir.

Cümle analizi

IF, hasatladığı bilgilerin bir kısmını cümle analizi yaparak elde etmektedir. Cümle analizi ile elde edilen bilgiler:

Meslek: Meslek bilgisi ANNIE'nin "JobTitle" olarak işaretlediği alanlardır. ANNIE aynı zamanda cümleleri de işaretlemektedir. Meslek olarak işaretlenmiş varlıkların geçtiği cümleler tespit edilir. Bu cümleler içerisinde kullanıcının ilgilendiği konu (kişi ismi) geçiyorsa ya da kişi isminin sözlüksel olarak benzediği bir metin geçiyorsa meslek bilgisinin o kişiye ait olduğu varsayılır. Bu şekilde kişinin meslek bilgisi çıkarılır.

Unvan: Unvan bilgisi de meslek bilgisine benzer bir şekilde hasatlanır. ANNIE unvan bilgisini "Title" olarak işaretlemektedir. Bazı durumlarda unvan ve meslek bilgileri birbiri yerine kullanılabilir. Bu nedenle bazen bu iki kavram eş anlamlı olacak şekilde kullanılmıştır.

Doğum Tarihi: IF cümle analizi ile farklı tarih bilgilerini hasatlayabilecek alt yapıya sahiptir. Bu çalışma kapsamında cümle analizi ile tarih olarak sadece doğum tarihi bilgisi hasatlanmaktadır. Öncelikle meslek ve unvan bilgilerine benzer şekilde "Date" olarak işaretlenmiş varlıkların geçtiği cümleler tespit edilmektedir. Daha sonra kişi ismini barındıranlar filtrelenmektedir. Filtrelenen cümleler StanfordPOS tagger ile işaretlenmektedir. Bu POS işaretleyicisi ile özne, yüklem gibi parçalar tespit edilebilmektedir. İşaretlenen cümlenin yüklemine WordNET veritabanından faydalanarak kök hali elde edilmektedir. Bu şekilde cümlenin yüklemi "born" ise o tarih bilgisinin doğum tarihi olduğu çıkarımı yapılmaktadır. Örneğin "Ömer was born in 1984." Şeklindeki bir cümle analiz edildiğinde "was born" yüklemi tespit edilmektedir. WordNET yardımı ile bu fiilin kök halinin "born" olduğu anlaşılmaktadır. Bu şekilde "Ömer : born 1984" bilgisi elde edilebilmektedir. Buna benzer şekilde farklı tarih bilgileri de çıkarılabilir. Örneğin bir kişinin ölüm tarihi bilgisi bu şekilde elde edilebilir.

Düzenli ifadeler

Düzenli ifadeler, yapılan çalışmanın farklı yerlerinde farklı amaçlar için kullanılmıştır. Örneğin; tarihlerin benzerliklerini hesaplarken yıl bilgisinin çıkarılmasında, söz dizilerini alt parçalara ayırırken kullanılmıştır. Bunun dışında Wikipedia'dan bilgi hasatlarken bilgiyi içeren XML parçalarının tespit edilmesinde kullanılmıştır. Düzenli ifadeler esas olarak hasatlama işleminde kullanılmamış olsa da hasatlama ve gruplama işlemlerinde yardımcı yöntem olarak kullanılmıştır.

XSL dönüşümü

IF belirli bir yapıya sahip Web sayfaları için XSL dönüşümünü kullanarak bu sayfalardan bilgi çıkarma işlemini yapabilmektedir. Wikipedia göreceli güvenilir bilgi kaynaklarından birisidir. Bu nedenle de hasatlama işlemi gerçekleşirken Wikipedia sayfalarından bir veya birkaçı varsa bunlar için özel bir işlem uygulanmaktadır. Wikipedia'da kişi bilgilerinin içerdiği bir HTML tablo vardır. Bunu karmaşık olmayan bir XSLT ile hasatlamak mümkündür. IF önce bu tabloyu tespit etmekte daha sonra tespit edilen tablo sayfanın içerisinden çıkarılmaktadır. Elde edilen HTML tablo bir XML'dir. Örneğin “Jessica Fox” ile ilgili bir sayfada elde edilen tablo şu şekildedir:

EK-3'te verilen Çizelge 3.1'de HTML tablosu ikişer kolondan oluşan birden fazla satır içeren bir yapıdadır. Bu yapıdaki bir veriden üçlü yapılar elde edilebilmektedir. Bu üçlü yapıları elde edebilmek için bir XSL hazırlanmıştır. Hazırlanan XSL ile her yeni bir bilginin başına “%”, bilgi ile bilgiyi anlatan ilişki ile arasına “?” karakteri eklenmiştir. Bu şekilde XSL dönüşümü sonucunda belirlenmiş özel karakterler ile ayrılmış bir metin ortaya çıkmıştır.

Çizelge 3.2. “PersonData” hasatlama XSL

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <xsl:for-each select="table/tr">
      <xsl:if test="not(td = '')">
        %<xsl:value-of select="td[@class='persondata-label']" />
        ? <xsl:value-of select="td[not(@class='persondata-label')]" />
      </xsl:if>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>

```

Elde edilen metin eklenen özel karakterlere göre parçalara ayrılarak üçlü bilgiler elde edilmiştir.

Çizelge 3.3. Wikipedia’dan “PersonData” bilgisi hasatlama [72]

Kişi İsmi	İlişki	Değer
Jessica Fox	Name	Fox, Jessica
Jessica Fox	Alternative names	-
Jessica Fox	Short description	Australian canoeist
Jessica Fox	Date of birth	11 June 1994
Jessica Fox	Place of birth	Marseille, France
Jessica Fox	Date of death	-
Jessica Fox	Place of death	-

Bunun dışında Wikipedia’nın VCard bilgisi de benzer şekilde hasatlanmaktadır. EK-3’te verilen Çizelge 3.2’de VCARD yapısı görülebilmektedir. XSL dönüşümü yapıldıktan sonra kişiye ait bilgiler elde edilmektedir.

Belirli bir yapıda olan, genellikle kişinin biyografi bilgilerini içeren veriden kişiye ait bilgiler XSL dönüşümü ile hasatlanmaktadır.

XSLT dönüşümü ile semantik ilişkideki üçlü bilgisi (nesne-ilişki-değer) elde edilir. Çizelge 3.5’te XSLT ile elde edilen bilgilere bir örnek verilmiştir. Bu yöntem hem hız açısından hem de belirli bir yapıya sahip herhangi bir Web sayfasına

uygulanabilmesi açısından başarılı bir yöntem olmuştur.

Çizelge 3.4. “VCard” hasatlama XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <xsl:for-each select="table/tr">
      <xsl:if test="not(td = '')">
        %<xsl:value-of select="th[@scope='row']" />
        ? <xsl:value-of select="td" />
      </xsl:if>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Çizelge 3.5. Wikipedia’da “VCard” bilgisi hasatlama [73]

Kişi İsmi	İlişki	Değer
Jessica Fox	Place of birth	Taplow, Berkshire
Jessica Fox	Years active	1992–present
Jessica Fox	JobTitle	Actor

Genişletilmiş arama

IF, kişilere ait resimleri 2 aşamada hasatlamaktadır. İlk aşamada gruplandırma işlemi sonlandığında her grup için özet bilgiler hasatlanırken, gruba ait dokümanlar içerisinde kişinin fotoğrafını tespit etmeye çalışır. Bunun için ANNIE’nin resim olarak tespit ettiği varlıklar içerisinde bir filtreleme yapılmaktadır. HTML resmi () ifade eden iki önemli özellik kullanılmıştır. Bunlardan ilki “alt” bilgisidir. Arama motorlarının da resimler hakkında bilgi edinmesini sağlayan bu özelliğin metni incelenir. Bunun dışında resmin “src” özelliği incelenir. Bu iki özellik resmin o kişiye ait olup olmadığı hakkında bilgi verebilmektedir. Örneğin “Jessica Fox” kişisi için bir resmin özellikleri şu şekilde olabilir: “src=//upload.wikimedia.org/wikipedia/commons/thumb/6/6a/Jessica_Fox_actress.jpg/220px-Jessica_Fox_actress.jpg”, “alt=”. Bu özellikler içerisinde kişi isminin farklı formatları bulunur ise resmin o kişiye ait olduğu anlaşılmaktadır. Bu formatlar şu

şekildedir: “Jessica Fox, Jessica-Fox, Jessica_Fox, JFox, JessicaF”. Daha sonra src bilgisi ve Web sayfasının sunucu bilgileri kullanılarak resime ulaşılacak URL üretilir ve gruplar ile beraber gösterilir.

Gruplandırma aşamasında kişi resmi tespit edilemez ise hasatlama aşamasında arama motorundan bir resim araması yapılarak kişi resmi tespit edilmeye çalışılır. Bunun için de kişisel bilgileri hasatlanacak kişiye özel resmi bulabilmek için mevcut arama genişletilir. Örneğin “Seref Sagiroglu” araması için gruplandırılan dokümanlar içerisinden uygun resim bulunamazsa, gruplandırmada gösterilen özet bilgiler kullanılarak “Seref Sagiroglu” araması genişletilir. Doğum tarihi olarak “1 August”, unvan olarak da “Prof.” tespit edilmiş ise, genişletilmiş arama şu şekilde olacaktır: “Seref Sagiroglu Prof. OR 1 August”. Burada arama motorlarında kullanılan “OR” işlecidir. Hazırlanan sorgu şu anlama gelmektedir: “Seref Sagiroglu” ve “Prof” ya da “1 August” bilgilerini içeren resimleri getir. Bu şekilde yapılan resim araması sonuçları birinci aşamada yapılan filtreleme sürecine benzer bir süreçten geçmektedir. Bu aşamada arama motoru sonuçları yani URL bilgileri olduğundan, kişi isminin farklı formatları bu URL’lerde bulunuyorsa, ilk bulunan URL seçilmektedir. Bu yöntem ile ilk aşamada bulunamayan kişi resimlerini resim arama özelliğini kullanarak bulmak mümkün olmaktadır.

3.1.5. Yapılan testler

IF NER çalışmasında 4 farklı test verisi kullanılmıştır. İlk test verisinde (farklı meslek test verisi) farklı meslek gruplarından 5 farklı isim ve her adaya ait 30 adet URL bilgisi bulunmaktadır. Farklı meslek test verisinde Türk ve yabancı kişilere ait isimler bulunmaktadır. İkinci test verisi ise (öğretim üyesi test verisi) Gazi Üniversitesi Bilgisayar Mühendisliği öğretim üyelerinin isimleri ve bu isimler ile arama yapılarak elde edilmiş farklı sayılarda arama motoru sonuçlarından oluşmaktadır. Öğretim üyesi test verisinin tamamı Türk isimleri içermektedir. Meslek, doğum tarihi ve sayfa sayısı bilgileri manuel olarak tespit edilmiştir. IF meslek, IF Doğum tarihi ise IF NER tarafından çıkarılan bilgilerdir. IF sayfa sayısı, IF’in bir adas için oluşturduğu gruptaki sayfa sayısı bilgisidir. IF doğru sayfa sayısı

bilgisi ise oluşturulan grup içerisindeki sayfaların kaç tanesinin gerçekten o adaş ile ilgili olduğu bilgisidir. IF NER'in Türkçe sayfalar üzerindeki davranışını ölçmek için öğretim görevlisi test verisi Türkçe sayfalar üzerinde de test edilmiştir. Son olarak diğer çalışmalar ile karşılaştırmak için IAF test verisi üzerinde de sonuçlar üretilmiştir [37].

Farklı meslek test verisi

Bu test verisi Türk ve yabancı kişilerin isimleri kullanılarak Google arama motorundan elde edilen ilk otuzar sayfayı içermektedir. Test verisi oluştururken 2 Türk akademik kişi seçilmiştir. Bunun nedeni bu kişilerin akademik tanınırlığı olan kişiler olması ve haklarında İngilizce dokümanlara ulaşma imkanı olmasıdır. Test verisini oluşturmada kullanılan yabancı isimler içerisinde bir akademik personel ve farklı mesleklere sahip olan adaşların olduğu 2 farklı isim de kullanılmıştır. Bu sayede adaşlara göre gruplandırma görülebilmektedir. Bu test verisi içerisindeki kişilere ait bilgiler bir yüksek lisans öğrencisi ve bir yüksek lisansını tamamlamış öğrenci tarafından Çizelge 3.6'daki gibi tespit edilmiştir.

Çizelge 3.6. Farklı meslek test verisi içeriği

Kişi No	Kişi Adı	Meslek	Doğum Tarihi	Resim
1	Şeref Sağıroğlu	Professor	August 1 1965	Var
2	Antonio Albanese	Chef	30 November 1947	Var
3	Antonio Albanese	Actor-director	October 10, 1964	Var
4	Antonio Albanese	Senior Vice President	-	Yok
5	Antonio Albanese	Fencer	5 December 1937	Yok
6	Erhan Karaesmen	Professor	1936	Var
7	Jessica Fox	Actor	1983-05-19	Var
8	Jessica Fox	Canoeist	11 June 1994	Var
9	Jessica Fox	Environmental engineer	-	Var
10	Jessica Fox	Model	-	Var
11	Jessica Fox	Blogger	-	Var
12	Jessica Fox	Dr.	-	Yok
13	Peter Dourmashkin	Professor	-	Var

Yapılan deneylerde “Meslek” ve “Doğum Tarihi” bilgileri test verisini hazırlayan kişiler tarafından göz ile tespit edilmiştir. “IF meslek”, “IF Doğum Tarihi”, “Resim” bilgileri ise IF NER tarafından otomatik olarak tespit edilmiştir.

Çizelge 3.7. Farklı meslek test verisi: Şeref Sağıroğlu

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
Professor	August 1 1965	25	Prof.	August 1		14	14

Şeref Sağıroğlu veri grubu IF ara yüzünden “Seref Sagiroglu” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 5 tanesinde sayfaların kime ait olduğu belli değildir. 14 sayfa ise hasatlama için gruplandırılmıştır. Geri kalan 11 sayfa IF tarafından yetersiz bilgi içerdiğinden elenmiştir. Bu veride sadece 1 kişiye ait bilgiler bulunmaktadır.

Şeref Sağıroğlu veri grubu için, gruplandırma aşaması için P ve R değerleri incelendiğinde; $P = \frac{14}{14} = 1$ olarak bulunmuştur. Bu da gruplandırmada kullanılan sayfaların tamamının aynı kişiye ait olduğunu göstermektedir. $R = \frac{14}{25} \cong 0.56$ olarak bulunmuştur. Bu değer sayfaların çoğunun gruplandırmada kullanıldığını göstermektedir. Elde edilen P ve R değerleri başarılı bir gruplandırma yapıldığını göstermektedir.

Sayfalar başarılı bir şekilde gruplandırıldığı için çıkarılan bilgilerin performansı da yüksek olmuştur. Sayfaların birinde doğum tarihi bilgisi vardır fakat hasatlanan doğum tarihi bilgisinin yıl kesimi hasatlanamamıştır. Bunun nedeni de tarih formatının Web sayfasında hatalı olmasıdır. Web sayfasındaki doğum tarihi bilgisi “August 1, .1965” şeklindedir. Cümle analizi ile hasatlama yapıldığından tarih

alanındaki virgül ve nokta karakterlerinin beraber olması bu ikisinin farklı tarih bilgileri olarak algılanmasına neden olmuştur. Meslek bilgisi doğru bir şekilde elde edilmiştir. Kişiye ait doğru resim bilgisi elde edilmiştir. Bu veri için mevcut sayfalar içerisinde var olan ve hasatlanması hedeflenen tüm bilgiler başarı ile çıkarılmıştır.

Çizelge 3.8. Farklı meslek test verisi: Antonio Albanese

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
Chef	30 November 1947	2	Chef	November 30, 1947		2	2
Actor-director	October 10, 1964	18	Senator	October 10, 1964		4	4
Senior Vice President	-	1	Senior Vice President	-	-	-	-
Fencer	5 December 1937	1	Italian fencer	1937-12-5	Yanlış	1	1

Antonio Albanese veri grubu IF ara yüzünden “Antonio Albanese” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 7 tanesi hasatlama için gruplandırılan dokümanlardır. 8 sayfa hangi kişiye ait olduğuna dair yeterli bilgileri içermemektedir. Geri kalan 15 sayfa IF tarafından yetersiz bilgi içerdiğinden elenmiştir. Yetersiz bilgi içeren sayfaların çoğu “Youtube” sayfalarıdır. IF metin üzerinden işlem yaptığı için bu sayfalarda yeteri kadar bilgi bulunamamıştır.

Çizelge 3.9. Farklı meslek test verisi: Erhan Karaesmen

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
Professor	1936	20	Professor	-		10	10

Erhan Karaesmen veri grubu IF ara yüzünden “Erhan Karaesmen” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 10 tanesi hasatlama işlemi için gruplandırmada kullanılmıştır. 10 sayfa ise kişiyi belirleme açısından yeterli bilgi barındırmamaktadır ya da işlenemez formatlardaki sayfalardır. Kalan 10 sayfa ise IF tarafından yetersiz bilgi içerdiğinden elenmiştir.

Çizelge 3.10. Farklı meslek test verisi: Jessica Fox

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
Actor	1983-05-19	7	Actor	1983-05-19		1	1
Canoeist	11 June 1994	9	Canoeist	11 June 1994		7	7
Environmental engineer	-	1	-	-	-	0	0
Model	-	1	-	-	-	0	0
Blogger	-	2	-	-	-	0	0
Dr.	-	1	-	-	-	0	0

Jessica Fox veri grubu IF ara yüzünden “Jessica Fox” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 8 tanesi gruplandırmada kullanılmıştır. 7 sayfanın hangi kişiye ait olduğu bilgisi belli değildir ya da geçersiz formatlardadır. Geri kalan 15 sayfa IF tarafından yetersiz içerdiğinden elenmiştir. Bu veride toplam 6 kişiye ait sayfalar vardır. IF sadece 2 kişiye ait bilgileri tespit etmiştir. Diğer kişiler hakkında fazla bilgi olmadığından bu bilgiler yetersiz olarak tespit edilmiş ve gruplandırma aşamasına dâhil olmamıştır.

Çizelge 3.11. Farklı meslek test verisi: Peter Dourmashkin

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
Professor	-	27	Prof.	-		21	21

Peter Dourmashkin veri grubu IF ara yüzünden “Peter Dourmashkin” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 21 tanesi yeterli miktarda bilgi içeren sayfalardır. Geri kalan 9 sayfanın 2 tanesi kime ait olduğu belli olmayan sayfalardır. Bu sayfalarda gruplandırmada kullanılan herhangi bir bilgi yoktur. 1 tanesi ise farklı bir kişiye aittir fakat bu sayfada da kişinin resmi dışında metin olarak bir bilgi bulunmamaktadır. IF tarafından geriye kalan 6 sayfa yetersiz bilgi içerdiğinden elenmiştir. Bu veride araştırılan bilgileri içeren sadece 1 kişiye ait sayfalar bulunmaktadır.

P ve R değerlerinin yanında eşit olarak ağırlıklandırılmış ortalaması olan F1 katsayısı da hesaplanmıştır.

Çizelge 3.12. IF NER farklı meslek test verisi için gruplandırma başarısı

Veriler / Katsayı	P	R	F1
Şeref Sağıroğlu	1,0	0,56	0,72
Antonio Albanese	0,87	0,32	0,50
Erhan Karaesmen	1,0	0,5	0,67
Jessica Fox	1,0	0,38	0,55
Peter Dourmashkin	1,0	0,78	0,88
Ortalama	0,97	0,5	0,66

İsimlere göre gruplandırma başarıları incelendiğinde, Çizelge 3.12'ye göre, gruplandırılmada kullanılan sayfaların %97'si doğru olarak gruplandırılmıştır. Buna karşılık tüm dokümanların %50'si gruplandırılabilmiştir. F1 katsayısına göre gruplandırma başarısı %66 olarak belirlenmiştir.

Çizelge 3.13. IF NER farklı meslek test verisi ile hasatlama başarısı

Kişi No	Kişi Adı	Hasatlanan Meslek	Hasatlanan Doğum Tarihi	Hasatlanan Resim
1	Başarılı	Başarılı	Başarılı	Başarılı
2	Başarılı	Başarılı	Başarılı	Başarılı
3	Başarılı	Başarısız	Başarılı	Başarılı
4	Başarılı	Başarılı	-	-
5	Başarılı	Başarılı	Başarılı	Başarısız
6	Başarılı	Başarılı	Başarısız	Başarılı
7	Başarılı	Başarılı	Başarılı	Başarılı
8	Başarılı	Başarılı	Başarılı	Başarılı
9	Başarısız	Başarısız	-	Başarısız
10	Başarısız	Başarısız	-	Başarısız
11	Başarısız	Başarısız	-	Başarısız
12	Başarısız	Başarısız	-	-
13	Başarılı	Başarılı	-	Başarılı
Hasatlama Başarısı (%)	69,2	61,5	85,7	63,6

Çizelge 3.13'teki farklı meslek test verisi hasatlama sonucu incelendiğinde kişilerin %69,2'si hasatlama çalışmasında tespit edilebilmiştir. Meslek bilgisinde %61,5 oranında başarı elde edilmiştir. Doğum tarihinde bu oran %85,7 olmuştur. Resim bilgisinde ise %63,6 başarı elde edilmiştir.

Öğretim üyesi test verisi

Öğretim üyesi test verisi, Gazi Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim üyelerinin isimlerinden ve bu isimlerden elde edilen ilk 30 arama motoru sonuçlarından oluşmaktadır. Bu test verisi içerisindeki kişilere ait bilgiler bir yüksek lisans öğrencisi ve bir yüksek lisansını tamamlamış öğrenci tarafından Çizelge 3.14'teki gibi tespit edilmiştir. Öğretim üyesi test verisinin oluşturulma amacı Gazi Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim üyelerinin Web üzerindeki tanınırlıklarının ölçülmesidir.

Çizelge 3.14. Öğretim üyesi test verisi içeriği

Kişi No	Kişi Adı	Meslek	Doğum Tarihi	Resim
1	Şeref Sağıroğlu	Professor	August 1 1965	Var
2	M. Ali Akcayol	Professor	-	Var
3	Hasan Şakir Bilge	-	-	-
4	Suat Özdemir	Assoc. Prof.	-	Var
5	Hacer Karacan	Assist. Prof. Dr.	-	Var
6	Mehmet Şimşek	Minister of finance	1967	Var
7	Mehmet Şimşek	Kuaför	-	-
8	Murat Hacıömeroğlu	PhD	-	Var
9	Oktay Yıldız	Project Leader	-	Var
10	Oktay Yıldız	Senior Test Leader	-	Var
11	Oktay Yıldız	SDET Leader	-	Var

Yapılan deneyde “Meslek”, “Doğum Tarihi”, “Sayfa Sayısı” bilgileri test verisini hazırlayan kişiler tarafından göz ile tespit edilmiştir. “IF meslek”, “IF Doğum Tarihi”, “Resim”, “IF Sayfa Sayısı” bilgileri ise IF NER tarafından otomatik olarak tespit edilmiştir.

Çizelge 3.15. Öğretim üyesi test verisi: Şeref Sağıroğlu

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
Professor	August 1 1965	25	Prof.	August 1		14	14

Şeref Sağıroğlu veri grubu IF ara yüzünden “Seref Sagiroglu” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 5 tanesinde sayfaların kime ait olduğu belli değildir. 14 sayfa ise hasatlama için gruplandırılmıştır. Geri kalan 11 sayfa IF tarafından yetersiz bilgi içerdiğinden elenmiştir. Bu veride sadece 1 kişiye ait bilgiler bulunmaktadır. Bu veride sadece 1 kişiye ait sayfalar bulunmaktadır.

Çizelge 3.16. Öğretim üyesi test verisi: M. Ali Akcayol

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
Professor	-	15	model	-		4	4

M. Ali Akcayol veri grubu IF ara yüzünden “M. Ali Akcayol” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 15 tanesinde sayfaların kime ait olduğu belli değildir. 4 sayfa ise hasatlama için gruplandırılmıştır. Geri kalan 11 sayfa IF tarafından yetersiz bilgi içerdiğinden elenmiştir. Bu veride sadece 1 kişiye ait bilgiler bulunmaktadır. Bu veride sadece 1 kişiye ait sayfalar bulunmaktadır.

Çizelge 3.17. Öğretim üyesi test verisi: Hasan Şakir Bilge

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
-	-	0	-	-	-	0	0

Hasan Şakir Bilge veri grubu IF ara yüzünden “Hasan Sakir Bilge” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 26 tanesinde sayfaların kime ait olduğu belli değildir ya da farklı isime sahip kişilere aittir. 4 sayfa ise bu isme sahip bir kişiye aittir.

Çizelge 3.18. Öğretim üyesi test verisi: Suat Özdemir

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
Assoc. Prof.	-	8	Prof.	-		5	5

Suat Özdemir veri grubu IF ara yüzünden “Suat Ozdemir” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 3 tanesinde sayfaların kime ait olduğu belli değildir. 5 sayfa ise hasatlama için gruplandırılmıştır. Geri kalan 22 sayfa IF tarafından yetersiz bilgi içerdiğinden elenmiştir. Bu veride sadece 9 kişiye ait bilgiler bulunmaktadır fakat bu kişilere ait yeterli bilgi bulunmadığından gruplandırma yapılamamıştır.

Çizelge 3.19. Öğretim üyesi test verisi: Hacer Karacan

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
ASSIST. PROF. DR.	-	10	-	-		1	1

Hacer Karacan veri grubu IF ara yüzünden “Hacer Karacan” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 20 tanesinde sayfaların kime ait olduğu belli değildir. 1 sayfa ise hasatlama için gruplandırılmıştır. Geri kalan 9 sayfa IF tarafından yetersiz bilgi içerdiğinden elenmiştir. Bu veride sadece 1 kişiye ait bilgiler bulunmaktadır.

Çizelge 3.20. Öğretim üyesi test verisi: Mehmet Şimşek

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
minister of finance	1967	24	minister of finance	1967		12	12
Kuaför	-	1	-	-	-	0	0

Mehmet Şimşek veri grubu IF ara yüzünden “Mehmet Simsek” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 5 tanesinde sayfaların kime ait olduğu belli değildir. 12 sayfa ise hasatlama için gruplandırılmıştır. Geri kalan 13 sayfa IF tarafından yetersiz bilgi içerdiğinden elenmiştir. Bu veride sadece 2 kişiye ait bilgiler bulunmaktadır fakat bu kişilerden sadece 1 tanesi hakkında yeterli bilgi vardır.

Çizelge 3.21. Öğretim üyesi test verisi: Murat Hacıömeroğlu

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
PhD	-	17	-	-		1	1

Murat Hacıömeroğlu veri grubu IF ara yüzünden “Murat Hacıomeroglu” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 13 tanesinde sayfaların kime ait olduğu belli değildir. 1 sayfa ise hasatlama için gruplandırılmıştır. Geri kalan 16 sayfa IF tarafından yetersiz bilgi içerdiğinden elenmiştir. Bu veride sadece 1 kişiye ait bilgiler bulunmaktadır.

Çizelge 3.22. Öğretim üyesi test verisi: Oktay Yıldız

Meslek	Doğum Tarihi	Sayfa Sayısı	IF Meslek	IF Doğum Tarihi	Resim	IF Sayfa Sayısı	IF Doğru Sayfa Sayısı
Project Leader	-	3	Leader	-		2	2
Senior Test Leader	-	1	-	-	-	0	0
SDET Leader	-	1	-	-	-	0	0

Oktay Yıldız veri grubu IF ara yüzünden “Oktay Yıldız” sorgusu ile arama yapılarak elde edilen ilk 30 sayfadır. Bu 30 sayfadan 25 tanesinde sayfaların kime ait olduğu belli değildir. 2 sayfa ise hasatlama için gruplandırılmıştır.

Çizelge 3.23. IF NER öğretim üyesi test verisi için gruplandırma başarısı

Veriler / Katsayı	P	R	F1
Şeref Sağıroğlu	1,0	0,56	0,72
M. Ali Akcayol	1,0	0,26	0,41
Hasan Şakir Bilge	0,0	0,0	0,0
Suat Özdemir	1,0	0,62	0,76
Hacer Karacan	1,0	0,1	0,18
Mehmet Şimşek	1,0	0,48	0,64
Murat Hacıömeroğlu	1,0	0,06	0,11
Oktay Yıldız	1,0	0,4	0,57
Ortalama	0,87	0,31	0,42

İsimlere göre gruplandırma başarıları incelendiğinde, Çizelge 3.23'e göre, gruplandırılmada kullanılan sayfaların %87'si doğru olarak gruplandırılmıştır. Buna karşılık tüm dokümanların %31'i gruplandırılabilmiştir. F1 katsayısına göre gruplandırma başarısı %42 olarak belirlenmiştir.

Çizelge 3.24'teki öğretim üyesi test verisi hasatlama sonucu incelendiğinde kişilerin %63,6'sı hasatlama çalışmasında tespit edilebilmiştir. Meslek bilgisinde %30 oranında başarı elde edilmiştir. Sadece 2 kişiye ait doğum tarihi bilgisi bulunmaktadır ve bu iki tarih de doğru olarak tespit edilmiştir. Bu nedenle doğum tarihinde bu oran %100 olmuştur. Resim bilgisinde ise %77,7 başarı elde edilmiştir.

Çizelge 3.24. IF NER öğretim üyesi test verisi hasatlama başarısı

Kişi No	Kişi Adı	Meslek	Doğum Tarihi	Resim
1	Başarılı	Başarılı	Başarılı	Başarılı
2	Başarılı	Başarısız	-	Başarılı
3	Başarısız	-	-	-
4	Başarılı	Başarısız	-	Başarılı
5	Başarılı	Başarısız	-	Başarılı
6	Başarılı	Başarılı	Başarılı	Başarılı
7	Başarısız	Başarısız	-	-
8	Başarılı	Başarısız	-	Başarılı
9	Başarılı	Başarılı	-	Başarılı
10	Başarısız	Başarısız	-	Başarısız
11	Başarısız	Başarısız	-	Başarısız
Hasatlama Başarısı (%)	63,6	30	100	77,7

Öğretim üyesi test verisi Türkçe

IF NER, Türkçe dokümanlar üzerinde çalışacak şekilde geliştirilmemiştir. Öğretim üyesi test verisi Türkçe olarak sorgulanarak tekrar hazırlanıp, Türkçe dokümanlar üzerindeki davranışı da incelenmiştir.

Çizelge 3.25. IF NER Türkçe öğretim üyesi test verisi üzerinde gruplandırma başarısı

Veriler/ Katsayı	P	R	F1
Şeref Sağiroğlu	1,0	0,1	0,18
M. Ali Akcayol	0,8	0,26	0,39
Hasan Şakir Bilge	0,0	0,0	0,0
Suat Özdemir	1,0	0,31	0,47
Hacer Karacan	0,9	0,47	0,61
Mehmet Şimşek	0,5	0,06	0,11
Murat Hacıömeroğlu	0,0	0,0	0,0
Oktay Yıldız	1,0	0,3	0,46
Ortalama	0,65	0,18	0,27

Çizelge 3.25'e bakıldığında Türkçe sayfalar üzerinde hasatlama yapıldığında gruplandırma performansının P, R ve F1 katsayılarını değerlendirerek değerlerinin düştüğü görülmüştür.

IAF test verisi

Kushmerick'in çalışmasındaki IAF test verisi, bu çalışmada da kullanılmıştır [37]. IAF test verisinde isim, e-posta, diğer ad, son güncelleme zamanı, organizasyon ve servis sağlayıcı bilgilerinin olduğu kişilere ait bilgiler bulunmaktadır [37]. Bu test verisi aynı zamanda başka çalışmalarda da kullanılmıştır [74]. IAF test verisi 11 Web sayfasından oluşmaktadır. Her sayfada farklı sayıda kişilere ait bilgiler bulunmaktadır. Test verisinin detayına EK-2'den ulaşılabilir.

IAF test verisi kullanılarak yapılan hasatlama çalışmasında, tüm bilgiler başarı ile hasatlanmıştır. IF NER aynı zamanda birden fazla olan organizasyon bilgilerinden bir tanesini seçmiştir.

Çizelge 3.26. IF NER IAF test verisi hasatlama performansı

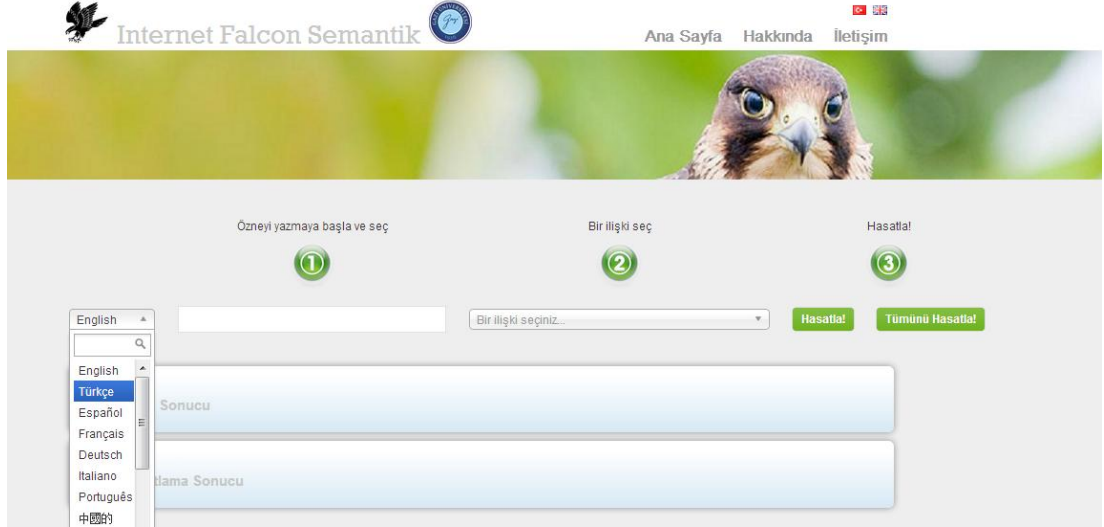
	P	R	F1
1. veri	1,0	1,0	1,0
2. veri	1,0	1,0	1,0
3. veri	1,0	1,0	1,0
4. veri	1,0	1,0	1,0
5. veri	1,0	1,0	1,0
6. veri	1,0	1,0	1,0
7. veri	1,0	1,0	1,0
8. veri	1,0	1,0	1,0
9. veri	1,0	1,0	1,0
10. veri	1,0	1,0	1,0
Ortalama	1,0	1,0	1,0

3.2. IFS

Bu tez çalışmasında düz metni işleyerek web sayfaları üzerinden bilgi çıkarmanın zorlukları tespit edilmiş ve web üzerinden bilgi çıkarma yöntemlerine alternatif olarak IFS adında ikinci bir uygulama geliştirilmiştir. Bu uygulama semantik bir veritabanı üzerinden sorgulamalar yaparak bilgi hasatlayan web tabanlı çevrimiçi bir sistemdir. <http://www.internetfalcon.com> adresinden erişmek mümkündür. Sağladığı basit bir ara yüz sayesinde Freebase veritabanındaki varlıklar hakkında 18 farklı dilde kısa açıklama ve varlığa ait semantik bilgiler hasatlanabilmektedir [49].

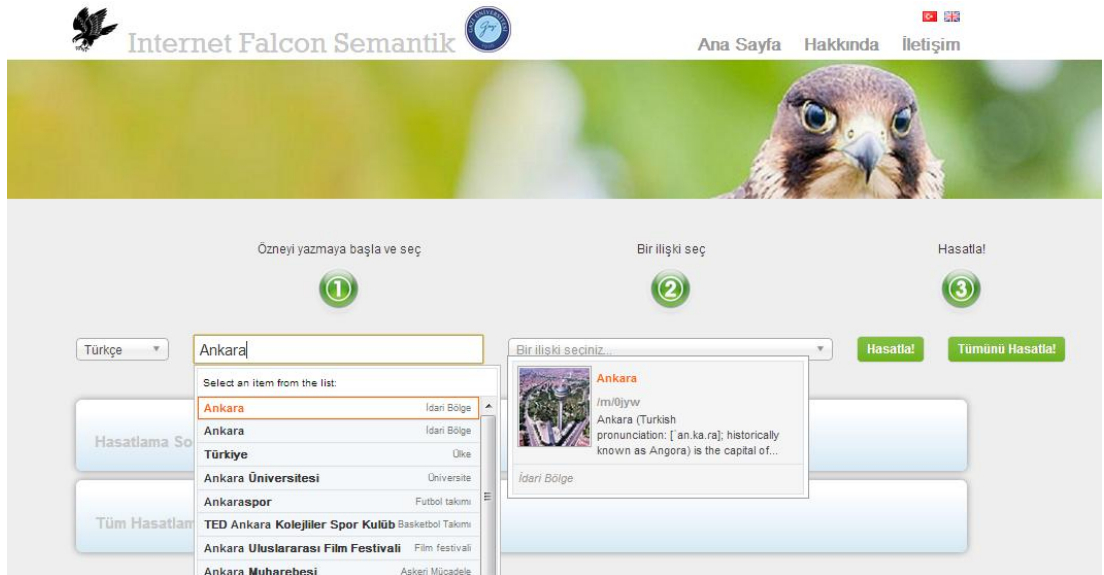
Şekil 3.8. IFS ara yüzü

IFS, 18 adet dili desteklemektedir. Varsayılan dil İngilizcedir fakat bunun dışında diğer dillerden herhangi birisi de seçilebilir.



Şekil 3.9. IFS için dil desteği

Daha sonra hasatlanacak öznenin (varlığın) adı yazılır. Aynı zamanda yazılan kelimeyi içeren varlıklar listelenir. Listelenen varlıkların üzerine gelinerek varlıklar hakkında kısa bilgi elde edilebilir.



Şekil 3.10. IFS hakkında bilgi hasatlanacak varlığın seçimi

İstenilen varlık seçildikten sonra, seçilen varlığa ait resim ve kısa açıklama bilgisi görülür. Bu bilgiler Topic API kullanılarak hasatlanmaktadır. Seçilen dile göre bu bilgiler değişmektedir.

Şekil 3.11. IFS varlık hakkında kısa bilgi ve resim bilgisi hasatlama

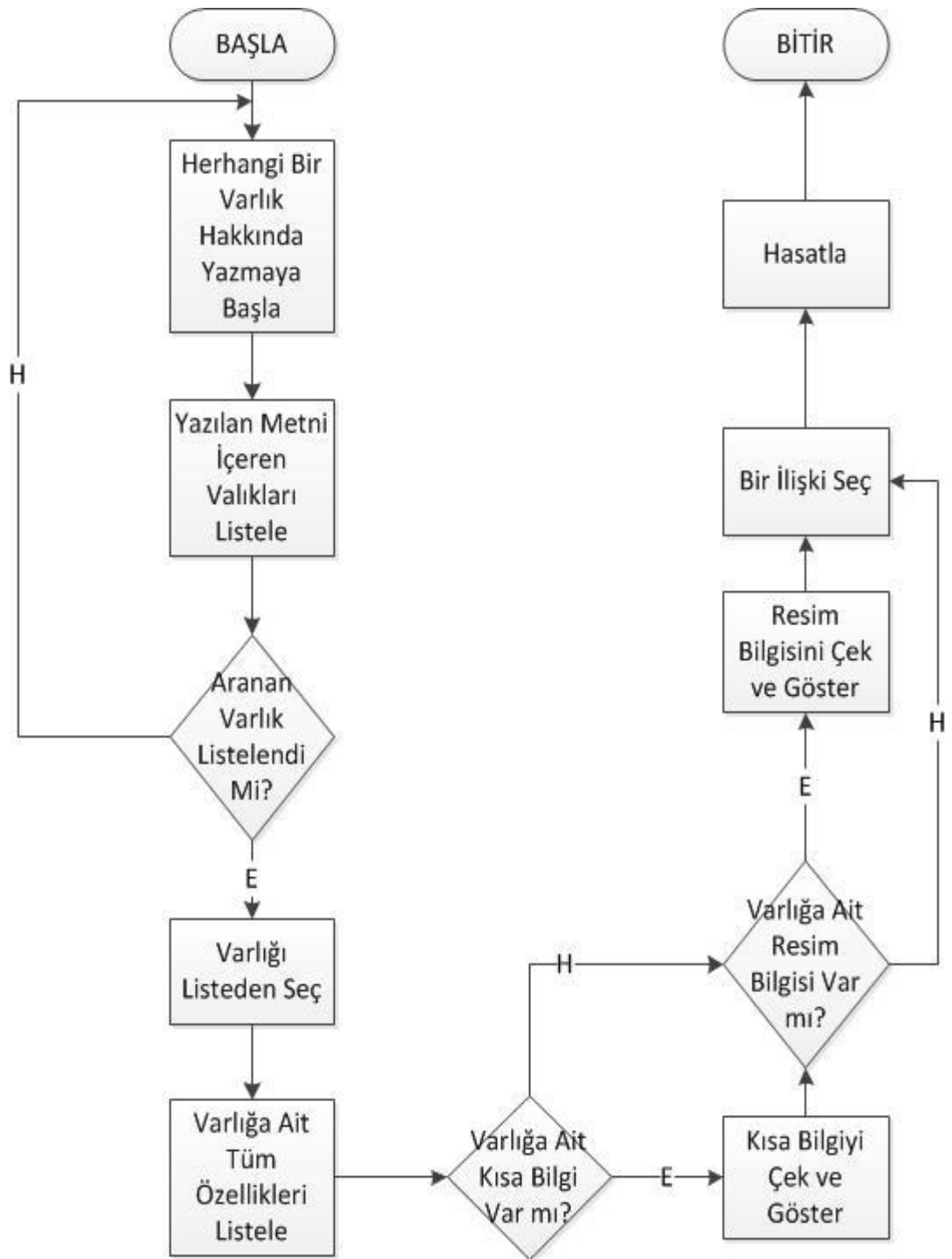
Son olarak tek tek ya da tüm ilişkiler hasatlanabilmektedir. Örneğin kullanıcı istediği ilişkiyi seçip hasatlama işlemini gerçekleştirebilir ya da tümünü tek seferde hasatlayabilir. Bu bilgiler de varsa seçilen dile göre getirilmektedir fakat seçilen dilde o bilgiler yoksa varsayılan dil olan İngilizce sonuçlar gösterilmektedir.

Özne	İlişki	Nesne
Ankara	id	/en/ankara
Ankara	alias	Ancyra,Ankara, Turkey,Capital of Turkey
Ankara	name	Ankara
Ankara	containedby	Turkey,Ankara Province,Western Asia
Ankara	nearby_airports	Esenboğa International Airport,Guvercinlik Airport,Etimesgut Airport,Akinci Airport,Akinci Air Base

Şekil 3.12. IFS bilgi hasatlama

3.2.1. Hasatlama süreci

IFS bu bölümde hasatlama yaparken kullanıcıdan minimum bir arayüz üzerinden bir işlem yapmasını beklemektedir. Kullanıcı hakkında hasatlama yapacağı varlığın adını yazmaya başladığında içerisinde yazılan verinin geçtiği varlıklar listelenmektedir. Kullanıcı bu sayede daha hangi varlıktan bahsettiğine varlıkların yanındaki kısa bilgilere bakarak karar verebilmekte ve aralarından istediği varlığı seçebilmektedir. Bu işlemden sonra IF, otomatik olarak ürettiği sorgu ile Topic API'ı kullanarak varlığa ait kısa bilgiyi sunmaktadır. Bu bilgi seçilen dile göre (varsayılan olarak İngilizce) kullanıcıya sunulmaktadır. Bunun dışında kullanıcı bir semantik ilişkiyi ifade eden “özne (subject) – ilişki (predicate) – nesne (object)” üçlüsü şeklinde hasatlama yapabilmektedir. Burada özne, hakkında hasatlama yapmak istenilen varlığın kendisidir. Nesne hasatlanacak bilgidir. İlişki ise özne ve nesne arasındaki ilişkiyi ifade eder. Örneğin, “Ankara – Turistik Yerler – Atakule, Anıtkabir” şeklindeki bir üçlü Ankara’daki turistik yerleri ifade etmektedir. Kullanıcı bu şekildeki üçlülere tek tek ya da hepsini birden hasatlayabilmektedir. Bunun için varlığa ait yüklemelerden birisi seçilebilir ya da var olan tüm yüklemeler için bir MQL sorgusu oluşturulur ve Freebase’den sorgulanır. Dönen cevap JQuery ve JSON kullanılarak işlenir ve kullanıcıya rapor olarak sunulur.



Şekil 3.13. IFS akış diyagramı

3.2.2. Mimarisi

IFS Web tabanlı bir uygulamadır ve bir tarayıcı yardımı ile çalışır. 4 ana modülden oluşmaktadır. Uygulamaya bir sunucu üzerinden çevrimiçi olarak erişmek mümkündür. IFS, 4 modülden oluşmaktadır.

Web modülü

Web modülü, kullanıcıya hasatlama işlemi için bir ara yüz sunar. Bu ara yüz ile kullanıcıdan hasatlama yapmak istediği varlık ve diğer bilgiler alınır. Hasatlamada kullanılacak 18 dilden bir tanesi seçilebilir. Eğer seçilen dile ait bilgi bulunamazsa o dilde bilginin bulunmadığı belirtilir ve varsayılan dil olan İngilizce sonucu gösterilir. Web modülü ayrıca raporlama modülünün ürettiği çıktıları kullanıcıya sunmaktadır.

Sorgulama

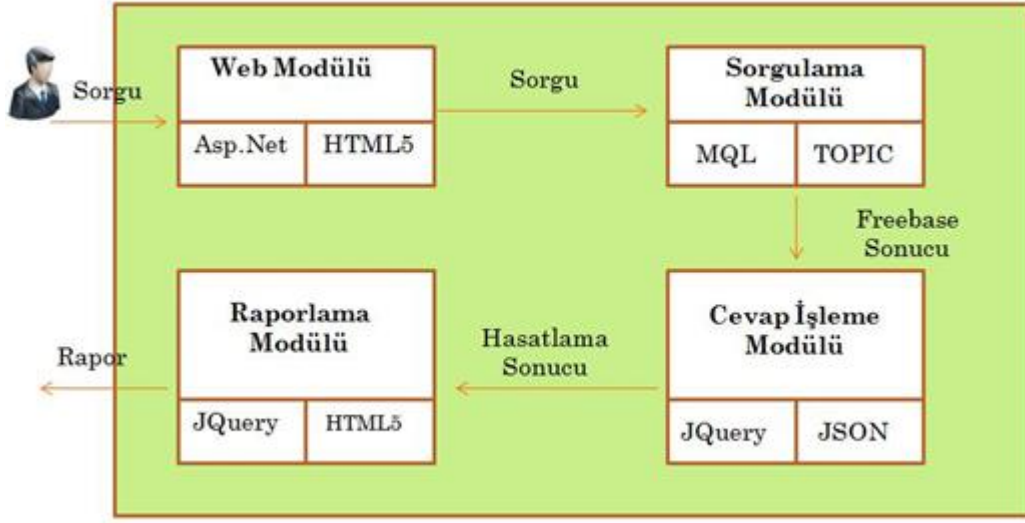
Sorgulama modülünde otomatik MQL (Web ara veri sorgulama dili) sorguları oluşturulmakta ve bu sorgular Freebase veritabanına gönderilmektedir.

Cevap işleme

Freebase veritabanının sorgu sonucu JSON (JavaScript nesne gösterim biçimi) olarak alınmakta ve bu cevap JQuery ile işlenmektedir.

Raporlama

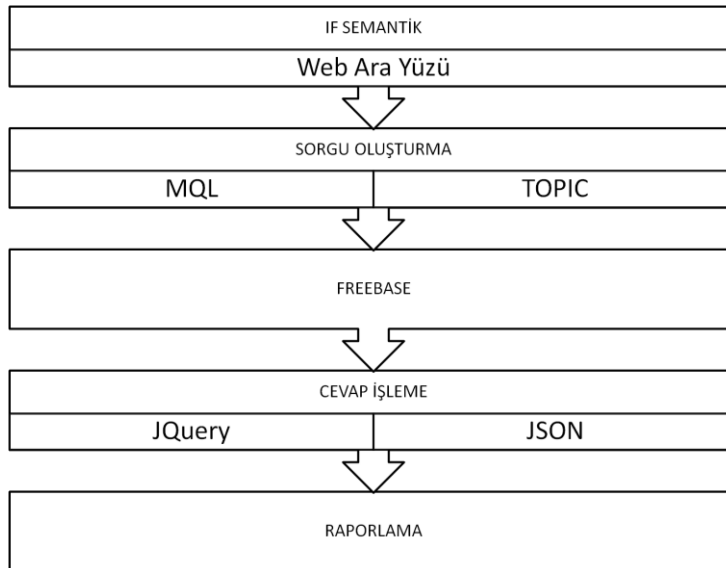
JSON formatında alınan sonuç işlendikten sonra HTML formatına çevrilerek kullanıcıya sunulmaktadır.



Şekil 3.14. IFS mimarisi

3.2.3. Kullanılan teknolojiler

IFS kapsamında bazı Web teknolojileri kullanılmıştır. Seçilen teknolojilerin kullanılan servislere uygunluğu esas alınmıştır.



Şekil 3.15. IFS hasatlama süreci ve kullanılan teknolojiler

Asp.Net MVC 4

Asp.Net MVC (model görünüm ve kontrol), MVC tasarım örüntüsüne uygun Web uygulamaları geliştirilmesine olanak veren bir uygulama geliştirme çatısıdır [75]. Bu teknoloji ile geliştirilen uygulamalar birçok servis sağlayıcının sunucularına düşük maliyetler ile yüklenebilmektedir. Bu çatının HTML5 (5. HTML sürümü) desteği de bulunmaktadır. Bu sayede geliştirilen Web uygulaması mobil cihazlarda da sorunsuz olarak çalışmaktadır.

Bu çatının, uygulamanın çalışma mantığı ya da kullanılan yöntemler ile doğrudan ilişkisi yoktur fakat kullanılan yöntem ve diğer teknolojilerin rahatça kullanılabileceği bir ortam sağlamaktadır.

AJAX

AJAX (Asenkron JavaScript ve XML), isminde XML geçse bile XML yerine JSON da kullanılabilir. AJAX ile istemci tarafından veri gönderilerek sunucudan veri istenebilir. IF kapsamında oluşturulan MQL sorguları AJAX istekleri ile Freebase sunucularına gönderilmekte ve bu sorgular çalıştırıldıktan sonra sorgu sonucu olarak geri dönülmektedir.

JSON

JSON, veri alış-verişinde kullanılan metin tabanlı nesne gösterim biçimidir. IF kapsamında Freebase sunucuları ile iletişimde veri transferi yaparken JSON formatı kullanılmıştır. Bu sayede hem gönderilen isteğin oluşturulması, hem de geri gelen cevabın işlenmesi basit ve belirli bir standarda uygun olarak yapılmıştır.

JQuery

JQuery, farklı tarayıcılarda çalışabilen bir JavaScript kütüphanesidir. JQuery ile AJAX istekleri, HTML değişiklikleri, JSON işleme gibi işlemler basit ve hızlı bir

şekilde yapılabilmektedir. IFS kapsamında bu avantajlarından dolayı JQuery kütüphanesi kullanılmıştır.

MQL API

MQL, Freebase'e programatik sorgular yapma amacıyla kullanılan bir sorgu kütüphanesidir. HTTP isteklerinde standart olarak JSON kullanmaktadır. IFS kullanıcıdan aldığı parametrelere göre MQL sorguları üretmekte ve JSON formatında dönen cevapları işlemektedir. Örneğin bir varlığa ait tüm nitelikleri sorgulamak için aşağıdaki sorgu kullanılabilir.

Çizelge 3.27. Bir varlığa ait tüm niteliklerin elde edileceği MQL sorgusu

```
[{
  "id": "/en/ankara",
  "type": []
}]
```

Topic API

Topic API, bir konu hakkında kısa açıklama ve resimler gibi bilgileri de içeren bir veritabanına erişmek mümkündür. IF kapsamında kısa açıklama ve resim bilgisi Topic API kullanılarak elde edilmiştir.

3.2.4. Kullanılan yöntemler

IFS kapsamında otomatik MQL sorguları üretilmekte, resim ve açıklama bilgileri için Topic API istekleri oluşturulmakta ve Freebase Suggest, varlıkların sorgulanıp filtrelenebildiği bileşen, bileşeni seçilen dile göre ayarlanabilmektedir.

Önceden hazırlanmış ve yapılandırılabilen şablon MQL sorguları kullanılarak anlık sorgular üretilebilmektedir. Aşağıda hazırlanan MQL sorgu şablonları vardır.

Çizelge 3.28. IFS’de kullanılan MQL sorgu şablonları

Açıklama	MQL Sorgu Şablonu
Seçilen varlığa ait tüm tip bilgileri	[{ "id": subjectId, "type": [] }]
Seçilen varlığa ait tespit edilen tipteki tüm nitelikler	[{ "id": subjectId, "type": type, "*": null }]
Seçilen varlığa ait seçilen yüklem için seçilen dildeki “nesne”(hasatlanacak bilgi) değeri	[{ "id": subjectId, "type": typeValue, "predicateValue": [{"lang": "/lang/" + lang, "value": null}] }]

Seçilen varlığa ve dile göre özet bilginin alınabilmesi için belirli bir formatta bir Http isteği yapılmaktadır. Dönen cevap yine JSON formatında Topic bilgisidir.

Çizelge 3.29. Topic API ile özet bilginin sorgulanması

<a "?filter='/common/topic/description&lang="' +="" href="https://www.googleapis.com/freebase/v1/topic" id="" lang"="">https://www.googleapis.com/freebase/v1/topic" + id + "?filter=/common/topic/description&lang=" + lang
--

Burada “id”, seçilen varlığa ait id bilgisidir. “lang” ise özet bilginin hangi dilde sorgulanacağı bilgisidir. Freebase 18 dil desteklemektedir [49].

3.2.5. Yapılan testler

IFS çalışması, semantik veritabanından hasatlama işlemi yaptığından daha tutarlı sonuçlar vermektedir. Her varlık için farklı bilgiler elde edilebilmektedir.

Yapılan deneylerde bir varlık aranmış daha sonra o varlık hakkındaki bilgiler

semantik ilişki olarak çıkarılmıştır. Burada “Özne” hakkında arama yaptığımız varlıktır. “Nesne”, hasatladığımız bilgidir. “Nesne”, özne ile nesne arasındaki ilişkiyi ifade eden bilgidir. “Nesne (Çeviri)” ise hasatlanan bilginin Türkçe karşılığıdır.

IF NER’de kullanılan farklı meslek test verisindeki kişi isimleri bu bölümde de kullanılmıştır. Sonucu verilmemiş kayıtlar için herhangi bir sonuç bulunamamıştır.

Çizelge 3.30. IFS Antonio Albanese verisi

Özne	İlişki	Nesne	Nesne (Çeviri)
Antonio Albanese	Doğum tarihi	1964-10-10	1964-10-10
	Cinsiyet	Male	Erkek
	Id	/en/antonio_albanese	/en/antonio_albanese
	İsim	Antonio Albanese	Antonio Albanese
	Doğum yeri	Olginate	Olginate
	Meslek	Actor,Comedian,Film Director,Screenwriter	Aktör, Komedyen, Film Yönetmeni, Senarist
	Resim	-	-

Çizelge 3.31. IFS Jessica Fox verisi

Özne	İlişki	Nesne	Nesne (Çeviri)
Jessica Fox	Doğum tarihi	1983-05-19	1983-05-19
	Cinsiyet	Female	Kadın
	Id	/en/jessica_fox	/en/jessica_fox
	İsim	Jessica Fox	Jessica Fox
	Doğum yeri	Englang	İngiltere
	Meslek	Actor	Aktör
	Resim	-	-
Jessica Fox	Doğum tarihi	1994-06-11	1994-06-11
	Cinsiyet	Female	Kadın
	Id	/m/0dd8wnb	/m/0dd8wnb
	İsim	Jessica Fox	Jessica Fox
	Doğum yeri	Marseille	Marsilya
	Meslek	Olympic Athlete	Olimpik Atlet
	Resim	-	-
Jessica Fox	Cinsiyet	Female	Kadın
	Id	/m/0n9w6jg	/m/0n9w6jg
	İsim	Jessica Fox	Jessica Fox
	Resim	-	-

3.3. IF NER ve IFS Karşılaştırma

IF iki farklı uygulama olarak geliştirilmiştir. Uygulamalarda farklı kütüphaneler ve teknolojiler kullanılmıştır. Kullanılan yöntemlere göre bu tercihler belirlenmiştir. Her iki uygulamanın da avantajları ve dezavantajları bulunmaktadır.

3.3.1. IF NER genel değerlendirme

IF NER, Web üzerindeki kişisel bilgileri hasatlamaktadır. Birçok farklı yöntem ve teknoloji bir arada kullanılmaktadır. Bunun en büyük dezavantajı olarak performans sorunları ile karşılaşmıştır. Web verisi üzerinden hasatlama işlemi yapısı itibari ile çok karmaşık bir işlemdir. Bu nedenle performans sorunları olması çok muhtemeldir.

Bu uygulamada sadece İngilizce dokümanlar üzerinde işlem yapılmıştır. Uygulama diğer dilleri desteklememektedir. Bunun nedeni kullanılan araçların tamamen İngilizce cümleleri inceleme ve işlemeye yönelik olmasıdır. Diller birbirinden morfolojik olarak çok farklıdır. Bir dile ait bir cümleyi analiz etmek için o dile özel olarak geliştirilmiş araçlar kullanılmalıdır. Bu uygulamada temel bilgi çıkarma tekniği cümle analizi olduğundan sadece İngilizce dokümanlar işlenebilmektedir. Bu yöntem ile örneğin Türkçe dokümanlar işlenmeye çalışıldığında beklenmedik sonuçlar ile karşılaşılabilir. Çünkü İngilizce ile Türkçe cümle yapıları birbirinden farklıdır. Cümle analizinde kullanılan araçlar hatalı sonuçlar üretebilir.

IF NER, sadece belirli bir veri seti üzerinde değil, Web üzerindeki İngilizce dokümanların tümü için çalışabilmektedir. Her sayfa için bir sonuç üretmeyebilir fakat herhangi bir sayfayı işleyebilmektedir. Web verisi sürekli gelişmektedir. Bu nedenle yeni oluşturulan sayfaların da işlenebilmesi önem kazanmaktadır. Bu bölüm sürekli gelişen veriye uygunluk göstermekte ve arama motoru aracılığı ile bu sayfalara erişebilmekte ve bunları işleyebilmektedir.

Kullanılan yöntemler açısından IF NER, WebHawk çalışması ile benzerlik göstermektedir [5]. Her iki çalışmada da aglomeratif gruplama algoritması

kullanılmıştır. Sonuçta adalara ait sayfalar elde edilmektedir. Her iki çalışmada da hasatlama yöntemi olarak cümle analizi kullanılmıştır. IF NER, bunun üstüne farklı birçok hasatlama tekniğini bir arada kullanmıştır.

3.3.2. IFS genel değerlendirme

IFS, IF NER'den farklı yöntem ve teknikleri kullanmaktadır. Karşılaşılan performans sorunları, elde edilen bilgilerin kapsamı açısından farklı bir çözüm getirmektedir. Web üzerinden bilgi hasatlama işlemi temelde semantik ilişkileri çıkarmaya dayanmaktadır. Web verisi üzerinden bu semantik ilişkileri çıkarmak performans gerektiren bir işlem olduğundan, bu duruma çözüm olarak Web verisini semantik bir yapıda tutup daha sonra bunun üzerinden hasatlama yapmak bilgi çıkarma hızını ve çıkarılan bilginin kalitesini artırabilir. Bu nedenle en yaygın ve ücretsiz semantik veritabanlarından biri olan Freebase kullanılarak ikinci bölüm uygulaması geliştirilmiştir.

IFS, birçok konuya çözüm getirebilmiştir fakat bununla beraber bazı dezavantajları da bulunmaktadır. Bu uygulamanın en büyük problemi hasatlama işlemini sadece Freebase veritabanında bulunan bilgiler dâhilinde yapılabilmesidir. Buna karşı sağladığı kazanımlar oldukça fazladır. Bilgiler semantik bir yapıda tutulduğundan düz metin işlemek gibi bir durum söz konusu değildir. Bu durum da sonuçların çok hızlı elde edilmesini sağlamaktadır. IF NER'de yaşanan performans sorununa çözüm getirmektedir. IF NER'de bir kişiye ait bilgilerin hasatlanması işi 10-15 dakika arasında sürmektedir. IFS'de ise 1-2 saniyeler mertebesinde sonuçları elde etmek mümkündür. Bu kazanım tamamen verilerin nasıl tutulduğu ile ilgilidir. Düz metin bilgisini işlemek maliyetli bir işlemdir.

Freebase veritabanında her varlığa ait farklı ilişkiler tutulabilmektedir. Örneğin bir öğretim görevlisinin uzmanlık alanı, futbolcunun oynadığı takımlar, bir şehrin turistik yerleri gibi her varlığa özel bilgiler bulunmaktadır. IFS ile Freebase veritabanında bulunan herhangi bir varlığa ait tüm semantik bilgileri, resim bilgisini ve kısa açıklamayı hasatlamak mümkündür. Bu sayede milyonlarca varlık hakkında

saniyeler içerisinde bilgileri hasatlamak mümkün olmaktadır.

IFS'nin sağladığı bir diğer avantaj ise aralarında Türkçenin de bulunduğu 18 dili desteklemesidir. Herhangi bir varlığa ait istenilen dilde bilgileri hasatlamak mümkündür fakat her dilde aynı miktarda bilgi yoktur. Örneğin İngilizce olarak olan bir bilgi Türkçe olarak olmayabilir. Bu nedenler her dil birbiri ile eşdeğer değildir. Eğer bir bilginin istenilen dilde karşılığı yoksa uygulama varsayılan dil olan İngilizce karşılığını göstermektedir.

IFS'yi diğer çalışmalardan ayıran en büyük özelliği 18 dilde hasatlama yapabilme yeteneğidir. Diğer çalışmaların hiçbiri bu özelliğe sahip değildir [5,14,42]. Günümüzde semantik bilgi hasatlama yöntemleri popülerlik kazanmaktadır. Bunların örneklerine geleneksel yöntemler ile hasatlama yapanlar kadar fazla olmasa da erişebilmekteyiz [38].

3.3.3. Değerlendirme

Bu çalışma kapsamında geliştirilen iki uygulama karşılaştırıldığında her iki uygulamanın da üstünlükleri ve kısıtları bulunmaktadır. IF NER, Web üzerinde kişisel bilgileri hasatlamak için daha başarılıdır. IFS ise gerçek dünya varlıkları hakkında hasatlama yapılmak istendiğinde başarılı bir uygulamadır.

Çizelge 3.32. IF NER – Semantik karşılaştırma

Özellik	NER	Semantik
İngilizce hasatlama	Var	Var
Farklı dil ile hasatlama	Yok	Var
Hasatlama süresi	10-15 dk.	1-2 sn.
Adaşlara göre gruplandırma	Var	Yok
Kişisel bilgi hasatlama	Var	Var
Genel bilgi hasatlama	Yok	Var
Doküman üzerinden hasatlama yapma	Var	Yok
Hasatlama tekniği	Cümle analizi, XSLT, düzenli ifadeler, genişletilmiş arama	MQL API, TOPIC API

Çizelge 3.32’de görüldüğü gibi IF NER ve IFS 8 özellik açısından karşılaştırılmıştır. IF NER daha geniş veri setleri üzerinde çalışma imkanı sunarken, performans sorunlarını beraberinde getirmektedir. Buna karşılık IFS ise farklı dil desteği, yüksek performans ve farklı konular üzerinde genel bir bilgi hasatlama imkanı sunmaktadır fakat sadece veritabanı içeriği kadar bilgi sunabilmektedir. IF NER farklı teknikleri bir arada kullanarak metinleri işlerken, arama motoru sorguları üretirken, IFS ise kendi veritabanından sorgulamalar yapmaktadır.

IF NER, diğer uygulamalar ile karşılaştırıldığında hem adalara göre sonuçları gruplandırabilmekte hem de hasatlama işlemini gerçekleştirmektedir [14,36,40]. Bu açıdan diğer uygulamalardan ayrılmaktadır. WebHawk ile ise benzer özelliklere sahiplerdir [5].

IFS ise mevcut bilgi çıkarma çalışmalarının karşılaştığı sorunların bir kısmına çözüm getirmektedir. Örneğin web verisindeki zorluklar üzerinde çalışmamaktadır [2]. Bunun yerine semantik ilişkilerin kurulduğu bir veritabanı kullanmaktadır.

4. SONUÇLAR VE TARTIŞMA

Bilgi hasatlama üzerine çokça araştırma yapılmıştır. Fakat araştırmaların çoğu belirli dilbilgisi kurallarına uyan ya da belirli formatlardaki metinler üzerinde gerçekleştirilmiştir. Örneğin, haber sayfaları, makaleler gibi. Yapılan araştırmaların performansları da kullanılan veriye göre farklılık göstermiştir. Sadece makaleler üzerinde belirli bir performansa sahip bir sistem, Web sayfaları üzerinde aynı performansı göstermemektedir. Bu nedenle metin üzerinden bilgi hasatlanırken verinin yapısı ve kalitesi hasatlama performansını doğrudan etkilemektedir.

Bu tez çalışmasında literatür özetlenmiş, IF NER ve IFS adında 2 uygulama başarı ile geliştirilmiştir ve bu çalışmalar ile elde edilen sonuçlar alt başlıklarda değerlendirilmiştir.

4.1. IF NER

IF NER çalışmasında; her kişi ile ilgili resim, doğum tarihi ve meslek bilgileri çıkarılmış, 4 farklı veri üstünde başarı ile test edilmiştir.

Resim bilgisi hasatlama durumu değerlendirildiğinde, bazı durumlarda Web sayfalarından bu bilgi çıkarılamamıştır. Bunun nedeni örneğin arama motorunun getirdiği ilk 30 sayfa içerisinde resim bilgisini içermemesidir. Bu nedenle resim bilgisi bulunamadığı durumlarda genişletilmiş sorgular kullanılarak kişiye ait doğru resim bilgisi bu ikinci aşama ile tespit edilmiştir. Genişletilmiş arama özelliği sayesinde kişilere ait doğru resim bilgisinin bulunması %70,6 değerine ulaşmıştır. Tespit edilemeyen ya da yanlış olarak tespit edilen resimler incelendiğinde bunun nedeninin Web verisindeki zorluklardan kaynaklandığı görülmüştür. Örneğin resmin alt bilgisi olarak kişi ismi yazsa ya da dosya ismi içerisinde kişi ismi geçse dahi bu resmin o kişiye ait olmayabilmektedir. Bu zorluklara rağmen geliştirilen IF NER, yapılan testlerde %70,6'lık bir performans ile adalara ait doğru resimleri tespit edebilmiştir.

Doğum tarihi bilgisi hasatlamada cümle analizi yöntemi kullanılarak tespitler yapılmıştır. Öncelikle ANNIE tarafından tespit edilen tarih bilgileri elde edilmiş, daha sonra bu tarih bilgisinin geçtiği cümle analiz edilmiştir. IF, yapılan deneylerde %92,8 performansı ile doğru doğum tarihi bilgisini elde etmiştir. Web verisindeki gramer ve yazım hataları cümle analizi ile elde edilen verideki performansı düşürmüştür. Örneğin “*August 1, .1965*” şeklinde yazılmış olan tarih bilgisi hatalı bir formattadır. Aynı tarih bilgisi içerisinde virgül ve nokta olması yıl bilgisinin farklı bir cümlede olduğu anlamı çıkarılmıştır. Bu nedenle elde edilen doğum tarihi bilgisi “*August 1*” şeklinde olmuştur. Doğum tarihi bilgisi hasatlama performansını artırmak için XSL dönüşümü ile elde edilen bilgi ile birleştirilmiştir. XSL dönüşümü ile elde edilen bilginin daha doğru olacağı varsayımı yapılmıştır. Bunun nedeni ise belirli bir formattaki veriden hasatlama yapmanın daha güvenilir olacağıdır.

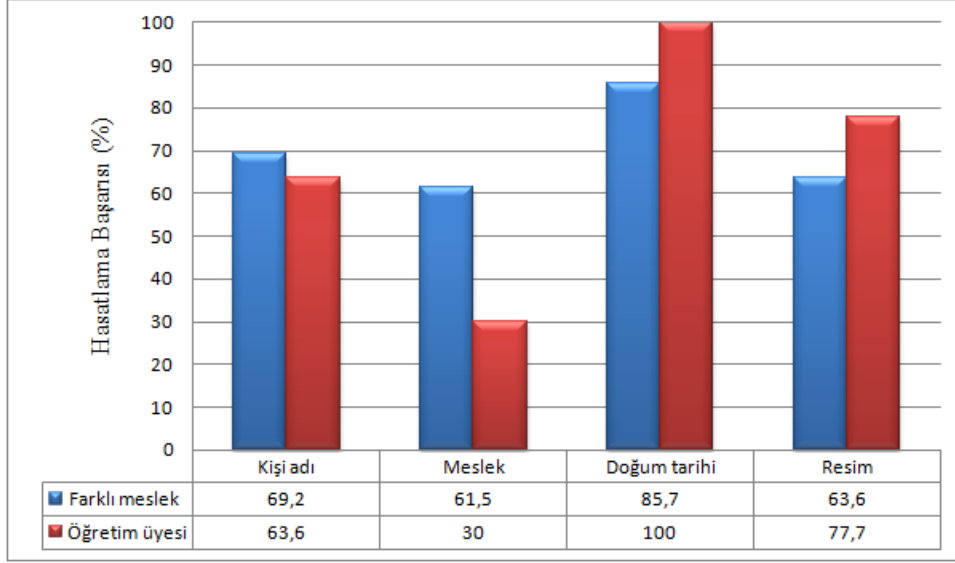
Meslek bilgisi ANNIE tarafından işaretlenen “jobTitle” ve “title” başlıklarının birleştirilmesi ile elde edilmiştir. %45,7’lık bir performans ile meslek bilgisi doğru olarak çıkarılmıştır. Doğum tarihi bilgisine benzer bir şekilde XSL dönüşümü ile cümle analizi ile elde edilen bilgiler birleştirilmiştir. XSL dönüşümünün daha doğru bilgi içerdiği varsayımı ile meslek bilgisi hasatlama performansı artırılmıştır.

IF hasatlanan bu 3 bilgi dışında XSL dönüşümü yardımı ile bazı farklı bilgileri de hasatlamaktadır. Bu bilgilerin hepsi “subject:predicate:object” şeklindedir. Bu üçlü ile bir semantik ilişki ifade edilmektedir. Örneğin “Şeref Sağıroğlu:title:Prof.” şeklinde bir üçleme ile kişinin unvanının ne olduğu bilgisi ifade edilmektedir. IF çalışmasında XSL dönüşümü Wikipedia sayfaları için kullanılmıştır. Belirli bir yapıya sahip herhangi bir Web sayfası için de bu yöntem sisteme kolayca entegre edilebilir.

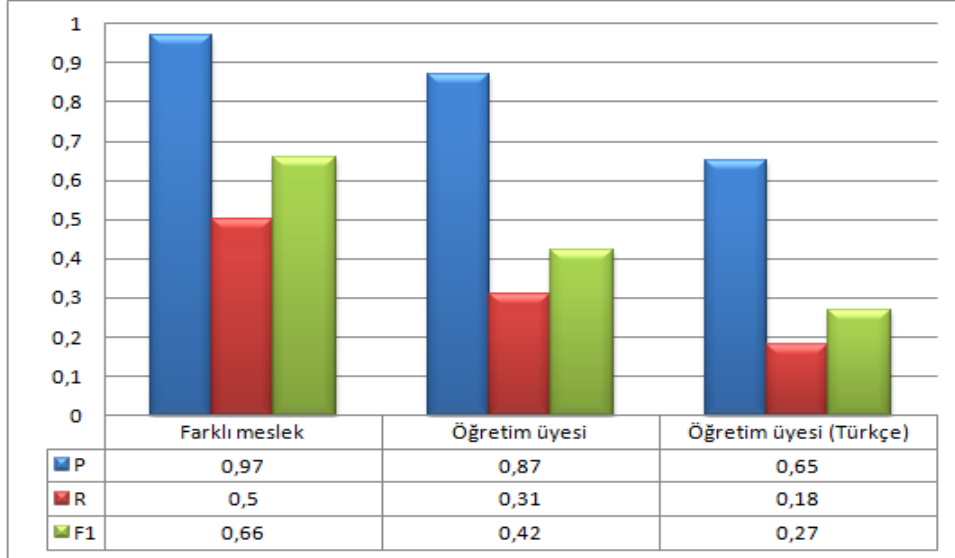
IF NER uygulamasını test ederken kullanılan 2 test verisi karşılaştırıldığında, genel olarak farklı meslek test verisinde ortalama başarı %70 olarak elde edilmiştir. Öğretim üyesi test verisinde ise ortalama başarı %67,8 olarak hesaplanmıştır. Bu durum da öğretim üyelerinin Web üzerindeki (İngilizce sayfalarda) tanınırlığının yeterli düzeyde olduğunu göstermektedir.

IF NER performans kriteri olarak çıkarılan bilgilerin yüzde olarak doğrulukları hesaplanmıştır. Bunun dışında P, R ve F1 değerleri hesaplanmıştır.

Çizelge 4.1. IF NER farklı meslek ve öğretim üyesi test verileri karşılaştırma



Çizelge 4.2. IF NER gruplandırma performansı



Performans değerleri diğer sistemler ile karşılaştırıldığında IF NER'in genel olarak diğer sistemlere yakın sonuçlar ürettiği görülmektedir. Fakat IF NER, bu performansı Web verisi üzerinden elde ettiği düşünüldüğünde başarılı bir sonuç elde edilmiştir.

Çizelge 4.3. Bilgi hasatlama çalışmaları karşılaştırma (IAF verisi ile)

Çalışma	Kullanılan Yöntem	Performans
IEPAD [14]	Düzenli ifade	1,0
STALKER [74]	Kural	0,92
SoftMealy [74]	Kural	0,99
IF NER	NER, Cümle Analizi	1,0

IAF verisine göre IF NER ve diğer çalışmalar karşılaştırıldığında IEPAD ve IF NER en iyi performansı göstermiştir. STALKER ve SoftMealy de iyi sonuçlar elde etmiştir fakat diğerlerine göre performansları düşüktür. IF NER, STALKER gibi fazladan yazılmış organizasyon bilgileri elenmiştir. IFS, bu veri üstünden karşılaştırılamamıştır. Bunun nedeni IFS'nin bir veritabanı kullanması ve doğrudan Web sayfaları üzerinden işlem yapamamasıdır.

Çizelge 4.4. IF NER İngilizce ve Türkçe veriye göre performansları

IF NER	P	R	F1
İngilizce veri	0,87	0,31	0,42
Türkçe Veri	0,65	0,18	0,27

Türkçe ile ilgili dokümanlar, kütüphane ve araçların kısıtlı olmasından dolayı IF NER, İngilizce dokümanları işlemek için geliştirilmiştir. İngilizce dilbilgisi kurallarına göre hasatlamalar yapılmaktadır. ANNIE'nin de Türkçe desteği bulunmamaktadır. Bu nedenle Türkçe veride performansın düştüğü açıkça gözlemlenmiştir.

4.2. IFS

IF NER'de karşılaşılan sorunlara çözüm getirmesi amacıyla IFS uygulaması geliştirilmiştir. IFS, hasatlama tekniği açısından birçok çalışmadan ayrılmaktadır. Günümüzde verileri semantik olarak saklamak ve bu veriler üzerinden bilgi elde etme önem kazanmaktadır. Bilgi etme süreci mümkün olduğu kadar hızlı ve doğruluk oranı yüksek olmalıdır. IFS, bu ihtiyaçları karşılamak için geliştirilmiştir. Diğer tüm çalışmalara kıyasla çok kısa sürelerde sonuç elde etmek mümkün

olmuştur. Örneğin IF NER’de bir kişiye ait bilgi hasatlama işlemi 10-15 dk sürebilmektedir. Bu süre IFS’de saniyeler ile ölçülmektedir. Bir arama motorunda arama yapmaya benzetilmektedir. Çok kısa sürelerde semantik bilgilere erişmek mümkündür. IFS’yi hem IF NER’den hem de diğer çalışmalardan ayıran bir diğer özellik de çok daha az karmaşık olmasıdır. Karmaşık algoritmalar ya da süreçler içermemektedir. Bunun nedeni verinin semantik olarak saklanıp, sorgulanabilmesidir. Bu sayede veri üzerinde doğal dil işleme gibi tekniklerin uygulanmasına gerek kalmamaktadır. Verinin semantik olarak saklanması kazanımları olmuştur.

Uygulama performansı: Semantik veriyi işlemek, düz metni işlemekten oldukça basittir. Veriler ve bu veriler arasındaki ilişkiler sayesinde bilgisayarın semantik veriyi işlemesi daha hızlı olacaktır.

Veri kalitesi: Web verisinin zorluklarından önceki bölümlerde bahsedilmiştir. Uygulamaların Web verisi üzerinde işlem yaparak elde ettikleri sonuçlar bu verilerin kalitesi ile doğrudan ilişkilidir. Yüksek kalitedeki veriler üzerinde çalışan uygulamalar daha iyi sonuçlar elde etmektedir. Freebase veritabanının kullanılmasının IFS’nin doğru sonuçlar üretmesinde büyük etkisi bulunmaktadır.

Farklı bağlamda çalışma: Uygulamaların çoğu belirli bir bağlamda çalışmaktadır. Örneğin IF NER, sadece kişisel bilgiler üzerinde ya da başka bir uygulama sadece haber sayfaları üzerinde çalışabilmektedir. IFS ise milyonlarca varlık üzerinde ve binlerce bağlam üzerinden bilgi elde edebilmektedir.

Farklı dil desteği: Dilden bağımsız varlık tanıma çalışması gibi çalışmalar farklı dillerde hasatlama işlemini yapmayı denemiştir [51]. Bu çalışmada sadece 2 dil desteklenmektedir. IFS’de ise 18 dil desteği vardır. Her dilde eşit miktarda bilgi edilemese bile bu dillerde sahip olunan tüm bilgilere ulaşılabilir. Çalışmaların çoğu ise dile bağımlıdır. Genellikle bir dil desteği vermektedirler.

4.3. Gelecekteki Çalışmalar ve Öneriler

IF NER çalışmasında cümle analizi ile bilgiler hasatlanmıştır. Bunun dışında genişletilmiş sorgular ve XSL dönüşümlerine yer verilmiştir. Asıl kullanılan yöntem cümle analizidir, genişletilmiş aramalar ve XSL dönüşümlerine bazı durumlarda yer verilmiştir.

Web verisindeki zorluklardan dolayı cümle analizi dışındaki yöntemler daha da genişletilebilir. Örneğin XSL dönüşümünün Wikipedia dışındaki başka Web siteleri için de XSL dokümanları hazırlanabilir.

Genişletilmiş aramalar sadece kişiye ait tespit edilemeyen resimler için uygulanmıştır. Bu yöntemin başarısı görülmüştür. Bu nedenle tespit edilemeyen başka alanlar için de genişletilmiş aramalar yapılabilir. Bu sayede ilk aramada elde edilen sayfaların dışında daha özel sayfalara ulaşılabilir ve eksik ya da tespit edilemeyen bilgiler hasatlanabilir.

Hasatlama işlemlerinin başındaki isimlendirilmiş varlıkların tespitinde ANNIE kullanılmıştır. Tüm hasatlama sürecinin süre olarak yaklaşık yarısını ANNIE'nin isimlendirilmiş varlıkları tanıması işlemi almaktadır. Sistemin performansını iyileştirmek için gruptandırma isimlendirilmiş varlıkların benzerliklerinin bakılması yerine farklı yöntemler kullanılabilir. Dokümanları gruptandırabilmek için tüm dokümanlar içerisindeki isimlendirilmiş varlıklar bulunmaktadır. Bunun yerine sadece gruptandırılmış dokümanlar üzerinden hasatlanacak olan grubun dokümanları üzerinden isimlendirilmiş varlık tespiti yapılabilir.

IFS, sadece Freebase veritabanı üzerinden sorgulama işlemi yapmaktadır. Fakat geliştirilen bu proje ile Freebase yazma servisleri ile veri girişi de yapılabilen bir ara yüz sağlanabilir. Bu sayede veritabanında bulunmayan bir veri olduğu zaman bunu kullanıcıların girmesi sağlanabilir. Bu sayede mevcut veritabanı kullanıcılar tarafından da genişletilebilir. IFS çevrimiçi olarak bir tarayıcı yardımı ile kullanılabilir. Günümüzde mobil uygulamalar oldukça yaygın olarak

kullanılmıştır. IFS uygulamasının mobil uygulaması yapılabilir. Bu sayede daha fazla son kullanıcıya ulaşılabilir.

Günümüzde semantik ilişkiler üzerinden bilgiye ulaşma ihtiyacı farklı alanlarda karşımıza çıkmaktadır. Örneğin çevrimiçi sosyal platformlarda kişilerin özelliklerine göre arama ihtiyaçları bulunmaktadır. Örneğin “Gazi Üniversitesi’nden mezun olan kişiler” şeklindeki bir sorguya cevap verebilmek için kişilerin mezun olduğu okul bilgisinin olması gerekmektedir. Bir diğer durum ise popüler arama motorlarının semantik sorguların çoğuna cevap verememesidir. Doğal dil ile ifade edilen sorgular kişiler tarafından çoğu zaman anahtar kelime tabanlı sorgulara çevrildikten sonra bu sorgular arama motorlarına iletilmektedir. Bu gibi nedenlerden dolayı Web üzerinde semantik sorgulamalara ihtiyaç bulunmaktadır. IF’in kullandığı teknikler geliştirilerek bu ihtiyaçlar karşılanabilir ve bir semantik arama motoru hizmeti verilebilir.

KAYNAKLAR

1. Bowman, C.M., Danzig, P.B., Hardy, D.R., Manber, U. Schwartz, M.F. "Harvest: A Scalable, Customizable Discovery and Access System" **Department of Computer Science, University of Colorado, Boulder**, 136-168 (1994).
2. Henzinger, M.R., Motwani, R., Silverstein, C., "Challenges In Web Search Engines", **SIGIR Forum**, 36: 11-12 (2002).
3. Brin, S., Page, L., "The anatomy Of A Large-Scale Hypertextual Web Search Engine" **Computer Networks and ISDN Systems**, 30: 107-117 (1998).
4. Ackerman, M.S., Alvarado, C., Karger, D.R., Teevan, J., "The Perfect Search Engine Is Not Enough: A Study Of Orienteering Behavior In Directed Search", **In Proceedings of CHI '04**, Vienna, 415-422 (2004).
5. Binggong, D., Jianfeng, G., Mu, L., Xiaojun, W., "Person Resolution in Person Search Results: WebHawk" **Proceedings of the 14th ACM international conference on Information and Knowledge Management**, NY USA, 163-170 (2005).
6. Andrews, K., Gütl, C., Maurer, H., "Future Information Harvesting and Processing on the Web", **European Telematics: advancing the information society**, 120–127 (1998).
7. Akbar, Z., Handoko, L.T., "A Simple Mechanism for Focused Web harvesting", **Proceeding of the International Conference on Advanced Computational Intelligence and Its Applications**, Depok, 178–183 (2008).
8. Antonatos, S., Gessiou, E., Kontaxis, G., Markatos, E.P., Petsas, T., Polakis, I., "Using Social Networks To Harvest Email Addresses", **WPES '10: Proceedings of the 9th annual ACM workshop on Privacy in the electronic society**, Chicago, 11–20 (2010).
9. Aranha, R., Cho, J., Crespo, A., Garcia-Molina, H., Hammer, J., "Extracting Semistructured Information from the Web" **Proceedings of the Workshop on Management of SemiStructured Data in conjunction**, 18–25 (1997).
10. Davies, J., Fensel D., Van Harmelen F., "Conclusion: Towards the Semantic Web Ontology-driven Knowledge Management" Towards The Semantic Web Ontology-driven Knowledge Management, **John Wiley & Sons Ltd**, Chichester 265-267 (2003).
11. Ding, X., "Product Configuration on the Semantic Web Using Multi-Agent", **Networking, Sensing and Control, 2008. ICNSC 2008 IEEE International Conference**, Quebec, 304-309 (2008).

12. Finke, M., Gasevic, D., Kaviani, N., Mohabbati, B., "Semantic Annotations of Feature Models for Dynamic Product Configuration in Ubiquitous Environments", *In Proceedings of the 4th International Workshop on Semantic Web Enabled Software Engineering at 7th International Semantic Web Conference*, Karlsruhe, 26-30 (2009).
13. Acquisti, A., Gross, R., "Predicting Social Security Numbers from Public Data" *Proceedings of National Academy of Science*, 106: 10975–10980 (2009).
14. Chang, C., Hsu, C., Lui, S. "Automatic Information Extraction From Semi-Structured Web Pages By Pattern Discovery", *Web Retrieval and Mining*, 35: 129-147 (2003).
15. Kumar, V., Sharma, K., Shrivastava, G., "Web Mining: Today And Tomorrow", *Electronics Computer Technology (ICECT)*, 399-403 (2011).
16. Menczer, F., "ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery", *Proceedings of the 14th International Conference on Machine Learning (ICML97)*, Nashville, 227-235 (1997).
17. Belew, R.K., Menczer, F., "Adaptive Information Agents in Distributed Textual Environments", *Proceedings of the 2nd International Conference on Autonomous Agents (Agents '98)*, ACM Press, Minneapolis, 157-164 (1998).
18. Chakrabarti, S., Dom, B., Van den Berg, M., "Focused Crawling: A New Approach To Topic-Specific Web Resource Discovery", *Computer Networks* 31, 1623-1640 (1999).
19. Internet : Hakia "Semantic Search Technology"
http://company.hakia.com/new/documents/White%20Paper_Semantic_Search_Technology.pdf (2010).
20. Bonsma, P., Bourdeau, M., Böhms, M., Kazi, A.S., "Semantic Product Modelling and Configuration: Challenges and Opportunities", *Journal of Information Technology in Construction*, 14: 507-525 (2009).
21. Artiles, J., Gonzalo, J., Sekine, S., "The Semeval-2007 WePS Evaluation: Establishing A Benchmark For The Web People Search Task," *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, PA USA, 64-69 (2007).
22. Liu, Z., Lu, Q., Xu, J., "High Performance Clustering For Web Person Name Disambiguation Using Topic Capturing", *Proceedings of the International Workshop on Entity-Oriented Search (EOS)*, Beijing, 1-6 (2011).

23. Ramos, J., "Using TF-IDF to Determine Word Relevance In Document Queries", ***In First International Conference on Machine Learning, New Brunswick: NJ, USA***, 303-322 (2003).
24. Dumais, S., Meek, S., Metzler, D., "Similarity Measures For Short Segments Of Text". ***Advances in Information Retrieval Lecture Notes in Computer Science*** 4425: 16-27 (2007).
25. Manning, D.C., Raghavan, P., Schutze, H., "Introduction to Information Retrieval", ***Cambridge University Press***, New York, 377 – 401 (2008).
26. Bagga, A., Baldwin, B., "Entity-Based Cross-Document Coreferencing Using The Vector Space Model", ***Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics***, Quebec Canada, 79–85 (1998).
27. Qin, J., Xu, Z., Yan, F., Zhu, H., "A Web Page Classification Algorithm Based On Link Information", ***in DCABES'11 Proceedings of the Tenth International Symposium on Distributed Computing and Applications to Business, Engineering and Science***, Wuxi, 82-86 (2011).
28. Ashish, N., Chen, Z., Kalashnikov, D.V., Mehrotra, S., Nuray, R., "Disambiguation Algorithm For People Search On The Web", ***In ICDE Poster***, İstanbul, 1258-1260 (2007).
29. Artiles, J., Gonzalo, J., Verdejo, F., "A Testbed for People Searching Strategies in the WWW", ***In Proceedings of the 28th annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR'05)***, Brazil, 569-570 (2005).
30. Bekkerman, R., McCallum, A., "Disambiguating Web Appearances Of People In A Social Network", ***In Proceedings of the World Wide Web Conference (WWW)***, Chiba, 463-470 (2005).
31. Luo, Y., "WAF -based Document Clustering Algorithm" ***Computer Science and Network Technology (ICCSNT), 2011 International Conference***, 1: 14-16 (2011).
32. Guo, J., Guo, H., Wang, Z., "Word Activation Forces Map Word Networks", ***Available from Nature Proceedings***, 25-41 (2011).
33. Li, X., "A Method For Person Name Disambiguation Based On Baidu Encyclopedia", ***Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference***, ChangChun, 423-426 (2011).

34. Jin, D., Shen, G., Zhang, H., “Algorithm of Disambiguation and Matching of Chinese Word Segmentation in Connected Strategies Research”, ***Advanced Materials Research***, 220: 702-1706 (2011).
35. Krishna, S., Pingali, P., Varma, V. “An Information Retrieval Approach To Spelling Suggestion”. ***In Proceedings of the 19th International Conference on WorldWideWeb(WWW'10), NewYork: ACM Press***, New York, 1135–1136 (2010).
36. Cafarella, M.J., Downey, D., Etzioni, O., Soderland, S., “KnowItNow: Fast, Scalable Information Extraction from the Web”, ***HLT '05 Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing***, Stroudsburg, 563-570 (2005).
37. Kushmerick, N., “Gleaning the Web”, ***IEEE Intelligent Systems***, 14: 20–22 (1999).
38. Jordan J., “Climbing Out of the Box and Into the Cloud: Building Web-Scale for Libraries”, ***Journal of Library Administration***, 51: 3-17 (2011).
39. Grishman, R., “Information Extraction: Techniques and Challenges”, ***Information Extraction A Multidisciplinary Approach to an Emerging Information Technology, Springer Berlin / Heidelberg***, 10-27 (1997).
40. Holmes, W., Kogut, P., “AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages”, ***In Proceedings of the First International Conference on Knowledge Capture***, Victoria, 49-79 (2001).
41. Kalashnikov, D.V., Mehrotra, S., Nuray, R., “Towards Breaking The Quality Curse: A Web-Querying Approach To Web People Search”, ***Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval***, Singapore, 27-34 (2008).
42. Cafarella, M.J., Downey, D., Etzioni, O., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D.S., Yates, A., “Web-Scale Information Extraction In Knowitall”, ***In WWW***, New York, 100-110 (2004).
43. Aseltine, J., Fisher, D., Lehnert, W., Soderland, S., “CRYSTAL: Inducing A Conceptual Dictionary”, ***In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence***, San Francisco, 1314–1321 (1995).
44. Grover, C., Mikheev, A., Moens, M., “Named Entity Recognition Without Gazetteers”, ***In Proceedings of the 9th EACL***, Bergen, 1–8 (1999).
45. Mikheev, A., “Feature Lattices For Maximum Entropy Modelling”, ***In Proceedings of the 36th Annual Meeting of the Association for Computational***

Linguistics and Proceedings of the 17th International Conference on Computational Linguistics, Montreal, 848-854 (1998).

46. Clark, S., Mausam, E.O., Ritter, A., “Named Entity Recognition In Tweets: An Experimental Study,” ***In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)***, Massachusetts, 1524–1534 (2011).
47. Pereira, F., Sha, F., “Shallow Parsing With Conditional Random Fields”, ***In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL***, Edmonton, 134-141 (2003).
48. Internet : Apache “OpenNLP”
<http://opennlp.apache.org/>, (2011)
49. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J., “Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge”, ***In SIGMOD '08***, Vancouver, 1247–1250 (2008).
50. Hall, D., Manning, C.D., Nallapati, R., Ramage, D., “Labeled Lda: A Supervised Topic Model For Credit Attribution In Multi-Labeled Corpora”, ***In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1, EMNLP '09***, Singapore, 248–256 (2009).
51. De Meulder, F., Sang, E.F., “Introduction to the CoNLL-2003 Shared Task: Language independent Named Entity Recognition”, ***In Proceedings of CoNLL-2003***, Edmonton, 142-147 (2003).
52. Daelemans, W., Van den Bosch, A., Van der Sloot, K., Zavrel, J., “MBT: Memory- Based Tagger, version 1.0, Reference Guide”, ***ILK Technical Report ILK-0209, University of Tilburg***, 4-10 (2002).
53. Saggion, H., Szasz, S., “Multi-domain Crosslingual Information Extraction from Clean and Noisy Texts”, ***In Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology, Cuiab´a, Brazil. BCS***, 24-26 (2011).
54. Internet : Google “Google Çeviri”
<http://translate.google.com> (2013).
55. Aswani, N., Bontcheva, K., Cunningham, H., Damljanovic, D., Funk, A., Greenwood, M.A., Gorrell, G., Heitz, T., Maynard, D., Peters, W., Petrak, J., Roberts, I., Roberts, A., Saggion, H., Tablan, V., Yaoyong, L., “GATE Basics” Developing Language Processing Components with GATE Version 7 (a User Guide), ***The University of Sheffield, Department of Computer Science***, 3-117 (2012).

56. Chen, Y., Huang, C.R., Lee, S.Y.M., “A Robust Web Personal Name Information Extraction System”, *Expert Systems with Applications*, 39: 2690-2699 (2012).
57. Banko, M., Broadhead, M., Cafarella, M.J., Etzioni, O., Soderland, S., “Open Information Extraction From The Web”, *Proceedings of IJCAI07*, Hyderabad, 68-74 (2007).
58. Bontcheva, K., Cunningham, H., Maynard, D., Tablan, V., “GATE: an Architecture for Development of Robust HLT Applications”, *In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*, 168–175 (2002).
59. Schwartz, C., “Web Search Engines”, *Journal of the American Society for Information Science*, 49: 973–982 (1998).
60. Voutilainen, A., “The Oxford Handbook of Computational Linguistics-Chapter 11 Part-Of Speech Tagging”, Editor, Ruslan Mitkov, *Oxford University Press*, 219-232 (2004).
61. Islam, A., Inkpen, D., “Managing the Google Web 1T 5-gram Data Set”. *Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE'09)*, Roma, 1-5 (2009).
62. Internet : Vaadin “Web Framework”
<https://vaadin.com/home> (2012).
63. Internet : Google “GWT – Google Web Toolkit”
<https://developers.google.com/web-toolkit/> (2013).
64. Internet : Sheffield Üniversitesi “ANNIE: A Nearly-New Information Extraction System” <http://gate.ac.uk/sale/tao/splitch6.html#chap:annie> (2002).
65. Internet : Spring “Spring Framework”
<http://www.springsource.org/> (2013).
66. Internet : Google “Google Custom Search”
<http://www.google.com/cse> (2013).
67. Internet : Sheffield Üniversitesi “GATE Embedded”
<http://gate.ac.uk/sale/tao/splitch7.html#chap:api> (2011).
68. Internet : Wordnet “Wordnet A Lexical Database For English”
<http://wordnet.princeton.edu/> (1995).

69. İnternet : Stanford Üniversitesi “Stanford Log-linear Part-Of-Speech Tagger”
<http://nlp.stanford.edu/software/tagger.shtml> (2010).
70. İnternet : Opengroup “Regular Expressions”
http://pubs.opengroup.org/onlinepubs/009695399/basedefs/xbd_chap09.html
(2004).
71. İnternet : W3C “The Extensible Stylesheet Language Family (XSL)”
<http://www.w3.org/Style/XSL> (2009).
72. İnternet : Wikipedia "PersonData yapısı"
[http://en.wikipedia.org/wiki/Jessica_Fox_\(canoeist\)](http://en.wikipedia.org/wiki/Jessica_Fox_(canoeist)) (2013).
73. İnternet : Wikipedia "VCard yapısı"
http://en.wikipedia.org/wiki/Jessica_Fox (2013).
74. Chang, C.C., Hsu, C.N., “Finite-state transducers for semistructured text mining”, *In Proceedings of IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, Pergamon Press, NY, Stockholm, 38–49 (1999).
75. İnternet : Microsoft “Asp.Net MVC”
<http://www.asp.net/mvc> (2011).

EKLER

EK-1. Dilden bağımsız isimlendirilmiş varlık tanıma

Çizelge 1.1. Dilden bağımsız isimlendirilmiş varlık tanıma İngilizce performansı [51]

İngilizce Test	P (%)	R (%)	$F_{\beta=1}$
Florian	88,99	88,54	88,76±0,7
Chieu	88,12	88,51	88,31±0,7
Klein	85,93	86,21	86,07±0,8
Zhang	86,13	84,88	85,50±0,9
Carreras(a)	84,05	85,96	85,00±0,8
Curran	84,29	85,50	84,89±0,9
Mayfield	84,45	84,90	84,67±1,0
Carreras(b)	85,81	82,84	84,30±0,9
McCallum	84,52	83,55	84,04±0,9
Bender	84,68	83,18	83,92±1,0
Munro	80,87	84,21	82,50±1,0
Wu	82,02	81,39	81,70±0,9
Whitelaw	81,60	78,05	79,78±1,0
Hendrickx	76,33	80,17	78,20±1,0
De Meulder	75,84	78,13	76,97±1,2
Hammerton	69,09	53,26	60,15±1,3
Baseline	71,91	50,90	59,61±1,2

EK-1. (Devam) Dilden bağımsız isimlendirilmiş varlık tanıma

Çizelge 1.2. Dilden bağımsız isimlendirilmiş varlık tanıma Almanca sonucu [51]

Almanca Test	P(%)	R(%)	$F_{\beta=1}$
Florian	83,87	63,71	72,41±1,3
Klein	80,38	65,04	71,90±1,2
Zhang	82,00	63,03	71,27±1,5
Mayfield	75,97	64,82	69,96±1,4
Carreras(a)	75,47	63,82	69,15±1,3
Bender	74,82	63,82	68,88±1,3
Curran	75,61	62,46	68,41±1,4
McCallum	75,97	61,72	68,11±1,4
Munro	69,37	66,21	67,75±1,4
Carreras(b)	77,83	58,02	66,48±1,5
Wu	75,20	59,35	66,34±1,3
Chieu	76,83	57,34	65,67±1,4
Hendrickx	71,15	56,55	63,02±1,4
De Meulder	63,93	51,86	57,27±1,6
Whitelaw	71,05	44,11	54,43±1,4
Hammerton	63,49	38,25	47,74±1,5
Baseline	31,86	28,89	30,30±1,3

Çizelge 1.3. İsimlendirilmiş varlık tanıma çalışmaları karşılaştırma

Çalışma	Kullanılan Yöntem	P(%)	R(%)	F1
NER [44]	İsimlendirilmiş Varlık Tanıma	79,6	74	0,76
T-NER [46]	İsimlendirilmiş Varlık Tanıma	65	42	0,51
Dilden Bağımsız NER [51]	İsimlendirilmiş Varlık Tanıma	82	80	0,8

EK-2. IAF verisi

Çizelge 2.1. IAF 1. verisi [37]

	İsim	E-Posta	Diğer Ad	Son Güncelleme	Organizasyon	Servis Sağlayıcı
'Lithium' J Smith	Var	Var	Yok	Var	Var	Yok
'Sir Brand' Gregrobin Smith	Var	Var	Var	Var	Var	Var
(raig Smith	Var	Var	Yok	Var	Var	Yok
- Richard Smith	Var	Var	Var	Var	Var	
- David S Smith	Var	Var	Var	Var	Yok	Var
- Rev Larry Smith	Var	Var	Var	Var	Yok	Yok
- Tracey L Smith	Var	Var	Var	Var	Yok	Yok
1 Smith	Var	Var	Yok	Var	Var	Yok
A Smith	Var	Var	Yok	Var	Yok	Yok
A Smith	Var	Var	Yok	Var	Yok	Var

Çizelge 2.2. IAF 2. verisi [37]

	İsim	E-Posta	Diğer Ad	Son Güncelleme	Organizasyon	Servis Sağlayıcı
Chris Kushmerick	Var	Var	Yok	Var	Var	Yok
Christopher Kushmerick	Var	Var	Yok	Var	Var	Yok
Jim Kushmerick	Var	Var	Yok	Var	Var	Yok
Jim Kushmerick	Var	Var	Yok	Var	Var	Yok
Nicholas Kushmerick	Var	Var	Yok	Var	Var	Yok
Nicholas Kushmerick	Var	Var	Yok	Var	Var	Yok
Nicholas Kushmerick	Var	Var	Yok	Var	Yok	Yok
Nicholas Kushmerick	Var	Var	Yok	Var	Var	Yok

EK-2. (Devam) IAF verisi

Çizelge 2.3. IAF 3. verisi [37]

	İsim	E-Posta	Diğer Ad	Son Güncelleme	Organizasyon	Servis Sağlayıcı
Weld	Var	Var	Yok	Var	Var	Yok
Weld	Var	Var	Yok	Var	Var	Var
Bruce Weld	Var	Var	Yok	Var	Var	Var
Charles Weld	Var	Var	Yok	Var	Var	Yok
Dave Weld	Var	Var	Yok	Var	Yok	Yok
Dave Weld	Var	Var	Yok	Var	Yok	Var
David Weld	Var	Var	Yok	Var	Var	Yok
Dick Weld	Var	Var	Yok	Var	Var	Yok
Donald Weld	Var	Var	Yok	Var	Var	Yok
Erika Weld	Var	Var	Var	Var	Var	Yok

Çizelge 2.4. IAF 4. verisi [37]

	İsim	E-Posta	Diğer Ad	Son Güncelleme	Organizasyon	Servis Sağlayıcı
Amitai Etzioni	Var	Var	Yok	Var	Var	Yok
Michael Etzioni	Var	Var	Yok	Var	Var	Yok

EK-2. (Devam) IAF verisi

Çizelge 2.5. IAF 5. verisi [37]

	İsim	E-Posta	Diğer Ad	Son Güncelleme	Organizasyon	Servis Sağlayıcı
A Barton	Var	Var	Yok	Var	Var	Yok
Aaron Joseph Barton	Var	Var		Var	Var	Yok
Aaron Barton	Var	Var	Yok	Var	Var	Var
Adam Barton	Var	Var	Yok	Var	Var	Yok
Adam Carlos Barton	Var	Var	Yok	Var	Var	Yok
Adam Carlos Barton	Var	Var	Yok	Var	Var	Yok
Alan B Barton	Var	Var	Yok	Var	Var	Yok
Alan Russell Barton	Var	Var	Yok	Var	Yok	Yok
Alan Barton	Var	Var	Yok	Var	Var	Yok
Alexander Barton	Var	Var	Yok	Var	Var	Yok

Çizelge 2.6. IAF 6. verisi [37]

	İsim	E-Posta	Diğer Ad	Son Güncelleme	Organizasyon	Servis Sağlayıcı
Matthew Davis	Var	Var	Yok	Var	Var	Yok
\$j>g\$h Mathusala Davis	Var	Var	Var	Var	Var	Yok
-ronald H Davis	Var	Var	Yok	Var	Var	Yok
-ronald H Davis	Var	Var	Yok	Var	Var	Yok
-ronald H Davis	Var	Var	Yok	Var	Var	Yok
A Or C Davis	Var	Var	Var	Var	Var	Yok
A E Davis	Var	Var	Yok	Var	Var	Yok
A Davis	Var	Var	Var	Var	Var	Var
A J Davis	Var	Var	Yok	Var	Var	Yok
A Davis	Var	Var	Yok	Var	Yok	Yok

EK-2. (Devam) IAF verisi

Çizelge 2.7. IAF 7. verisi [37]

	İsim	E-Posta	Diğer Ad	Son Güncelleme	Organizasyon	Servis Sağlayıcı
Accuracy Lesh	Var	Var	Var	Var	Var	Yok
Adam Lesh	Var	Var	Yok	Var	Yok	Var
Andrew Lesh	Var	Var	Yok	Var	Yok	Var
Anne Lesh	Var	Var	Yok	Var	Var	Yok
Ayal Lesh	Var	Var	Yok	Var	Var	Yok
Charles J Lesh Jr	Var	Var	Yok	Var	Var	Var
Charles J Lesh	Var	Var	Yok	Var	Var	Yok
Charles J Lesh Jr	Var	Var	Yok	Var	Var	Yok
DAN LESH	Var	Var	Yok	Var	Var	Yok
David Lesh	Var	Var	Yok	Var	Yok	Yok

Çizelge 2.8. IAF 8. verisi [37]

	İsim	E-Posta	Diğer Ad	Son Güncelleme	Organizasyon	Servis Sağlayıcı
Williamson	Var	Var	Yok	Var	Yok	Yok
Williamson	Var	Var	Yok	Var	Yok	Yok
Williamson	Var	Var	Yok	Var	Var	Yok
Williamson	Var	Var	Yok	Var	Var	Yok
Williamson	Var	Var	Yok	Var	Yok	Var
Williamson	Var	Var	Yok	Var	Yok	Var
Williamson	Var	Var	Yok	Var	Var	Yok
Williamson	Var	Var	Yok	Var	Var	Yok
Williamson	Var	Var	Yok	Var	Var	Yok
Williamson	Var	Var	Var	Var	Var	Yok

EK-3. Wikipedia veri yapısı

Çizelge 3.1. Wikipedia “PersonData” yapısı [72]

```

<table id="persondata" class="persondata" style="border:1px solid #aaa;
display:none; speak:none;">
  <tr><th colspan="2">
    <a href="/wiki/Wikipedia:Persondata"
title="Wikipedia:Persondata">Persondata</a>
  </th>
</tr><tr>
  <td class="persondata-label" style="color:#aaa;">Name</td>
  <td>Fox, Jessica</td>
</tr><tr>
  <td class="persondata-label" style="color:#aaa;">Alternative
names</td><td></td></tr><tr>
  <td class="persondata-label" style="color:#aaa;">Short
description</td>
  <td>Australian canoeist</td>
</tr><tr>
  <td class="persondata-label" style="color:#aaa;">Date of birth</td>
  <td>11 June 1994</td>
</tr><tr>
  <td class="persondata-label" style="color:#aaa;">Place of
birth</td>
  <td>Marseille, France</td>
</tr><tr>
  <td class="persondata-label" style="color:#aaa;">Date of
death</td>
  <td></td>
</tr><tr>
  <td class="persondata-label" style="color:#aaa;">Place of
death</td><td></td></tr></table>

```

EK-3. (Devam) Wikipedia veri yapısı

Çizelge 3.2. Wikipedia “VCard” yapısı [73]

```

<table class="infobox biography vcard" cellspacing="3"
      style="border-spacing: 3px; width:22em;">
  <tbody><tr><th colspan="2" class="" style="text-align:center; font-
size:125%; font-weight:bold;">
    <span class="fn">Jessica Fox</span>
  </th></tr>
<tr class=""><td colspan="2" class="" style="text-align:center;">
  <a href="/wiki/File:Jessica_Fox_actress.jpg" class="image">
    
    </a></td></tr>
  <tr class=""><th scope="row" style="text-align:left;">Born</th>
    <td class="" style=""><span style="display:none">
      (<span class="bday">1983-05-19</span>)
    </span>19 May 1983<span class="noprint
    <span class="noprintForceAgeToShow">(age&nbsp;29)
    </span><sup id="cite_ref-autogenerated1_1-0"
class="reference"><a href="#cite_note-autogenerated1-
1"><span>[</span>1<span></span>]</span>
    </a></sup><br>
    <span class="birthplace">Taplow, Berkshire,
England</span></td></tr><tr class="">
  <th scope="row" style="text-align:left;">Occupation</th>
  <td class="role" style="">Actor</td></tr><tr class="">
  <th scope="row" style="text-align:left;">Years active</th>
  <td class="" style="">1992–present</td>
</tr></tbody></table>

```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ERAKMAN, Ömer

Uyruğu : T.C.

Doğum tarihi ve yeri : 16.07.1984 Mardin

Medeni hali : Bekar

Telefon : 0 (555) 557 35 95

e-mail : omererakman@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek Lisans	Gazi Üniversitesi/ Bilgisayar Mühendisliği Bölümü	2013
Lisans	Hacettepe Üniversitesi/ Bilgisayar Mühendisliği Bölümü	2008
Lise	Ankara Atatürk Anadolu Lisesi	2003

İş Deneyimi

Yıl	Yer	Görev
2010-2013	Innova Bilişim A.Ş.	Bilgisayar Mühendisi
2008-2010	EES A.Ş.	Bilgisayar Mühendisi

Yabancı Dil

İngilizce

Hobiler

Java Web teknolojileri, Web güvenliği, Futbol

