TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BAĞLI VERİ KAYNAKLARI VE İLİŞKİLERİ KULLANILARAK HABERLERİN ÖBEKLENDİRİLMESİ

YÜKSEK LİSANS TEZİ

Mehmet Mert YÜCESAN

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı Prof. Dr. Erdoğan DOĞDU

Aralık 2016

Fen Bilimleri Enstitüsü Onayı

.....
Prof.Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

.....
Doç.Dr. Oğuz ERGİN
Anabilimdalı Başkan V.

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 131111033 numaralı Yüksek Lisans Öğrencisi **Mehmet Mert YÜCESAN**'ın ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"BAĞLI VERİ KAYNAKLARI VE İLİŞKİLERİ KULLANILARAK HABERLERİN ÖBEKLENDİRİLMESİ"** başlıklı tezi 14.12.2016 tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı:	**Prof.Dr. Erdoğan DOĞDU** TOBB Ekonomi ve Teknoloji Üniversitesi	.....
Jüri Üyeleri:	**Prof.Dr. Mehmet Ali AKÇAYOL (Başkan)** Gazi Üniversitesi	.....
	Yrd. Doç. Dr. Ahmet Murat ÖZBAYOĞLU TOBB Ekonomi ve Teknoloji Üniversitesi	.....

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Mehmet Mert YÜCESAN

ÖZET

Yüksek Lisans Tezi

BAĞLI VERİ KAYNAKLARI VE İLİŞKİLERİ KULLANILARAK HABERLERİN ÖBEKLENDİRİLMESİ

Mehmet Mert YÜCESAN

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı Prof.Dr. Erdoğan DOĞDU

Tarih: Aralık 2016

Metin veya doküman öbeklendirilmesi, aynı konuyla ilgili olan metin belgelerinin belirlenerek gruplandırılması işlemidir. Bu işlem, metin belgelerinin sayısının artmaya devam ettiği sürekli büyüyen Web için özellikle önemlidir. Haber öbeklendirilmesi bu alanda, haber belgelerinin konu bazında sınıflandırılmasının hedeflendiği özel bir konudur. Bu probleme ilişkin daha önce geliştirilmiş çözümler, belgelerin içlerinde geçen kelimelerle ve bu kelimelerin sıklıklarıyla temsil edildiği "sözcük çantası" yaklaşımını kullanmıştır ve öbeklendirme işlemi belgelerin bu gösterimi kullanılarak ölçülen benzerlikler kullanılarak yapılmıştır. Bununla birlikte, bu yaklaşım sözcüklerin anlamını veya önemini dikkate almaz ve sözcüklerdeki muğlaklık çözümlenmez. Bu çalışmada doküman veya haber öbeklendirilmesi konusunda "bağlı veri" kullanan yeni bir yaklaşım geliştirilmiştir. Bu yaklaşımda haber belgelerindeki sözcükler ve cümleler, DBpedia gibi bağlı veri bilgi tabanlarındaki gerçek dünya karşılıklarına eşlenir ve belgeler sahip oldukları bağlı veri varlıklarıyla temsil edilmektedir. Daha sonra haberler bu varlıklar ve bu varlıkların kategori hiyerarşisi benzerlikleri kullanılarak öbeklendirilmektedir. Değerlendirme sonuçları, geliştirilen yaklaşımın kelime çantasına göre daha iyi sonuç verdiğini göstermektedir.

Anahtar Kelimeler: Haber öbekleme, Bağlı veri, Anlamsal Web, Anlamsal benzerlik.

ABSTRACT

Master of Science

NEWS CLUSTERING USING LINKED DATA RESOURCES AND THEIR RELATIONSHIPS

Mehmet Mert YÜCESAN

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Prof.Dr. Erdoğan DOĞDU

Date: December 2016

Text clustering or document clustering is the task of identifying and grouping text documents that are about the same topic. This is especially important for the ever growing Web where the number of free-text documents just keep increasing. News clustering is a special task in this domain in which the goal is to classify news documents by topic. Earlier solutions on this problem utilized "bag of words" approach in which documents are represented with words and their frequencies in documents, and the clustering task measures the similarity of documents using this representation. However, this approach does not take into consideration the meaning or the importance of words and ambiguity in words is not resolved. We present a new approach to document or news clustering, we utilize "linked data". We map words or phrases in news documents to their real-world counterparts in "linked data" knowledge bases such as DBpedia and represent documents with linked data entities they have. Then we cluster documents using these entities and their category hierarchy similarities. Evaluation results show that our approach performs better than the bag of words approach.

Keywords: News clustering, Linked data, Semantic Web, Semantic similarity.

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren hocam Prof.Dr. Erdoğan Doğdu, kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim üyelerine, eğitimim boyunca bana burs veren TOBB Ekonomi ve Teknoloji Üniversitesine ve destekleriyle her zaman yanımda olan aileme ve arkadaşlarıma çok teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET . . . iv
ABSTRACT . . . v
TEŞEKKÜR . . . vi
İÇİNDEKİLER . . . vii
ŞEKİL LİSTESİ . . . ix
ÇİZELGE LiSTESİ . . . x
KISALTMALAR . . . xi
1. GİRİŞ . . . 1
1.1 Problem ve Motivasyon . . . 2
1.2 Tezin Katkıları . . . 2
2. WEB BİLGİ KAYNAKLARI VE KULLANIM ALANLARI . . . 5
2.1 Web bilgi kaynakları . . . 5
2.1.1 WordNet . . . 5
2.1.2 Wikipedia . . . 5
2.1.3 Semantik Web . . . 6
2.1.4 Bağlı veri . . . 7
2.2 Web bilgi tabanları kullanım alanları . . . 9
2.2.1 Açık bilgi çıkarma . . . 9
2.2.2 Soru cevaplama . . . 9
2.2.3 Doküman öbekleme . . . 10
3. İLGİLİ ÇALIŞMALAR . . . 11
3.1 Öbekleme Yöntemleri . . . 11
3.1.1 K-means öbekleme . . . 11
3.1.2 Hiyerarşik öbekleme . . . 12
3.2 Döküman ve Haber Öbeklemesi . . . 13
3.3 Kelime Çantası . . . 14
3.4 Gizli Anlamsal Analiz . . . 15
3.5 Bilgi Tabanları ve Bağlı Veri . . . 15
3.5.1 WordNet . . . 16
3.5.2 Wikipedia . . . 17
3.5.3 Bağlı veri . . . 17
4. BAĞLI VERİ KAYNAKLARI KULLANILARAK HABERLERİN ÖBEKLENDİRİLMESİ . . . 21
4.1 Bağlı Veri Kaynakları Arasındaki Anlamsal Benzerliklerin Hesaplanması . . . 22
4.2 Haber Dökümanları Arasındaki Benzerliklerin Hesaplanması . . . 25
4.3 Öbekleme . . . 25
4.4 Uygulama . . . 26
5. DEĞERLENDİRME . . . 29
5.1 Veri Seti . . . 29
5.1.1 BBC news . . . 29
5.1.2 20Newsgroup . . . 29

5.2 Deneyler 30
5.3 Analiz 33
5.4 Süre Analizi 35
6. SONUÇ 37
KAYNAKLAR 38
ÖZGEÇMİŞ 45

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: LOD bulut diagramı 8
Şekil 3.1: Örnek dendrogram 13
Şekil 4.1: Albert Einstein, Peter Higgs ve Gary Speed için oluşturulan 5 seviyeli tür hiyerarşileri 23
Şekil 4.2: Haber öbeklendirilmesi süreci 26
Şekil 5.1: Süre analizi 35

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1: Uzaklık ölçüteri 12
Çizelge 3.2: Doküman öbeklendirilmesinde dış bilgi kaynağı kullanan yayınlar . . 20
Çizelge 5.1: BBC news veri seti: haber kategorileri ve kategori başına haber sayısı . 29
Çizelge 5.2: 20Newsgroup veri seti: seçilen gruplar ve grup başına haber sayısı . . 30
Çizelge 5.3: Kullanılan veri setlerinde Alchemy API tarafından bulunan farklı varlıkların sayıları 30
Çizelge 5.4: Vektör örneği (*m*:#doküman, *n*:#varlık) 31
Çizelge 5.5: Hassaslık (P), Hatırlama (R) ve F1 puanları. 34

KISALTMALAR

BOW : Bag-of-Words - Kelime çantası
LOD : Linked Open Data - Açık Bağlı Veri
NMF : Non-Negative Natrix Factorization - Negatif Olmayan Matris Faktorizasyonu
NLP : Natural Language Processing - Doğal Dil İşleme
RDF : Resource Description Framework - Kaynak Tanımlama Çerçevesi
RDFS : RDF Schema - Kaynak Tanımlama Çerçevesi Şeması
OWL : Web Ontologu Language - Web Ontoloji Dili
URI : Uniform Resource Identifier - Tekbiçimli Kaynak Tanımlayıcısı
LSA : Latent Semantic Analysis - Gizli Anlamsal Analiz
TF : Term Frequency - Terim Sıklığı
IDF : Inverse Document Frequency - Ters Doküman Sıklıgı
P : Precision - Hassaslık
R : Recall - Hatırlama

1. GİRİŞ

Öbekleme birbirine benzeyen nesneleri bir arada gruplama işlemidir. Yaygın olarak çalışılan bir makina öğrenmesi konudur ve öbekleme konusunda birçok yöntem geliştirilmiştir. Doküman öbeklenmesi de metin belgelerinin konu, içerik veya kategori bakımından gruplanması ile ilgilenmektedir. İnternette dijital içeriklerin miktarının hızla artmasıyla bu konu oldukça önem kazanmıştır. Doküman öbeklemesi, dokümanların düzenlenmesi, aranması, korpus özetlemesi, doküman sınıflandırılması vb. birçok alanda kullanılmaktadır [1].

Haber öbeklemesi ise haber dokümanlarının gruplandırmasıyla ilgilenen doküman öbeklenmesinin özel bir türüdür. Haber öbeklemesindeki amaç aynı öbek içerisinde yer alan haberlerin konu veya kategori olarak birbirlerine benzerken diğer öbeklerdekine benzemeyen şekilde gruplara ayrılmasıdır. Haber öbeklenmesi özellikle arama motorları gibi Web uygulamaları için oldukça önemlidir. Son yıllarda haberler ve medya internete yönelmeye başladı ve internette yayınlanan haber miktarı hızlı bir şekilde arttı. Haber dokümanlarının sınıflandırılması konusunda son zamanlardaki çalışmalarda metinlerdeki kavramları ve varlık isimlerini bulmak ya da metinlerden anlam çıkarmak amacıyla WordNet[1] gibi bilgi tabanları kaynak olarak kullanılmaya başlandı.

"Bağlı veri"[2] de kullanılan kaynak türlerinden birisidir. Bağlı veri ilk olarak Tim Berners Lee tarafından ortaya atıldı ve "Web'deki yapılandırılmış verinin yayınlanması ve bağlanması için kullanılan en iyi uygulamalar topluluğu" olarak tanımlandı [9, 10].

Bu tez çalışmasında haber öbekleme problemine çözüm olarak anlamsal bir doküman benzerliği (semantic document similarity) yöntemi geliştirilmiştir. Bu yöntemde bağlı veri kaynakları kullanılarak dokümanlar arasındaki benzerlik hesaplanarak bu hesaplamalara göre öbekleme yapılmıştır. Geliştirilen yöntem ile daha önce kullanılan başka yöntemlerle de karşılaştırma yapılmıştır. Tezin konusu olan öbekleme, öbekleme yöntemleri ve bağlı veri hakkındaki giriş bilgileri aşağıda verilmiştir.

[1] https://wordnet.princeton.edu/

[2] http://www.w3.org/standards/semanticweb/data

1.1 Problem ve Motivasyon

Web'deki verinin miktarı hızla artmaktadır. İnsanlar gün geçtikçe interneti temel bilgi kaynağı olarak kabul etmeye başlamışlardır. Bu sebeple istenilen bilgiye, özellikle haberlere ulaşma konusu önem kazanmıştır.

Bu konuda geliştirilen bazı uygulamalar bulunmaktadır. Bunların başında da Google News[3] gelmektedir. Google News ücretsiz çevrimiçi bir haber toplayıcıdır. İnternet üzerindeki birçok kaynaktan haberleri toplayıp, birbirleriyle ilgili olanları bir arada gruplandırarak kullanıcılara sunmaktadır.

İlgili haberlerin bir arada gruplandırılması işlemi haber öbeklemesinin konusudur. Bu konuda yapılan ilk çalışmalar dokümanların içerdikleri kelimeler ve bu kelimelerin sayıları ile temsil edildiği "bag of words" yaklaşımını kullanmaktadır. Bu yaklaşımda kelimelerin yazılışlarına bakılarak dokümanların birbirlerine olan benzerlikleri hesaplanmakta ve dokümanlar öbeklenmektedir. Fakat kelimelerin anlamları veya birbirleriyle olan anlamsal ilişkileri dikkate alınmamaktadır.

Son zamanlarda gelişmekte olan anlamsal Web (semantic Web) ve bağlı veriler (linked data) dokümanların benzerliğinin hesaplanması konusunda anlamsal bir yaklaşım sağlamaktadır. Kelimelerin anlamları ve ilişkilerinin, haber makalelerinin benzerliğinin hesaplanmasına katılması öbekleme performansını artırmaktadır. Bu tezde, bağlı veriler kullanılarak doküman benzerliği ölçmek için geliştirilen anlamsal benzerlik yöntemleri sunulacak ve performansları analiz edilecektir.

1.2 Tezin Katkıları

Bu tez çalışmasında, haber öbeklemesi problemine bağlı veri kaynakları kullanılarak anlamsal bir yaklaşım sergilenerek yeni bir yöntem geliştirilmiştir. Tezin katkıları özet olarak aşağıda sıralanmıştır:

- Haber öbeklemesi konusunda kullanılmak üzere yeni bir veri seti (BBC News[4]) oluşturulmuştur.
- Doküman öbeklemesi için bağlı veri kaynaklarını kullanan yeni bir anlamsal benzerlik yöntemi geliştirilmiştir.
- Geliştirilen yöntem oluşturulan veri seti ve standart bir veri seti üzerinde test edilmiştir.

[3] https://news.google.com/
[4] http://bigdata.etu.edu.tr

- Doküman öbeklemesi konusunda diğer yöntemler ile karşılaştırma ve analiz yapılarak, geliştirdiğimiz yöntemin daha iyi performansa sahip olduğu gösterilmiştir.

2. WEB BİLGİ KAYNAKLARI VE KULLANIM ALANLARI

Web'deki veri miktarı her geçen gün artmaktadır. Web verilerinin artmasıyla birlikte Web'de yapısal bilgi kaynakları da ortaya çıkmaktadır. Bu bölümde Web'de bulunan yapısal bilgi kaynakları ve bu kaynakların kullanım alanları özetlenmektedir. Önerdiğimiz anlamsal benzerlik yöntemi ile haber öbeklendirme yaklaşımında bu yapısal bilgi kaynaklarından olan "bağlı veri" (özellikle DBpedia) kullanılmaktadır.

2.1 Web bilgi kaynakları

2.1.1 WordNet

WordNet[5] İngilizce sözcüksel bir veritabanıdır. İsimler, fiiller, sıfatlar ve zamirler kavramsal olarak eşanlamlılarıyla birlikte eş küme (synset) adlı gruplara ayrılmışlardır. Bu eş kümeler sözcüksel (lexical) linkler ile birbirlerine bağlanmıştır. Sonuç olarak birbirlerine benzeyen kelimeleri birbirlerine bağlayan bir ağ oluşturulmuştur. WordNet'in bu yapısı, doğal dil işleme (NLP) gibi bilişimsel dilbilim alanında kullanılabilmesini sağlamıştır.

WordNet'teki kelimeler arasındaki temel ilişki eş anlamlılıktır. Eş anlamlılık ilişkileri ile oluşturulan eş kümeler de birbirleriyle kavramsal ilişkiler ile bağlanmaktadır. Bu kavramsal ilişkiler anlamsal benzerlik içerse de bağlı verideki gibi çok kapsamlı değillerdir. WordNet'te 2016 itibariyle 117.000 eş küme bulunmaktadır.

2.1.2 Wikipedia

Wikipedia[6] herhangi bir kişinin içerik ekleyip düzenleyebileceği ücretsiz çevrimiçi bir ansiklopedidir. 290'dan fazla dil ve 40 milyondan fazla makaleyle internet üzerinde bulunan en büyük veri kaynaklarından birisidir. Wikipedia da bulunan makaleler içerilerinde geçen linkler ile diğer makalelere bağlanmakta ve konularına göre kategorilere ayrılmaktadır. Bu sayede varlık ismi anlam ayrışması (word sense disambiguation) veya anlamsal ilişkililik (semantic relatedness) gibi araştırma konularında oldukça yaygın bir şekilde kullanılan bir kaynak olmuştur.

[5] http://wordnet.princeton.edu/
[6] https://www.wikipedia.org/

2.1.3 Semantik Web

Anlamsal ağ, Web'in daha yapısal verilere dönüştürülmesi ve böylece yazılımlar tarafından da etkin kullanılabilmesini sağlamak üzere, standartları World Wide Web Consortium (W3C[7]) tarafından belirlenen geleceğin Web'idir. İlk olarak Tim-Berners Lee tarafından ortaya atılan bu kavramın amacı veri ağı (Web of Data) oluşturarak, Web'deki verinin bilgisayarlar tarafından da anlaşılabilir bir hale getirerek, yazılımların bu veriyi yorumlayarak kullanılabilmesini sağlamaktır [8]. Anlamsal ağda verileri tanımlamak için "Kaynak Tanımlama Çerçevesi"[8] (RDF - Resource Description Framework), "Kaynak Tanımlama Çerçevesi Şeması"[9] (RDFS - Resource Description Framework Schema) ve "Web Ontoloji Dili"[10] (OWL - Web Ontology Language) gibi protokoller/diller kullanılmaktadır.

RDF, Web'deki verilerin gösterimi ve tanımlanması için belirlenmiş basit bir modeldir [37]. RDFS ontolojileri tanımlamak için kullanılan temel özellikleri sağlamaktadır. OWL ise bu temel özellikleri genişleterek ontolojileri detaylı bir şekilde oluşturabilmek için kullanılan dildir. RDF modelinde genellikle özne (subject) - yüklem (predicate) - nesne (object) şeklinde bir üçlü (triple) gösterimi[11] kullanılmaktadır. Özne, yüklem ve nesne Web'deki birer veri kaynağını göstermektedir. Bu kaynaklar genelde Tekbiçimli Kaynak Tanımlayıcısı[12] (URI - Uniform Resource Identifier) ile ifade edilmektedir. Üçlüler, N3, XML[13] veya Turtle[14] gibi gösterim dilleri ile tanımlanabilmektedir.

Anlamsal ağdaki RDF, RDFS, OWL ile tanımlı veriler (üçlüler), SPARQL[15] (SPARQL Protocol and RDF Query Language) dili ile sorgulanabilmektedir. Bu dil veritabanlarında kullanılan SQL[16] (Structured Query Language) dilinin yapısına benzemektedir. Aşağıdaki SPARQL sorgusu örneğinde New York ile United States verileri arasındaki ilişkiler sorgulanmaktadır. Sorgunun içinde yer alan (:New_York ?predicate :United_States) üçlü örüntüsünde ?predicate bir değişken olup, herhangi bir yüklem anlamına gelmektedir.

```
SELECT DISTINCT ?predicate WHERE {
    <http://dbpedia.org/resource/New_York>
        ?predicate
    <http://dbpedia.org/resource/United_States> }
```

[7] http://www.w3.org/
[8] https://www.w3.org/RDF/
[9] https://www.w3.org/2001/sw/wiki/RDFS
[10] https://www.w3.org/OWL/
[11] https://www.w3.org/2001/sw/RDFCore/ntriples/
[12] https://www.w3.org/wiki/URI
[13] https://www.w3.org/XML/
[14] https://www.w3.org/TeamSubmission/turtle/
[15] https://www.w3.org/TR/rdf-sparql-query/
[16] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=53681

2.1.4 Bağlı veri

Bağlı veri de yine Tim-Berners Lee tarafından tanımlanan bir kavramdır. Tim-Berners Lee'ye göre bağlı veri Web'de bulunan yapılandırılmış verinin yayınlanması ve bağlı hale getirilmesidir [9, 10]. Bağlı veriyi geliştirmek için 4 kural tanımlanmıştır[17]:

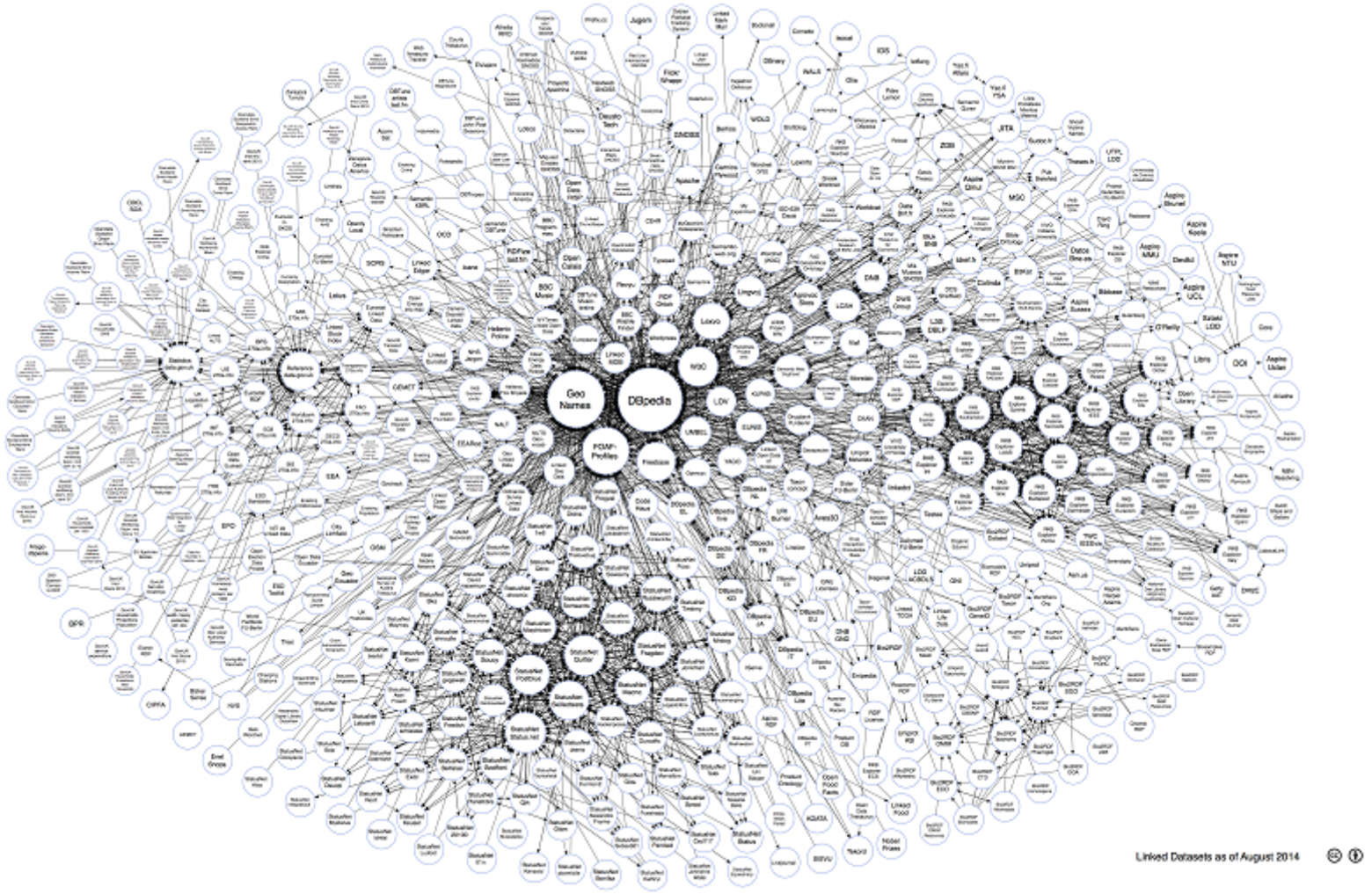
1. Bir şeyi ifade etmek için isim olarak URI kullanılması.
2. URI'ların ulaşılabilmesi için HTTP olarak belirlenmesi.
3. Bir veri kaynağının URI adresine ulaşıldığında o kaynak hakkında kullanışlı bilgilerin yine RDF gibi standartlarda sağlanması.
4. Bir kaynak üzerinden daha fazla bilgi elde edilebilmesi için diğer URI'lara bağlantı içerilmesi.

Bağlı verinin geliştirilmesi konusunda açık veri (Open Data) fikri oldukça önemlidir. Bu fikir doğrultusunda Açık Verinin Bağlanması (LOD - Linking Open Data)[18] projesi geliştirilmiştir. LOD farklı veri kaynaklarındaki yapılandırılmış verinin bağlanmasını amaçlamaktadır. 2014 itibariyle LOD projesinde 1014 veriseti bulunmaktadır[19]. LOD bulut diagramı Şekil 2.1'de gösterilmektedir. Buradaki verisetlerinin en büyüklerinden bazıları DBpedia[20], Freebase[21] ve YAGO[22]'dur.

DBpedia: DBpedia projesi, en geniş çevrimiçi ansiklopedi olan Wikipedia[23] sayfalarında yer alan yapılandırılmış verilerin otomatik olarak çekilerek Web'de yayınlanması amacıyla başlatılmıştır [3]. Şekil 2.1'de görüldüğü gibi Açık Bağlı Veri'de bulunan verisetleri arasında en büyüklerinden ve en merkezde olanlardan birisidir. Temmuz 2016 itibariyle DBpedia verisetinde 125 farklı dilde 4,58 milyon varlık (resource), 3 milyar civarında RDF üçlüsü bulunmaktadır.

Freebase: LOD projesinde bulunan bir diğer veriseti de Freebase'dir. Freebase'in amacı insan bilgisini yapılandırarak bir uygulama programlama arayüzü (API) sayesinde kullanıma sunmaktır [11]. Freebase 2010 yılında Google tarafından satın alınarak Knowledge Graph projesine dahil edilmiştir [43]. 2016 yılında ise API durdurularak proje sonlandırılma kararı alınmıştır. Veri ise indirilebilir durumdadır [24].

[17] https://www.w3.org/DesignIssues/LinkedData.html
[18] https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
[19] http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/
[20] http://wiki.dbpedia.org/
[21] http://datahub.io/dataset/freebase
[22] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/
[23] https://www.wikipedia.org/
[24] https://developers.google.com/freebase/



Şekil 2.1: LOD bulut diagramı[a]

[a]http://lod-cloud.net/versions/2014-08-30/lod-cloud.png

YAGO: YAGO (Yet Another Great Ontology) Max-Planck Enstitüsü tarafından WordNet ve Wikipedia kaynakları kullanılarak geliştirilen bir bilgi tabanıdır [67]. Veri setinde Temmuz 2016 itibariyle 10 milyondan fazla varlık ve bu varlıklarla ilgili 120 milyondan fazla nitelik bulunmaktadır.

2.2 Web bilgi tabanları kullanım alanları

2.2.1 Açık bilgi çıkarma

Bilgi çıkarma, yapılandırılmamış veri üzerinden otomatik olarak yapılandırılmış veri elde edilmesi işlemidir. Örneğin bir metin içerisinde geçen konum bilgilerinin bulunması bilgi çıkarma işlemi ile yapılmaktadır. Açık bilgi çıkarma da önceden bir kelime haznesine sahip olmadan anlamsal ilişkiler çıkarma işlemidir. Bu alanda önemli birçok çalışma yapılmıştır [5, 17] ve bu çalışmaların bazılarında Wikipedia gibi Web bilgi tabanları kullanılmaktadır [74, 75]. Çıkarılan ilişkiler özne, yüklem ve nesne üçlüleri halinde bulunmaktadır. Bu ilişkiler NLP, soru cevaplama gibi başka araştırma alanlarında da sıklıkla kullanılmaktadır.

2.2.2 Soru cevaplama

Soru cevaplama, insanlar tarafından sorulan soruların makineler tarafından anlaşılıp cevaplanması problemini temel alan bir araştırma alanıdır ve bu alanda yapılmış birçok çalışma bulunmaktadır [59]. Soru cevaplamada genel olarak ilk adımda sorunun içerisinde geçen "Kim", "Nerede" ve "Ne zaman" gibi soru kelimeleri bulunur ve sorunun türü anlaşılır, bu sayede verilecek cevabın kişi, konum veya zaman türünde mi olacağı belirlenir. Burada "Ne", "Hangi" gibi kelimelerden tür elde edilemeyeceği için sorunun anlaşılması amacıyla soruda geçen diğer anahtar kelimeler WordNet, DBpedia ve diğer bağlı veri kaynakları gibi dış veri kaynakları kullanılarak sorunun türü bulunmaktadır [54, 70, 71, 15]. Sorunun türü anlaşıldıktan sonra soruda geçen anahtar kelimeler bir korpus üzerinde aratılarak sorunun türüne uygun olan bilgiler bulunur.

Soru cevaplama arama motorları, Siri[25] gibi akıllı kişisel asistan uygulamaları ve IBM Watson[26] gibi yapay zeka uygulamalarında kullanılmaktadır.

[25] http://www.apple.com/tr/ios/siri/
[26] http://www.ibm.com/watson/

2.2.3 Doküman öbekleme

Öbekleme, verilerin gruplara (cluster) ayrıldığı gözetimsiz (unsupervised) bir öğrenme işlemidir [27]. Veri madenciliği ve makine öğrenme gibi çeşitli alanlarda, veri setlerinden anlam çıkarma gibi uygulamalarda sıkça kullanılır [7, 62]. Doküman öbekleme ise dokümanların gruplanması işlemidir.

Doküman öbeklemesindeki amaç aynı grup içerisinde olan dokümanların içerik olarak birbirlerine olan benzerliği yüksek, diğer gruplardaki dokümanlara olan benzerlikleri ise düşük olmasıdır [2]. Web'deki verinin hızla artmasıyla doküman öbeklemenin de önemi artmıştır. Arama motorları gibi bilgi çekme (information retrieval) alanındaki uygulamalarda daha anlamlı sonuçlar ortaya koymak için doküman öbekleme yöntemleri kullanılmaktadır [65].

Doküman öbeklemesinin özel bir alt konusu ise haber öbeklemesidir. Haber öbeklemesindeki amaç birbirine benzer haber dokümanlarının gruplandırılmasıdır. Haberler bilgiye ulaşma konusunda en önemli kaynaklardan birisi olduğu için, haber öbeklemesi de oldukça önemli bir konudur. Bu konuda en bilinen sistemlerden birisi olan Google News, birbiriyle ilgili olan haberleri gruplayarak kullanıcıya sunabilmektedir.

3. İLGİLİ ÇALIŞMALAR

Haber makalelerinin öbeklenmesi, doküman öbeklemesinin bir alt konusudur. Bu bölümde, haber ve doküman öbeklemesi konusunda kullanılan yöntemler, yapılan çalışmalar ve yaklaşımlar anlatılmaktadır.

3.1 Öbekleme Yöntemleri

Doküman öbeklemesi konusunda çeşitli yöntemler bulunmaktadır. Bu yöntemlerden en yaygın kullanılanları hiyerarşik öbekeleme ve k-means öbeklemedir [66]. Hiyerarşik öbekleme bağlantı tabanlı bir öbekleme yöntemidir. Hiyerarşik öbeklemede nesneler aralarındaki uzaklığa göre birbirlerine bağlanarak bir hiyerarşi oluştururlar. Buna göre bu hiyerarşide birbirlerine yakın olan nesneler birbirlerine daha benzerlerdir [32]. K-means öbekleme ise bölme tabanlı bir öbekelemedir. Nesnelerin öbeklerin merkezlerine olan uzaklıkları hesaplanır ve nesne kendisine en yakın olan öbeğe ait olur [2].

3.1.1 K-means öbekleme

K-means öbeklemesinde elemanlar K adet öbeğe bölünmektedir. $X = \{x_1, x_2, ..., x_n\}$ şeklinde n adet d boyutlu vektörlerden oluşan ve $C = \{c_1, c_2, ..., c_k\}$ şeklinde k adet öbeğe bölünecek bir set olsun. K-means algoritması, öbekteki noktalar ile öbeğin ortalaması arasındaki karesel hatayı (squared error) minimize edecek şekilde bir bölme işlemi yapar. μ_k, c_k öbeğinin ortalaması olsun. Bu durumda μ_k ile c_k öbeğindeki noktaların karesel hatası aşağıdaki şekilde hesaplanır [26]:

$$J(c_k) = \sum_{x_i \in c_k} ||x_i - \mu_k||^2 \tag{3.1}$$

K-means algoritmasındaki amaç bütün öbeklerdeki karesel hatanın toplamını minimize etmektir:

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} ||x_i - \mu_k||^2 \tag{3.2}$$

K-means algoritmasının adımları aşağıda sırasıyla verilmiştir [28]:

1. K tane başlangıç öbeği seç, 2. ve 3. adımları öbek aitlikleri stabil hale gelene kadar tekrarla.

Çizelge 3.1: Uzaklık ölçüteri[a]

Uzaklık Ölçütü	Formül
Öklit	$d_E(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Manhattan	$d_M(x,y) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$
Ortalama Karesel Hata	$d_{MSE}(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$
Canberra	$d_c(x,y) = \sum_{i=1}^{n} \frac{\lvert x_i - y_i \rvert}{\lvert x_i \rvert + \lvert y_i \rvert}$
Kosinüs	$d_{cos}(x,y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$

[a]http://numerics.mathdotnet.com/Distance.html

2. Her elemanı merkezi kendisine en yakın olan öbeğe ekle.

3. Yeni öbek merkezlerini hesapla.

Bu adımlarda öbek merkezleri ile elemanlar arasındaki uzaklıkların hesaplamasında kullanılan yöntem önemlidir. En yaygın kullanılan uzaklık hesaplama yöntemi Öklit uzaklığıdır. Uzaklık hesaplama yöntemlerinden bazıları Çizelge 3.1'de verilmiştir.

3.1.2 Hiyerarşik öbekleme

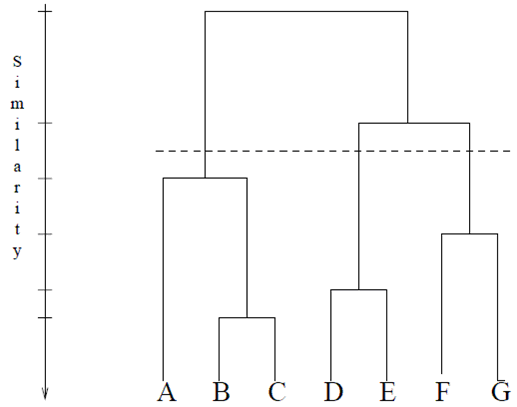
Hiyerarşik öbekleme bilgi çekme alanında tercih edilen öbekleme yöntemlerinden biridir [29]. Hiyerarşik öbeklemede amaç bir hiyerarşi öbeği oluşturmaktır. Bu hiyerarşi kullanılarak 2 farklı şekilde öbekleme yapılmaktadır [19]:

- Toplayıcı (agglomerative): Toplayıcı öbeklemede hiyerarşi aşağıdan yukarı (bottom up) bir şekilde oluşturulur. Başta her bir eleman birer öbek halindedir. Bu öbekler diğer öbekler ile birleşerek yukarıya doğru çıkarak yeni öbekler meydana getirir.

  Toplayıcı öbeklemede girdi olarak N*N'lik bir uzaklık matrisi verilir. Algoritmanın adımları şu şekildedir [32]:

  1. Sadece 1 elemanı olan öbekler yarat.
  2. Benzerliği en yüksek olan öbekleri birleştirerek yeni bir öbek oluştur.
  3. Oluşan yeni öbek ile diğer öbekler arasındaki uzaklığı hesapla.
  4. 3 ve 4 numaralı adımları bütün elemanlar tek bir öbek oluşturana kadar tekrarla.

- Bölücü (divisive): Bölücü öbeklemede hiyerarşi yukarıdan aşağı (top down) bir şekilde oluşturulur. Burada gruplanmış şekildeki öbekler aşağı doğru farklı gruplara bölünür ve en sonunda her eleman kendi öbeğinde yer alır [33].



Şekil 3.1: Dendrogram

Hiyerarşik öbeklemede oluşturulan hiyerarşi bir ağaç yapısı olan dendrogram şeklinde gösterilir. Bu dendrogram Şekil 3.1'deki gibi istenen yükseklikte kesilir ve bunun altında kalan öbeklere ayrılır.

3.2 Döküman ve Haber Öbeklemesi

Haber öbekleme, doküman öbeklemenin bir çeşididir. Birbirleriyle ilgili olan haber makalelerinin öbeklere ayrılmasıyla ilgilenmektedir. Bu alanda bir çok çalışma yapılmıştır [13, 12, 48, 47, 53, 58]. Bouras vd. çalışmalarında k-means algoritmasını geliştirerek ve WordNet bilgi tabanını kullanarak BBC, CNN gibi haber portallarından topladıkları dokümanları kategorilerine göre öbeklemişlerdir [13, 12].

Montalvo vd. ise çalışmalarında farklı dillerdeki haber makalelerini öbeklemeyi amaçlamıştır [48, 47]. Bu çalışmalarda İspanyolca ve İngilizce haber makaleleri öbeklenirken varlık isimlerinden faydalanılmıştır.

Radev vd. NewsInEssence isimli projelerinde verilen bir haberin konusuyla ilgili haberleri gerçek zamanlı olarak Web üzerinde bularak öbeğe eklemektedir. İlgili haberleri ararken haberin anahtar kelimelerini bulmakta ve bu kelimeler üzerinden aramayı gerçekleştirmektedir [58].

Haber öbeklemesi doküman öbeklemesinin bir alt türü olduğu için, doküman öbeklemesi alanında kullanılan yöntemler haber öbeklemesinde de kullanılabilmektedir. Sonraki bölümlerde bu alanda yapılan çalışmalar yaklaşımlarına göre gruplandırarak özetlendirilmiştir.

3.3 Kelime Çantası

Kelime çantası modeli dokümanları içlerinde geçen kelimeler ve bu kelimelerin sıklıklarını kullanarak oluşturulan vektörler şeklinde göstermeye yarayan bir modeldir. Doküman öbekleme konusunda bir çok çalışmada kullanılmıştır [4, 24]. Bu yaklaşımda kelimelerin sıklıkları genelde TF-IDF (term frequency - inverse document frequency) denilen bir ağırlıklandırma yöntemi ile hesaplanmaktadır. TF-IDF kelimeleri bulundukları dokümanın içindeki sıklığı (term frequency) ve bütün dokümanların içerisindeki sıklığı (inverse document frequency) değerlendirerek aşağıdaki formülle hesaplar [60]:

$$tfidf_{ij} = \frac{f_{ij}}{\sum_{j=1} f_{ij}} \times log(\frac{|D|}{|\{d_i|t_j \in d_i \in D\}|}) \tag{3.3}$$

Formül (3.3)'te d_i dokümanının içindeki t_j kelimesinin sıklığı f_{ij} ile gösterilmektedir. d_i içerisindeki bütün kelimelerinin toplam sıklıklarına bölünür. Veri seti D'nin içerisindeki toplam doküman sayısı $|D|$ ve t_j kelimesini içeren doküman sayısı $|\{d_i|t_j \in d_i \in D\}|$ şeklindedir.

Agrawal vd. dokümanları öbekleme yaparken kelime çantası yaklaşımını kullanmaktadır [2]. Dokümanları TF-IDF ile ağırlıklandırılmış kelime vektörleri halinde göstermektedirler. Öbekleme yaparken de uzaklıkları kosinüs benzerliği ile hesaplayan k-means yöntemi kullanılmaktadır. Öbek sayısı girdi olarak verilmek yerine otomatik olarak hesaplanmaktadır. Kosinüs benzerlik matrisini her tekrarda hesaplamak yerine sadece en başta hesaplayacak şekilde k-means algoritması modifiye edilmiştir. Verdikleri sonuçlara göre kendi yöntemleri F1 puanı bakımından normal k-means algoritmasından bir miktar daha iyi sonuç vermektedir.

Kelime çantası yaklaşımını kullanan bir diğer çalışma da Forsati vd. tarafından gerçekleştirilmiştir [18]. Vektörler şeklinde gösterilen dokümanlar TF-IDF ile ağırlıklandırılmıştır. Öbekleme yöntemi olarak kullanılan k-means algoritmasının performansını arttırmak için "harmony search" optimizasyon yöntemi kullanılmıştır. "Harmony search" yöntemi ilk öbek merkezlerini bulmak için kullanılmıştır. Standart k-means algoritmasına göre daha iyi sonuçlar verdiği çalışmada gösterilmiştir.

Kelime çantası yaklaşımları en bilinen yöntemlerden biri olmasına rağmen doküman öbeklemede yeterli değildir. Bu yaklaşımda iki anlamlılık (ambiguity) ve eş anlamlılık (synonymy) problemleri vardır. Birden fazla anlama gelen bazı kelimeler aynı kelime olarak gözükmekte iken, aynı anlama gelen iki kelime ise yazılışları farklı olduğu için benzerlik hesabını olumsuz etkilemektedir. Bunun dışındaki bir diğer problem ise kelimelerin dokümanların içerisinde bulundukları yerlerin benzerlik hesabına katılmamasıdır.

3.4 Gizli Anlamsal Analiz

Gizli anlamsal analiz (Latent Semantic Analysis) (LSA) dokümanlar arasındaki benzerliği hesaplamak için dokümanlardaki kavram ve terimleri çıkaran istatistiksel bir doğal dil işleme yöntemidir. LSA yönteminde benzer anlamlara sahip olan kelimelerin, metinin benzer bölümlerinde yer aldığı düşünülmektedir. Tekil değer ayrışımı (Singular Value Decomposition - SVD) yöntemi kullanılarak bir sıklık matrisi oluşturulmaktadır. Bu matriste her bir sıra bir kelimeyi, her bir sütun ise bir paragrafı ifade etmektedir. Benzer içeriklerde kullanılan kelimeler SVD uygulandıktan sonra birleşmektedir. Bu sayede farklı terminoloji kullanan fakat benzer anlamlarda olan dokümanlar, bu gösterim şeklinde birbirine yakınlaşmaktadırlar [39]. LSA kullanılarak doküman benzerliği ve sınıflandırılması alanlarında yapılan bir çok çalışma bulunmaktadır [22, 38, 42, 68, 76].

Hofmann vd. LSA yöntemini modifiye ederek olasılıksal LSA (PLSA) isminde bir yöntem geliştirmişlerdir [22]. Bu yöntemde standart LSA yönteminden farklı olarak bir gizli sınıf modelinden (latent class model) elde edilen karışık ayrışma uygulanmaktadır.

3.5 Bilgi Tabanları ve Bağlı Veri

Döküman öbekleme alanındaki son zamanlardaki çalışmalar WordNet, Wikipedia ya da DBpedia gibi bilgi tabanları kullanmaya başlamışlardır. Bu yaklaşım yazılış olarak aynı olma şartı gerektiren kelime çantası ya da LSA yöntemlerinden farklıdır. Bilgi tabanı temelli yaklaşımlar daha çok dökümanların anlamsal benzerliğine odaklanmaktadır. Anlamsal benzerlik dökümanlar arasındaki benzerliği hesaplarken yazılış olarak benzerlik dışında, dökümanlardaki kavramların veya varlık isimlerinin birbirleriyle karşılaştırılmasını da hesaba katmaktadır. Örneğin bir insan haberleri okurken, George W. Bush hakkında olan bir haber ile Barack H. Obama hakkındaki bir haberin birbirleriyle alakalı (Amerika Birleşik Devletleri başkanları) olabileceğini anlayabilmektedir. Ama kelime çantası yaklaşımı kavramlar arasındaki ilişkiler yerine sadece kelimeler ve yazılışlarını dikkate aldığı için bu durumda benzerliği ortaya çıkaramamaktadır. Anlamsal benzerlik hesaplamasında dökümanlardaki bu şekildeki kavramlar arasında benzerlikleri anlayabilmek ve ilişkilerini bulabilmek için dış referans kaynaklarına ihtiyaç vardır. Verilen örnek için DBpedia gibi bir bilgi tabanı Bush ve Obama için yapılandırılmış bilgi sağlayabilmekte ve ikisinin de Amerika Birleşik Devletleri başkanlarından oldukları bilgisi sayesinde bu iki varlık ismini ilişkilendirebilmektedir.

Bilgi kaynakları kullanılarak anlamsal benzerlik hesaplama alanındaki çalışmalar ilk olarak İngilizce kaynak olan WordNet kullanılmasıyla başladı. WordNet aynı anlamlara gelen kavram setlerinin gruplar halinde bulunduğu İngilizce bir sözcüksel veritabanıdır [45]. İngilizce diliyle sınırlı olduğu için sadece bu dildeki benzerlik hesaplamalarında

işe yaramaktadır. Daha sonraki araştırmalar daha genel bir bilgi kaynağı olan Wikipedia'yı, en büyük internet ansiklopedisini, kullanmaya başladılar. Bu durumda benzerlik hesaplamasına Wikipedia'daki kavramlar, kategoriler ve sayfalar arasındaki bağlantılar katıldı. Daha yakın zamanda gerçekleştirilen çalışmalarda ise daha gelişmiş bilgi kaynakları, çoğunlukla DBpedia veya Freebase gibi bağlı veri kaynakları kullanılmaya başlandı. DBpedia, Açık Bağlı Veri Bulutu'nda (LOD) bulunan en büyük veri setlerinden birisidir. Bu veri seti Wikipedia verisinden elde edilerek yapılandırılmış ve çeşitli çalışmalarda kullanılmıştır. Anlamsal benzerlik hesaplamasında bağlı veriden faydalanan yöntemler sadece varlık isimleri veya konseptleri değil, aynı zamanda varlık türleri, kategorileri gibi daha karmaşık ilişkileri de kullanmaktadır.

3.5.1 WordNet

Naik vd. [50] anlamsal doküman öbeklendirilmesi konusundaki yöntemler hakkında bir derleme yayınlamıştır. Bu derlemede değerlendirilen yöntemler ontoloji tabanlı, anlamsal çizge tabanlı, sık tekrarlayan kavram tabanlı, LSA tabanlı ve WordNet tabanlı olarak kategorilere ayrılmıştır. WordNet ve ontoloji tabanlı yöntemler, bir dış kaynak kullanarak benzerlik bulma açısından bu tez çalışmasında sunulan yönteme benzerlik gösterse de, son zamanlarda kullanılan bağlı veri kaynaklarıyla ilgili çalışmalar bu derlemede yer almamaktadır.

Kim vd. de doküman öbeklemesi konusunda WordNet'ten faydalanmıştır [35]. Bu çalışmada anlamsal özellik matrisleri oluşturmak için terim doküman sıklığı matrisi (term document frequency matrix) üzerinde negatif olmayan matris faktorizasyonu (NMF[27]) uygulanmıştır. Öbek terimlerinin bulunmasında bu anlamsal özellik matrisleri kullanılmıştır. Terimlerin ağırlıkları WordNet eşanlamlılarından faydalanılarak karşılıklı terim bilgisi (term mutual information - TMI) kullanılarak hesaplanmıştır. Daha sonra dokümanlar arasındaki kosinüs benzerlikleri öbek terimleri ve terim ağırlıkları kullanılarak elde edilmiştir. Sonuçlarına göre NMF yöntemi uygulanırken WordNet'ten faydalanmak performansı artırmaktadır.

Bouras vd. iki anlamlılık ve eş anlamlılık sorunlarını aşmak için WordNet ile kelime çantası modelini zenginleştiren W-k means isminde bir yöntem geliştirmiştir [12]. Bu yöntemde WordNet kullanılarak dokümanlardaki her bir terim için kapsayıcı terim çizgeleri oluşturulmuştur. Daha sonra bu kapsayıcı terimlerin ağırlıkları hesaplanarak bulunan anahtar kelimeler dokümanların çantalarına eklenmiştir. Doküman çantaları k-means algoritması kullanılarak öbeklenmiştir. Oluşan öbeklerin etiketleri, her öbekte bulunan en önemli (en sık bulunan) anahtar kelime seçilmiştir. Bu yöntemde eş anlamlılık ve iki anlamlılık problemleri WordNet kullanılarak aşılmıştır, ama terimler arasındaki ilişkiler

[27] https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

dikkate alınmamıştır.

Wei vd. dokümanların anlamlarını WordNet kullanılarak bulmuştur [73]. Kelime-anlam ayrımı (Word Sense Disambiguation - WSD) prosedürü ile dokümanlardaki her bir kelimenin anlamı bulunmuştur. Daha sonra da bu anlamlar arasındaki eşanlamlılık, iki anlamlılık gibi ilişkiler ile sözcük zincirleri oluşturulmuştur. Oluşturulan sözcük zincirleri bisecting k-means yöntemi ile öbeklenmiştir. WordNet ile WSD uygulanan yöntemin temel yöntemlerden daha iyi sonuç verdiği belirtilmektedir.

WSD için WordNet kullanan bir başka çalışma ise Patil vd. tarafından gerçekleştirilmiştir [55]. Bu çalışmada WordNet kullanılarak kelimelerin kategorileri çıkarıldıktan sonra, TF-IDF ile ağırlıklandırılarak her bir doküman için birer anahtar terim seti elde edilmiştir. Sadece konuyla alakalı terimleri almak için TF-IDF ağırlıkları için bir eşik değeri (threshold) kullanılmıştır. Herhangi bir öbekleme sonucu verilmemesine rağmen bu çalışmada ortaya konan yöntemin öbekleme doğruluğunu arttırabileceği belirtilmiştir.

3.5.2 Wikipedia

Referans bilgi kaynakları kullanılarak benzerlik hesaplama alanında en kapsamlı kaynaklardan birisi Wikipedia'dır. Bu alanda büyük bir yenilik getirmiş olan çalışmayı Gabrilovich ve Markovitch gerçekleştirmiştir [20]. Dokümanları Wikipedia kategorilerini ağırlıklandırılmış vektörler olarak temsil etmişlerdir. Belirgin anlamsal analiz (Explicit Semantic Analysis - ESA) dedikleri yöntemde kelimelerin ilgili Wikipedia makalelerindeki TF-IDF puanlarını kullanmışlardır. Bu yöntemde Wikipedia kategorileri kullanılmıştır fakat bu kategoriler arasındaki ilişkiler hesaplamaya katılmamıştır.

Jiang vd. [31, 30] anlamsal kavram benzerliği konusunda Wikipedia kategori yapısına dayalı çeşitli yöntemler sunmuştur. Bu yöntemlerde, bu tez çalışmasına benzer şekilde, kategori ağacındaki en düşük ortak ata (lowest common ancestor) kullanılmaktadır. Bizim çalışmamıza göre eksik yanı ise sadece Wikipedia kategorileriyle sınırlı kalmasıdır. Ayrıca geliştirdikleri yöntemler doküman öbeklendirilmesi gibi bir görevde kullanılmamış ve test edilmemiştir. Bunun yerine kullanıcı değerlendirmesi yapılmış, bir kavram listesi üzerinden değerlendirmeler sunulmuştur.

3.5.3 Bağlı veri

Günümüzde en kapsamlı "yapısal" referans bilgi kaynağı "bağlı veri" denilebilir. Ve bu alanda en kapsamlı genel bilgi kaynağı DBPedia'dır. Bağlı veriler de yakın zamanda makina öğrenme görevlerinde sıklıkla kullanılmaya başlanmıştır. Biz de bu tez çalışmasında bağlı veri ve DBpedia kullanarak doküman benzerlik ölçümü yapıyoruz. Bu alandaki bazı

ön çalışmalar aşağıda değerlendirilmiştir.

Zhang vd. [77] doğal dil işleme alanında anlamsal ilişkililik konusunda bir derleme gerçekleştirmiştir. Bu derlemede anlamsal ilişkililik veya benzerlik konusunda bağlı verinin bir dış kaynak olarak kullanılmasının büyük bir potansiyele sahip olduğunu belirtilmiştir. O zamandan beri gerçekleştirilen çalışmaların çoğunda bağlı veri kaynağı olarak DBpedia kullanılmıştır. Bu çalışmalardaki benzerlik hesaplamalarında dikey (kategori veya tür hiyerarşileri) veya yatay (konsept veya varlıklar arasındaki DBpedia özellikleri gibi ilişkiler) bağlantılar kullanılmıştır. Aşağıda bu çalışmalar değerlendirilmektedir.

Oto [53] tez çalışmasında varlık isimleri ve türlerinin ilişkileri kullanılarak doküman benzerliği hesabı yapan anlamsal bir yöntem geliştirmiştir. Varlıklar arasındaki ilişkiler DBpedia kullanılarak bulunmakta ve iki varlık arasında isim, tür ve bulunan ilişkiler kullanılarak bir benzerlik hesaplaması yapılmaktadır. Geliştirilen yöntemde tür olarak sadece YAGO türleri kullanılmıştır ve bu tez çalışmasındaki gibi bir benzerlik yöntemi kullanılmamış, sadece aynı kategorilere sahip olunup olunmadığına bakılmıştır. Geliştirilen yöntem Google News'ten toplanan haber makalelerinde test edilmiş fakat standart veri setleri testlerde kullanılmamıştır.

Hulpus vd. DBpedia bağlı veri kaynağını kullanan çizge tabanlı bir etiket bulma yöntemi geliştirmiştir [25]. Bu tez çalışmasında da faydalanılan DBpedia kavramları kullanılarak kavram çizgeleri oluşturulmuştur. Bunun dışında özdeğer (eigenvalue) tabanlı WSD uygulanarak her bir kavram için kelime anlam çizgeleri (word-sense graphs) oluşturulmuştur. Daha sonra çizge merkeziyet ölçümleri yapılarak dokümanların konuları bulunmuştur. Hulpus vd.'e göre iyi bir konu etiketi, çizgenin merkezindeki bir düğümde olmalıdır.

DBpedia kavramları Szczuka vd.'nin çalışmasında kelime çantası yaklaşımıyla birlikte kullanılmıştır [69]. Dokümanlar DBpedia'dan bulunan kavramları kullanılarak vektörler haline getirilmiştir. Kelimeler yerine kavramlar kullanıldığı için bu yönteme kavram çantası (bag of concepts) denilmiştir. TF-IDF ile ağırlıklandırılan bu vektörler arasındaki kosinüs benzerliği hesaplandıktan sonra toplayıcı hiyerarşik öbekleme (agglomerative hierarchical clustering) yöntemi ile öbeklenmiştir. Kelime çantasına göre daha iyi sonuçlar verildiği belirtilmektedir.

Leal vd. [40] DBpedia tür hiyerarşilerinden elde edilen ontoloji yollarını kullanarak konseptler arasında anlamsal benzerlik hesabı yapan bir yöntem geliştirmiştir. Geliştirdikleri yöntem Shakti adı verilen bir araca dönüştürülerek haber önerme sisteminde test edilmiştir. Ancak değerlendirmeleri sınırlı kalmış ve standart bir veri seti üzerinde performansı değerlendirilmemiştir.

Zhu ve Iglesias [78] anlamsal benzerlik konusunda hem korpus tabanlı hem de bilgi tabanı odaklı yaklaşımları değerlendirmiş ve bazı mevcut benzerlik metriklerini karşılaştır-

mıştır. Ayrıca DBpedia çizgesindeki kavramlar arasında anlamsal benzerlik hesabı yapan bir yöntem geliştirmişlerdir. Hem konseptler arasındaki en kısa yol, hem de en düşük ortak kapsayıcının (lowest common subsumer) bilgi içeriği (IC) hesaba katılmıştır. Bilgi içeriği, bir konseptin korpus üzerindeki önemini ve sıklığını ölçmektedir. Bu önerilen yöntem diğerleriyle karşılaştırılmamış ve değerlendirme olarak sadece kelime benzerlik veri setleri kullanılmıştır. Gelecekte çalışılabilecek bir konu olarak doküman öbeklendirilmesi belirtilmiştir.

Ni vd. [51] bağlı veri üzerinde konsept çizge benzerliği konusunda bir yöntem geliştirmişlerdir. Bu yaklaşımda ikili doküman benzerliği, her doküman için en iyi eşleşen ikili konseptlerin benzerliği kullanılarak bulunmaktadır. Buradaki konsept benzerliği çizge merkezliliği kullanılarak hesaplanmaktadır. Değerlendirme olarak LP50 veri seti kullanılmış ve ESA yönteminden daha iyi sonuç verdiği belirtilmiştir.

Meymandpour vd. tavsiye sistemlerinde bağlı veriyi kullanan, bilgi içeriği (Information Content) tabanlı anlamsal bir benzerlik yöntemi sunmaktadır [44]. Meymandpour vd.'e göre sıklığı daha az olan özellikler daha fazla bilgi içermektedir. Özellik olarak bağlı veri kaynakları arasındaki ilişkiler seçilmiştir. İki kaynak arasındaki benzerlik bu kaynakların bölüntülenmiş bilgi içeriklerine (PIC) göre hesaplanmıştır. Eğer paylaşılan özelliklerin PIC değeri yüksek ise, bu kaynakların benzer olduğu anlaşılmaktadır. Sonuçlarına göre bağlı verinin benzerlik hesabında kullanılması kök ortalama karesel hatayı (RMSE) düşürmektedir.

Schuhmacher vd. [61] doküman temsili için DBpedia veri setini kullanan çizge tabanlı anlamsal bir model sunmuştur. Ancak yöntemleri ESA yöntemi kadar iyi bir sonuç vermemiştir. Bu çalışmada geliştirilen "çizge düzenleme uzaklığı" (GED) modelini Paul vd. [56] daha sonra genişletmiştir. Paul vd. dokümanlardaki her bir varlık için DBpedia konseptleri kullanarak hiyerarşik ve enine genişleterek birer çizge oluşturmuştur. Daha sonra bu varlıklar arasındaki hiyerarşik benzerlik, bu çizgeler üzerindeki en düşük ortak atanın ve çizgenin köküne olan uzaklığı kullanılarak hesaplanmaktadır. Enine benzerlik ise varlıklar arasındaki direk veya dolaylı (arada başka bir varlık da bulunan) ilişkiler sayılarak hesaplanmaktadır. Yaptıkları testlerin sonuçlarına göre geliştirdikleri yöntem ESA yönteminden daha iyi sonuç vermektedir. Bu tez çalışmasında geliştirilen yöntem de Paul vd.'nin hiyerarşik benzerlik yöntemine benzemektedir. O yöntemden farklı olarak, bu tez çalışmasında benzerlik hesabı yapılırken varlıklar IDF yöntemi kullanılarak ağırlıklandırılmaktadır. Bu sayede varlıkların önemi de dikkate alınmıştır. Bunun dışında kategoriler de ağırlıklandırılarak öbekleme performansının artması sağlanmıştır.

Nunes vd. [52] dokümanlarda geçen varlıklar arasındaki anlamsal bağlantıları temel alan bir doküman benzerlik yöntemi geliştirmişlerdir. Varlıklar ve dokümanlar arasında yol tabanlı bir benzerlik hesabı yapmaktadırlar. Varlıklar arasındaki ilişkiler bulunurken DB-

Çizelge 3.2: Doküman öbeklendirilmesinde dış bilgi kaynağı kullanan yayınlar

Yayın	**Bilgi Tabanı**	**Yöntem**	**Veri setleri**
Song vd.[63]	WordNet	Latent Semantic Indexing (LSI), Genetic Algorithm	Reuters-21578
Kim vd.[35]	WordNet	Non-negative Matrix Factorization (NMF)	20Newsgroup
Bouras vd.[12]	WordNet	W-kmeans	–
Li vd.[41]	WordNet	Frequent Word Meaning Sequences	Reuters-21578
Wei vd.[73]	WordNet	Sözcüksel Zincirleri	Reuters-21578
Patil vd.[55]	WordNet	WordNet Eşanlamlıları	Reuters-21578, 20Newsgroup
Hu vd.[23]	Wikipedia	Wikipedia Eşanlamlıları	Reuters-21578, OHSUMED
Jiang vd.[31, 30]	Wikipedia	Wikipedia Kategorileri	–
Kim vd.[36]	Wikipedia	Wikipedia Konseptleri	–
Gabrilovich vd.[20]	Wikipedia	Explicit Semantic Analysis (ESA)	Lee50
Szczuka vd.[69]	DBpedia	Bag-of-Concepts	Sci. papers
Hulpus vd.[25]	DBpedia	Graph Centrality	BBC News, ...
Dostal vd.[16]	DBpedia	PageRank	20Newsgroup
Schuhmacher vd.[61]	DBpedia	Graph Edit Distance	LP50
Paul vd.[56]	DBpedia	Hiyerarşik ve Yol Tabanlı	Lee50
Nunes vd.[52]	DBpedia	Yol Tabanlı	USAToday news
Cano vd. [14]	DBpedia, Freebase	Konsept Çizgeleri	Twitter

pedia kullanılmıştır fakat kategorilerden faydalanılmamıştır. Geliştirdikleri yöntem doküman öbeklemesinde kullanılmamış ama kitle kaynaklı bir çalışma ile oluşturulan ikili doküman benzerlikleri ile karşılaştırılmıştır.

Doküman öbeklendirilmesinde veya benzerliğinde dış bilgi kaynağı kullanan yöntemler Çizelge 3.2'de özetlenmiştir.

4. BAĞLI VERİ KAYNAKLARI KULLANILARAK HABERLERİN ÖBEKLENDİRİLMESİ

Bu tez çalışmasında haber veya doküman öbeklendirilmesi için geliştirilen yöntem, kelime çantası veya LSA gibi yöntemlerinin aksine, bağlı veri dış kaynağını kullanan anlamsal bir benzerlik hesaplaması yapmaktadır. Wikipedia [20, 23, 46, 64] ve WordNet [63, 35, 12, 73, 55] gibi bilgi kaynakları kullanan önceki çalışmalar bu konuda umut verici sonuçlar ortaya koymuştur. Yeni geliştirilen "bağlı veri" kaynakları ise anlamsal ağ ve ontoloji prensiplerini temel aldığı için anlamsal benzerlik hesaplama konusunda daha zengin bilgiler sunmaktadır.

Bağlı veri kaynakları kullanılarak haberlerin öbeklendirilmesi için geliştirilen yöntemin adımları aşağıdaki gibidir:

1. Dokümanlarda geçen varlık isimlerinin ve bu varlıkların bağlı veri kaynaklarının bulunması
2. Doküman ikilileri arasındaki anlamsal benzerliğin hesaplanması ve bu benzerlikler kullanılarak bir uzaklık matrisi oluşturulması
3. Hiyerarşik öbekleme yöntemiyle uzaklık matrisi üzerinden dokümanların öbeklendirilmesi

İlk adımda dokümanlar çözümlenerek içerdikleri varlık isimleri ve bu varlıkların bağlı veri kaynaklarına olan bağlantıları bulunmaktadır. Burada önemli olan nokta varlıkların bağlı veri kaynaklarının doğru bir şekilde bulunmasıdır. Örneğin eğer bir dokümanda "Apple" ifadesi geçiyorsa, bunun elma (http://dbpedia.org/page/Apple) mı yoksa teknoloji şirketi (http://dbpedia.org/page/Apple Inc.) mi anlamına geldiğinin belirlenmesi gerekmektedir. Varlık ismi anlam ayrımı (named entity disambiguation) [21], bilgi çıkarma alanında önemli bir konudur, fakat bu tez çalışmasının bir konusu değildir. Bu çalışmada varlık ismi bulma ve anlam ayrımı için daha önceden geliştirilmiş olan araçlar kullanılmaktadır.

İkinci adımda ise dokümanlar arasındaki benzerikler hesaplanarak bir uzaklık matrisi oluşturulmaktadır. Dokümanlar arasındaki benzerlik hesabı için bu tez çalışmasında bağlı veri kaynakları kullanılarak yeni bir anlamsal benzerlik yöntemi geliştirilmiştir. Bu yöntemin detayları bir sonraki bölümde anlatılmaktadır.

Son olarak oluşturulan uzaklık matrisine hiyerarşik öbekleme yöntemi [32] uygulanmakta ve sonuç öbekleri elde edilmektedir.

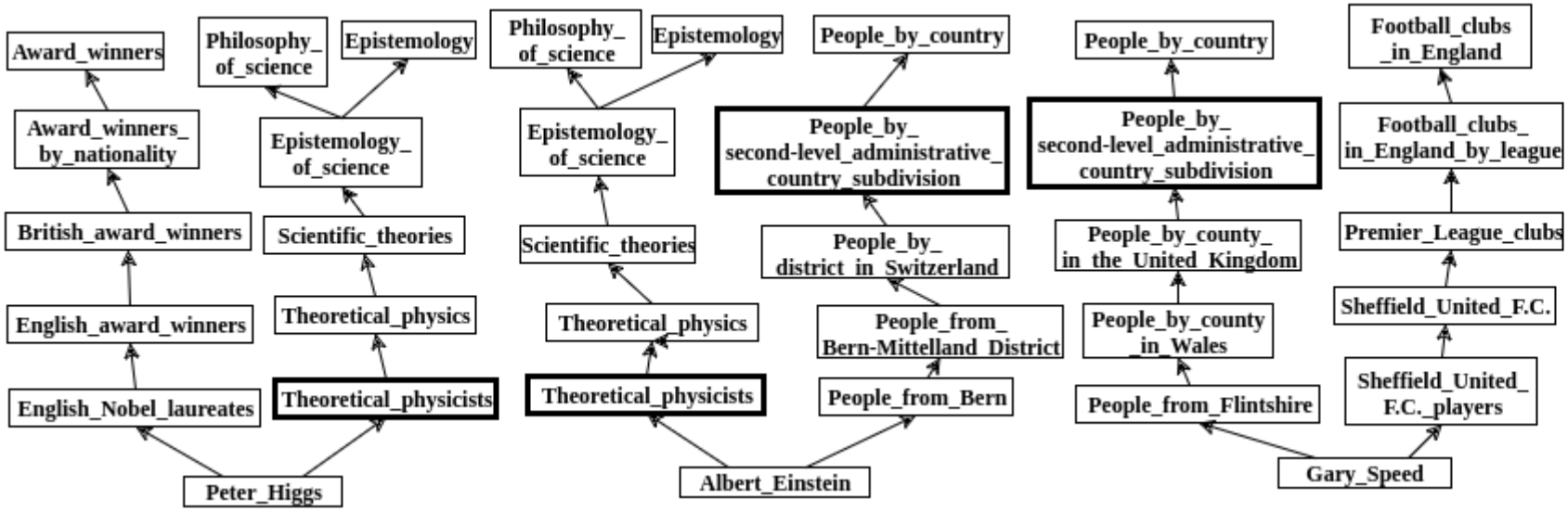
4.1 Bağlı Veri Kaynakları Arasındaki Anlamsal Benzerliklerin Hesaplanması

Bağlı veri kaynakları sayesinde varlıklar arasındaki benzerlikler hesaplanırken yazılışları dışında bu varlıkların farklı özellikleri dikkate alınabilmektedir. Örneğin "Albert Einstein" ve "Peter Higgs" varlıkları ele alındığında ikisinin de "fizikçi" (tür bilgisi) ve aynı zamanda "teorik fizikçi" (daha açık tür bilgisi) olduğu bilgisine ulaşılmaktadır. Böylece bu iki varlığın birbiriyle benzer olduğu bilgisi türlerine bakarak anlaşılabilmektedir. Bağlı veri bu tarz özelden genele doğru giden bir tür hiyerarşisi sunmaktadır. Örnek verilen varlıklar için DBpedia'dan elde edilen tür hiyerarşisinin bir kısmı Şekil 4.1'de gösterilmektedir.

Bu hiyerarşiler DBpedia Web servisine aşağıdaki SPARQL sorgusu yapılarak elde edilmiştir:

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX skos:<http://www.w3.org/2004/02/skos/core#>
PREFIX dct:<http://purl.org/dc/terms/>
SELECT DISTINCT ?l1 ?l2 ?l3 ?l4 ?l5
WHERE {{{<http://dbpedia.org/resource/Albert_Einstein> skos:broader ?l1}
UNION {<http://dbpedia.org/resource/Albert_Einstein> dct:subject ?l1}} .
{?l1 rdf:type skos:Concept} .
{{?l1 skos:broader ?l2} UNION {?l1 dct:subject ?l2}} .
{?l2 rdf:type skos:Concept} .
{{?l2 skos:broader ?l3} UNION {?l2 dct:subject ?l3}} .
{?l3 rdf:type skos:Concept} .
{{?l3 skos:broader ?l4} UNION {?l3 dct:subject ?l4}} .
{?l4 rdf:type skos:Concept} .
{{?l4 skos:broader ?l5} UNION {?l4 dct:subject ?l5}} .
{?l5 rdf:type skos:Concept}}
```

Bu sorguda türler elde edilirken `skos:broader` ve `dct:subject` özellikleri kullanılmış ve sadece 5 seviye boyunca türler bulunmuştur. Böylece sadece önemli türler bulunmuş ve sistem daha hızlı hale getirilmiştir.



Şekil 4.1: Albert Einstein, Peter Higgs ve Gary Speed için oluşturulan 5 seviyeli tür hiyerarşileri

Şekil 4.1'de görüldüğü gibi eğer iki varlık birbirine anlamsal olarak benzer ise, tür hiyerarşilerinde aynı ve yakın türlere (Higgs ve Einstein örneğinde Theoretical physicts türü gibi) sahip olmaktadırlar. Bunun dışında eğer iki varlık hiyerarşilerinde ortak türler içermiyorsa ya da içerdikleri ortak türler daha yüksek (varlığa uzak) seviyelerdeyse, daha düşük benzerlik göstermektedirler.

Benik vd. [6] çizgelerde (annotation graphs) çapraz genom desenleri bulmak amacıyla taksonomik uzaklık d_{tax} yöntemini geliştirmiştir. Bu uzaklık yöntemi daha sonra Paul vd. [56] tarafından geliştirilen varlık benzerliği bulma yönteminde kullanılmıştır. x ve y varlıkları arasındaki taksonomik uzaklığın formülü ağaıdaki şekildedir:

$$d_{tax}(x,y) = \frac{d(lca(x,y),x) + d(lca(x,y),y)}{d(root,x) + d(root,y)} \tag{4.1}$$

Formül 4.1'de $d(x,y)$, x ve y varlıklarının tür hiyerarşileri üzerindeki derinliklerinin farkı anlamına gelmektedir ve $|depth(x) - depth(y)|$ şeklinde hesaplanmaktadır. $lca(x,y)$ ise x ve y varlıklarının tür hiyerarşileri üzerindeki en küçük ortak ataları anlamına gelmektedir. Bu formül 0 ile 1 arasında bir değer vermektedir.

Taksonomik uzaklık yöntemi varlıklar arasındaki benzerlik *URLsim* hesaplanırken aşağıdaki şekilde kullanılabilmektedir:

$$URLsim(x,y) = 1 - d_{tax}(x,y) \tag{4.2}$$

Şekil 4.1'te verilen üç varlık ele alındığında `Peter_Higgs` (PH) ve `Albert_Einstein` (AE) varlıkları arasındaki benzerlik, bu iki varlığın en küçük ortak atası `Theoretical physicists`'e olan uzaklıklarının toplamının, iki tür hiyerarşisindeki en büyük uzaklıkların(bu durumda 5) toplamına bölümünün 1'den çıkarılması şeklinde hesaplandığında

$$URLsim(PH,AE) = 1 - (1+1)/(5+5) = 0.8 \tag{4.3}$$

elde edilmektedir. Şekil 4.1'te `Gary_Speed` (GS) isminde üçüncü bir varlık bulunmaktadır. Bu varlık tür olarak PH'ye hiç benzememektedir, yani ortak bir atası bulunmamaktadır. AE'ye ise iki varlık da `People by second level administrative country subdivision` ortak atasını bulundurduğu için az benzemektedir. Bu benzerlik

$$URLsim(AE,GS) = 1 - (4+4)/(5+5) = 0.2 \tag{4.4}$$

şeklinde hesaplanmaktadır.

4.2 Haber Dökümanları Arasındaki Benzerliklerin Hesaplanması

İki doküman d_{i} ve d_{j} arasındaki anlamsal benzerliği hesaplamak için yukarıda açıklanan, Formül 4.2 ile belirtilen yöntem aşağıdaki şekilde kullanılabilmekedir:

$$DocSim(d_i, d_j) = \frac{\sum_{\forall u \in d_i} \sum_{\forall v \in d_j} URLsim(u,v) \times \frac{w(u)+w(v)}{2}}{|d_i| \times |d_j|} \tag{4.5}$$

Formül 4.5'de gösterilen $|d_i|$, d_i dokümanında bulunan farklı bağlı veri kaynağı sayısını ifade etmektedir. $URLsim(u,v)$, doküman d_i'de yer alan u varlığı ile d_j'de yer alan v varlığı arasında URL benzerliğinin değeri (Formül 4.5) ve $w(u)$ ise u varlığının (URL bağlantısı) ağırlığı anlamına gelmektedir ($w(v)$ de v varlığının ağırlığı). $w(u)$ ve $w(v)$ ters doküman sıklığı (IDF) yöntemi ile aşağıdaki şekilde hesaplanmaktadır:

$$w(u) = log_{10} \frac{N}{n_u} \tag{4.6}$$

Formül 4.6'deki N, veri setindeki toplam doküman sayısı, n_u ise u veri kaynağını içeren doküman sayısıdır.

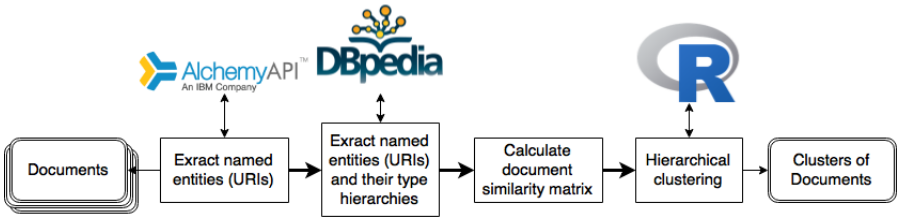
4.3 Öbekleme

Öbekleme işlemi için öncelikle hesaplanan benzerlikler kullanılarak bir uzaklık matrisi ($dist$) oluşturulmaktadır. Bu matris oluşturulurken her bir doküman ikilisi x ve y için aşağıdaki şekilde uzaklıklar hesaplanmaktadır:

$$dist(x,y) = \frac{DocSim(x,y)}{maxSim} \times 100 \tag{4.7}$$

Formül 4.7'te belirtilen $maxSim$, veri setindeki iki doküman arasında bulunabilecek en büyük uzaklığı belirtmektedir. Hesaplanan uzaklık değerleri 0 ile 100 arasında normalize edilmektedir. Daha sonra her doküman ikilisi için hesaplanan bu uzaklık değerleri kullanılarak uzaklık matirisi oluşturulmakta ve bu matris üzerine Bölüm 3.1.2'de açıklanan hiyerarşik öbekleme yöntemi uygulanmaktadır.

Öbekleme yöntemi olarak k-means daha sık kullanılan bir yöntemdir. Ama yapılan bazı



Şekil 4.2: Haber öbeklendirilmesi süreci

çalışmalara göre, hız olarak k-means daha iyi performans sergilemesine rağmen, doğruluk olarak bakıldığında hiyerarşik öbekleme daha iyi sonuçlar vermektedir [66, 34]. Bu çalışmada geliştirilen yöntemlerin performansı doğruluk açısından değerlendirildiği için hiyerarşik öbekleme yöntemi kullanılmıştır. Ayrıca k-means vektörel şekilde temsil edilen verilerin öbeklendirilmesinde kullanılmaktadır. Geliştirilen yöntemde vektörel bir gösterim olmadığı için k-means algoritması buna uygun değildir.

4.4 Uygulama

Geliştirilen bağlı veri kaynakları kullanılarak haberlerin öbeklendirilmesi yönteminin süreci Şekil 4.2'te gösterilmiştir. Bu sürece göre ilk adımda dokümanlarda geçen varlık isimleri ve bu varlıkların bağlı veri kaynaklarına olan bağlamtıları (link) bulunmaktadır. Bu işlem için Alchemy API[28] servisi kullanılmaktadır. Bu servis dokümanların gönderildiği ve sonuç olarak dokümanda geçen varlık isimlerinin, sıklıklarının, türlerinin ve DBpedia, YAGO gibi bağlı veri kaynaklarına olan linklerinin RDF/XML formatlarında sonuç olarak verildiği, yazılım servisi (SaaS) olarak sunulan çevrimiçi bir araçtır. Alchemy API ayrıca varlık ismi anlam ayrımı (named entity disambiguation) uygulayarak aynı şekilde yazılan fakat başka anlamlarda kullanılan varlıkları ayırt edebilmektedir.

Süreçteki sonraki adım dokümanlarda bulunan bağlı veri varlıklarının tür hiyerarşilerinin oluşturulmasıdır. Alchemy API servisiyle bağlı veri kaynağı olarak varlıkların DBpedia linkleri çıkarıldığı için tür hiyerarşileri oluşturulurken de DBpedia[29] kullanılmaktadır. Varlıklar için tür hiyerarşileri oluşturulduktan sonra, bu hiyerarşiler kullanılarak varlıklar arasındaki benzerlik hesaplanmaktadır. Bu benzerlikler kullanılarak da dokümanlar arasındaki anlamsal benzerlik elde edilerek uzaklık matrisi oluşturulmaktadır. Son olarak bu uzaklık matrisine R yazılımında[30] hiyerarşik öbekleme yöntemi uygulanarak sonuç öbekleri elde edilmektedir.

Algoritma 1'de detaylı algoritma sunulmaktadır. Bir doküman seti D verildiğinde, D içinde bulunan her bir doküman d önce Alchemy API[31] uygulamasıyla çözümlenmek-

[28] http://www.alchemyapi.com
[29] DBpedia endpoint, http://dbpedia.org/sparql
[30] R project, http://www.r-project.org
[31] http://www.alchemyapi.com

Algoritma 1 Bağlı veri kaynakları kullanılarak haberlerin öbeklendirilmesi

input: *set of news documents D*
output: *set of clusters C*

for *each document* $d \in D$ ***do***
    $URI_d \leftarrow$ *linkedDataURI(d) using AlchemyAPI*

for *each unique* $URI \in URI_d$ *where* $d \in D$ ***do***
    obtainTypeHierarchy(URI,5) via DBpedia endpoint
    calculateWeight(URI) using Inverse Document Frequency

for *each document* $d_1 \in D$ ***do***
    for *each document* $d_2 \in D$ $(d_1 \neq d_2)$ ***do***
        for *each* $u_1 \in URI_{d_1}$ ***do***
            for *each* $u_2 \in URI_{d_2}$ ***do***
                $DocSim[d_1,d_2] \leftarrow DocSim[d_1,d_2] + URLsim(u_1,u_2) * ((w(u_1) + w(u_2))/2)$
        $DocSim[d_1,d_2] = DocSim[d_1,d_2]/(entityCount(d_1) * entityCount(d_2))$

$maxRel \leftarrow max(DocSim[][])$
for *each* d_i *and* $d_j \in DocSim[]$ ***do***
    if $d_i \neq d_j$
        $dist[d_i,d_j] \leftarrow 100 * DocSim[d_i,d_j]/maxRel$
    else
        $dist[d_i,d_j] \leftarrow 0$

$C \leftarrow hclust(dist[][])$
return C

tedir. Çözümlenen dokümandaki varlık isimleri, türleri ve DBpedia linkleri XML formatında elde edilmektedir. Her bir link daha sonra DBpedia servisinde[32] *skos*:*broader* ve *dc*:*terms* yüklemleri(predicate) kullanılarak hiyerarşik olarak 5 seviye genişletilmektedir. Bu genişletmenin bir örneği üç varlık için Şekil 4.1'de gösterilmektedir. Örnek varlıklar fizikçiler Albert Einstein ve Peter Higgs ile futbolcu Gary Speed'dir. Her bir link 5 seviye boyunca kategorileriyle genişletilmiştir.

[32]DBpedia endpoint, http://dbpedia.org/sparql

5. DEĞERLENDİRME

Geliştirilen yöntem iki farklı veri seti üzerinde test edilmiştir. Veri setlerinin içeriği, yapısı ve uygulanan testlerin karşılaştırmalı sonuçları bu bölümde değerlendirilmektedir.

5.1 Veri Seti

Bu tez çalışmasında geliştirilen yöntemin test edilmesi için birisi bu çalışma için toplanan BBC News veri seti, diğeri ise doküman öbeklemesi ve sınıflandırılması çalışmalarında sıkça kullanılan stadart bir veri seti olan 20Newsgroup olmak üzere iki veri seti kullanılmıştır. Bu veri setlerinin detay bilgileri aşağıda verilmektedir. Veri setindeki dosyalar ve ara çıktılar `https://github.com/mertyucesan/newsclustering` adresinden ulaşılabilmektedir.

5.1.1 BBC news

Kullanılan veri setlerinden ilki bu çalışma için özel olarak oluşturulan BBC News veri setidir. Bu veri setinde BBC News[33] Web sitesinden toplanan 4 farklı kategoriden toplamda 209 tane haber makalesi bulunmaktadır. Her kategoride kaç adet makalenin bulunduğu Çizelge 5.1'de gösterilmektedir.

Çizelge 5.1: BBC news veri seti: haber kategorileri ve kategori başına haber sayısı

Kategori	**Doküman Sayısı**
science and environment	59
technology	55
sports	51
entertainment and arts	44

5.1.2 20Newsgroup

Diğer veri seti ise 20Newsgroup[34]'tur. 20Newsgroup doküman öbeklendirilmesi ve sınıflandırılması alanlarında en yaygın kullanılan veri setlerinden birisidir. Bu veri setinde 20 farklı haber grubunda her bir grupta 1.000 adet makale olacak şekilde toplamda 20.000

[33] 0http://www.bbc.com/news
[34] http://qwone.com/ jason/20Newsgroups/

haber dokümanı bulunmaktadır. Bu tez çalışmasında değerlendirme amacıyla 20Newsgroup veri setinin 3 farklı alt kümesi seçilmiştir. Alchemy API tarafından bulunan bağlı veri linkleri sayısı kontrol edildiğinde bütün dokümanlarda yeterince link bulunamadığı için en az 5 link içeren haberler değerlendirme için seçilmiştir. 3 tane alt küme seçilmesinin sebebi de, artan grup ve doküman sayısıyla birlikte seçilen kategorilerin birbirinden farklılaşması durumunda gösterilen performansın değerlendirilebilmesidir. Seçilen gruplar ve bu gruplardaki doküman sayıları Çizelge 5.2'de gösterilmektedir.

Çizelge 5.2: 20Newsgroup veri seti: seçilen gruplar ve grup başına haber sayısı

NG-1		**NG-2**		**NG-3**	
Gruplar	**#docs**	**Gruplar**	**#docs**	**Gruplar**	**#docs**
rec.sport.baseball	90	rec.sport.baseball	90	rec.sport.baseball	90
rec.sport.hockey	90	rec.sport.hockey	90	talk.politics.guns	90
talk.politics.misc	90	talk.politics.mideast	90	sci.space	90
talk.politics.guns	90	sci.crypt	90	soc.religion.christian	90
talk.politics.mideast	90	sci.space	90		
sci.crypt	90	soc.religion.christian	90		
sci.space	90				
soc.religion.christian	90				
Toplam	720	**Toplam**	540	**Toplam**	360

Kullanılan 4 veri setinden Alchemy API kullanılarak bulunan varlık isimleri, türleri ve DBpedia linklerinin sayıları Çizelge 5.3'te gösterilmektedir.

Çizelge 5.3: Kullanılan veri setlerinde Alchemy API tarafından bulunan farklı varlıkların sayıları

Veri Seti	**Varlık**	**#farklı varlıklar**	**#doküman başına varlık**
BBC News	Varlık İsmi	2837	24,18 $\pm$ 11,46
	Varlık Türü	31	8,29 $\pm$ 2,26
	DBpedia Linki	963	9,15 $\pm$ 6,72
NG-1	Varlık İsmi	10157	27,27 $\pm$ 12,93
	Varlık Türü	35	8,02 $\pm$ 2,42
	DBpedia Linki	2446	8,94 $\pm$ 5,98
NG-2	Varlık İsmi	8012	27,28 $\pm$ 13,06
	Varlık Türü	35	7,78 $\pm$ 2,42
	DBpedia Linki	2023	9,08 $\pm$ 6,26
NG-3	Varlık İsmi	5520	25,71 $\pm$ 12,49
	Varlık Türü	33	7,80 $\pm$ 2,51
	DBpedia Linki	1403	8,64 $\pm$ 6,12

5.2 Deneyler

Geliştirilen yöntem vektör benzerliği yöntemleriyle karşılaştırılmıştır. Dokümanların, içlerinde bulunan varlık isimleri, türleri ve DBpedia linkleri kullanılarak Çizelge 5.4'teki

şekilde seyrek (sparse) vektörleri oluşturulmuştur. Örneğin Çizelge 5.4'e göre doküman d_1'de toplamda 2 kere e_2 varlığı bulunmaktadır.

Çizelge 5.4: Vektör örneği (m:#doküman, n:#varlık)

	e_1	e_2	...	e_n
d_1	0	2	...	0
d_2	1	3	...	0
...	...	...	...	...
d_m	0	0	...	1

Dokümanlar farklı varlıklar (isim, tür ve DBpedia linki) kullanılarak vektörlere dönüştürülüp öbekleme işleminden geçirilmişlerdir. Oluşturulan vektör çeşitleri aşağıdaki gibidir:

- Kelime Çantası ve TF-IDF (BoW): Klasik kelime çantası yaklaşımı. Bütün dokümanlar için, içlerinde geçen kelimeler, o dokümandaki sıklığı ve veri setindeki bütün dosyalar içerisindeki sıklığı hesaplanarak, Formül 3.3 ile ağırlıklandırılmakta ve bu kelimeler TF-IDF ağırlıklarıyla birlikte dokümanların vektörlerini oluşturmaktadırlar.
- Varlık İsmi Sayısı (Named Entity / NE): Dokümanlar, içlerinde geçen varlık isimleri ve bu varlıkların o dokümanda geçme sayısı kullanılarak vektör haline getirilmektedir. Burada kullanılan varlıklar, sadece DBpedia linkleri olan varlıklar değil, Alchemy API tarafından bulunan bütün varlıkları içermektedir.
- Varlık Türü (Entity Type / ET): Varlıkların türleri ve bu türlerin dokümanda geçme sayıları kullanılarak vektörler oluşturulmaktadır.
- Bağlı Veri Kaynağı (URI): Dokümanlarda bulunan varlıkların bağlı veri kaynağı linkleri (bu durumda DBpedia linkleri) ve bu linklerin o dokümanda geçme sayıları kullanılarak vektörler oluşturulmaktadır.

Doküman vektörleri arasında uzaklık hesaplanırken en sık kullanılan yöntem Öklit Uzaklığıdır. Ancak yapılan testler sonucu Canberra Uzaklığı kullanılarak oluşturulan uzaklık matrisine hiyerarşik öbekleme yapıldığında daha iyi sonuçlar elde edildiği gözlenmiştir. Bu sebeple yukarıda belirtilen yöntemlerde vektörler arası uzaklık yöntemi olarak Formül 5.1'de gösterilen Canberra Uzaklığı kullanılmıştır.

$$d_c(x,y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{5.1}$$

Bu tez çalışmasında geliştirilen yöntemde ise bir vektör gösterim şekli yoktur ve uzaklıklar anlamsal şekilde taksonomik uzaklık kullanılarak hesaplanmıştır. Geliştirilen ve test edilen yöntemler aşağdaki gibidir:

- Bağlı Veri Kullanılarak Anlamsal Benzerlik (Linked Data Semantic Similarity - LDSS): Bağlı veri varlıkları arasındaki taksonomik uzaklık hesaplanarak uzaklık matrisi oluşturulmaktadır (Formül 4.7).

- Bağlı Veri Kullanılarak Kategori Ağırlıklı Anlamsal Benzerlik (Linked Data Semantic Similarity with Category Weights - LDSS-CW): Bu yöntemde Formül 4.2'e ek olarak bir kategori ağırlığı faktörü eklenmiştir. İki varlık arasında hesaplanan anlamsal benzerlik, o iki varlığın en düşük ortak atasının ağırlığı ile çarpılmaktadır. Bu sayede bulunan ortak ata çok sık bulunan (daha genel) bir kategoriyse o benzerliğin puanı azalmaktadır. Örneğin `Living_People` diye bir kategori bulunmaktadır. Bu kategori hayatta olan insan türündeki varlıkların hepsinde bulunmaktadır. Fakat Şekil 4.1'de örnek olarak verilen varlıklarda en düşük ortak ata olarak bulunan kategori `Theoretical_physicists` daha az görülen bir kategoridir ve bu yüzden önemlidir. Oluşturulan yeni benzerlik yönteminin formülü aşağıdaki hale gelmiştir:

$$URLsim(x,y) = (1 - d_{tax}(x,y)) \times w(lca(x,y)) \tag{5.2}$$

  Buradaki $w(lca)$ en yakın ortak ata (lowest common ancestor) kategorisinin ağırlığı anlamına gelmektedir. Herhangi bir kategori c'nin ağırlığı IDF yöntemi ile aşağıdaki şekilde hesaplanmaktadır:

$$w(c) = log_{10}\frac{N}{count(c)} \tag{5.3}$$

  Formül 5.3'teki N, veri setinde bulunan toplam doküman sayısı, $count(c)$ ise c kategorisini tür hiyerarşisinde bulunduran bağlı veri varlıklarını içeren dokümanların sayısıdır.

Yukarıdaki yöntemler ile uzaklık matrisleri hesaplandıktan sonra, bu matrislere hiyerarşik öbekleme yöntemi uygulanarak sonuç öbekleri elde edilmektedir. Hiyerarşik öbekleme yapılırken, bağlanma şekli olarak Ward'ın minimum varyans yöntemi [72] kullanılmıştır. Bu yöntem klasik bir karelerin toplamı kriterine dayanmaktadır, öbek içi varyans minimum olacak şekilde gruplama yapmaktadır. Hiyerarşik öbekleme işlemi için R yazılımı kullanılmıştır [57]. R'da bu yöntem Murtagh vd.'nin [49] çalışmasında tanımlanan amaç fonksiyonu kullanılarak geliştirilmiştir.

5.3 Analiz

Gerçekleştirilen deneylerin sonuçları Hassaslık (Precision/P), Hatırlama (Recall/R) ve F1 puanları hesaplanarak karşılaştırılmıştır. Hassaslık bir kategoriye ait olduğu söylenen dokümanların ne kadarının doğru olduğu bilgisini vermektedir. Hatırlama ise bir kategoriye ait olan dokümanların ne kadarının bulunduğunun ölçütüdür. F1 puanı da bu iki puanın eşit ağırlıklandırılarak ortalamasının alınmasıyla elde edilir. Bu puanlar Formüller 5.4, 5.5 ve 5.6 ile hesaplanmaktadır. Bu formüllerdeki TP gerçek pozitif, FP yanlış pozitif ve FN yanlış negatif değerlerini göstermektedir. TP bir kategoriye ait olarak bulunan ve gerçekten o gruba ait olan doküman sayısı, FP de bir kategoriye ait olduğu belirlenen ama aslında o kategoriye ait olmayan dokümanların sayısıdır. FN ise bir kategoriye ait olmadığı belirlenen ancak aslında o kategoriye ait olan doküman sayısını ifade etmektedir.

$$P = \frac{TP}{TP + FP} \tag{5.4}$$

$$R = \frac{TP}{TP + FN} \tag{5.5}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{5.6}$$

Hesaplanan bu puanlar Çizelge 5.5'te özetlenmektedir.

Çizelge 5.5'te görüldüğü gibi BBC News veri setinde, geliştirilen LDSS yöntemi ve bu yöntemin kategori ağırlıklı hali LDSS-CW yöntemi F1 puanı (F1=0,73 ve F1=0,88) ve hassaslık bakımından diğer vektör benzerliği yöntemlerinden çok daha iyi sonuç vermiştir. Ayrıca temel karşılaştırma yöntemi olarak alınan kelime çantası (BoW) yöntemine göre %8 ve %23 daha iyi F1 puanına sahiptir. Kelime çantası yöntemi, anlamsal benzerlik yöntemlerine göre daha kötü sonuç verse de, diğer vektör benzerliği yöntemlerinden F1 puanı (F1=0,65) olarak daha iyidir. Diğer vektör yöntemlerinden varlık türleri (ET) ve linkleri (URI) F1 puanı olarak sırasıyla 0,46 ve 0,53 alarak, varlık isimleri (NE) vektör yönteminden de düşük (F1=0,65), en kötü performansları göstermişlerdir. Bunun sebebi olarak, bu yöntemlerin Çizelge 5.3'te görülebileceği gibi daha az bilgi içermesi ve anlamsal bir yaklaşım sergilememesi gösterilebilir.

20Newsgroup veri setinin alt kümeleri olan NG-1, NG-2 ve NG-3 setlerinde de geliştirilen LDSS ve LDSS-CW yöntemleri daha iyi sonuç vermektedir. Referans olarak alınan kelime çantası yöntemi bu veri setlerinde, BBC News veri setine göre daha kötü sonuç vermektedir (F1=0,65'e karşı F1=0,50, 0,50, 0,52). LDSS yöntemi NG veri setlerinde yine kelime çantası yönteminden sırasıyla %7, %10 ve %6 daha iyi sonuç vermektedir, ve yine LDSS-CW %16, %16 ve %7 daha iyi olmak üzere en iyi sonuçları sağlamaktadır.

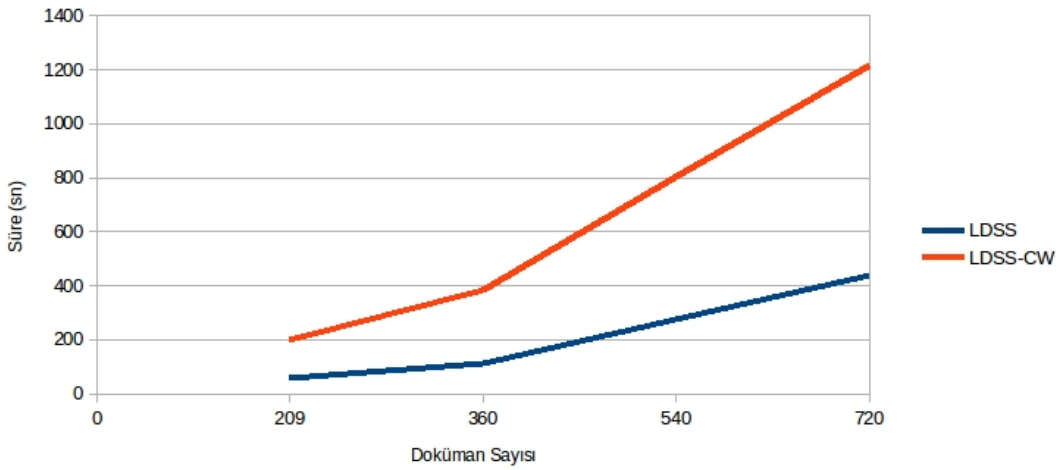
Çizelge 5.5: Hassaslık (P), Hatırlama (R) ve F1 puanları.

Veri seti	Yöntem	P	R	F1
BBC News	BoW	0.60	0.72	0.65
	NE	0.60	0.66	0.63
	ET	0.40	0.55	0.46
	URI	0.55	0.52	0.53
	LDSS	0.74	0.72	0.73
	LDSS-CW	**0.89**	**0.87**	**0.88**
NG-1	BoW	0.60	0.43	0.50
	NE	0.46	0.26	0.34
	ET	0.26	0.22	0.24
	URI	0.64	0.49	0.55
	LDSS	0.55	0.58	0.57
	LDSS-CW	**0.74**	**0.59**	**0.66**
NG-2	BoW	0.53	0.47	0.50
	NE	0.53	0.27	0.36
	ET	0.41	0.37	0.39
	URI	0.62	0.50	0.56
	LDSS	0.60	0.60	0.60
	LDSS-CW	**0.64**	**0.69**	**0.66**
NG-3	BoW	0.49	0.56	0.52
	NE	0.56	0.39	0.46
	ET	0.41	0.42	0.41
	URI	0.49	0.55	0.52
	LDSS	0.54	**0.61**	0.58
	LDSS-CW	**0.59**	0.58	**0.59**

Veri setlerinin içeriklerinin de öbekleme sonuçlarını etkilediği açıkça görülmektedir. Örneğin BBC News veri setinde 290 doküman, 4 öbek (kategori) ve doküman başına ortalama 9,15 DBpedia linki bulunmaktadır. NG-3 veri setinde ise 360 doküman, 4 öbek ve doküman başına ortalama 8,64 DBpedia linki bulunmaktadır. Bunlara bakarak benzer setler olduğunu söyleyebiliriz, fakat sonuçlar oldukça farklı çıkmaktadır. F1 puanı BBC News için 0,73 (LDSS) ve 0,88 (LDSS-CW) iken NG-3 için 0.58 (LDSS) ve 0.59'dur (LDSS-CW).

Diğer bir çıkarım da öbek sayısı azaldıkça, öbekleme performansının F1 puanı bakımından artmasıdır. Örneğin varlık isimleri yöntemi (NE) için NG veri setlerinde sonuçlara bakıldığında öbek sayısı sırasıyla 8, 6 ve 4 şeklinde azalırken F1 puanı 0,34 0,36, ve 0,46 şeklinde artmaktadır.

Hassaslık ve hatırlama puanlarına bakıldığında da LDSS-CW yönteminin NG-3 veri setinin hatırlama puanı dışındaki bütün veri setlerinde en iyi sonuçları verdiği görülmektedir. NG-3 veri setinde de en iyi hatırlama puanı LDSS yönteminde, en iyi ikinci puan da LDSS-CW yöntemindedir. Diğer yöntemlere bakıldığında tutarlı sonuçlar gözükmemektedir. Örneğin NG-1 veri setinde en düşük ikinci hassaslık puanı NE yöntemi sergilemiş-



Şekil 5.1: Süre analizi

ken, NG-3 veri setinde en yüksek ikinci puanı NE yöntemi sergilemiştir. NE yönteminin hatırlama puanlarına bakıldığında ise 20Newsgroup veri setlerinde en düşük performansları sergilemiştir.

Sonuç olarak bakıldığında bu tez çalışmasında geliştirilen anlamsal benzerlik yöntemleri LDSS ve LDSS-CW dokümanları doğru bir şekilde öbeklere ayırma konusunda klasik kelime çantası yöntemi ile varlık isimleri, türleri ve linkleriyle oluşturulan vektörlerden daha iyi sonuçlar vermektedir.

5.4 Süre Analizi

Bu çalışmada geliştirilen anlamsal benzerlik yöntemlerin hesaplama süreleri Şekil 5.1 gösterilmektedir. Burada gösterildiği üzere doküman sayısı ve hesaplama süresi arasındaki artış ilişkisi doğrusaldır.

Kategori ağırlıklarının da hesaplandığı yöntem LDSS-CW daha fazla süre almaktadır ve doküman sayısıyla birlikte artış hızı da daha fazladır. Bunun sebebi doküman sayısı arttıkça kategori sayısının da artmasıdır.

Hesaplama süresinin iyileştirilmesine bu çalışmada değinilmemiştir, fakat geliştirilen yöntemin daha kullanılabilir bir yöntem olabilmesi için gelecekte süre verimliliğinin arttırılması konusunda da çalışmalar yapılacaktır.

6. SONUÇ

Bu çalışmada doküman veya haber öbeklendirilmesi için varlık isimleri arasında bağlı veri kaynakları tabanlı anlamsal bir benzerlik yöntemi (LDSS) geliştirilmiştir. Geliştirilen anlamsal benzerlik yöntemi dokümanlarda bulunan varlıkların bağlı veri (DBpedia) linkleri üzerinden elde edilen tür hiyerarşilerini kullanmaktadır. Varlık isimleri ve bu varlıkların DBpedia linkleri Alchemy API servisi kullanılarak elde edilmektedir. Daha sonra bu linklerin türleri DBpedia üzerinden sorgulanarak bulunmakta ve linkler 5 seviye boyunca hiyerarşik olarak genişletilmektedir. Oluşturulan tür hiyerarşileri kullanılarak varlıklar arasında anlamsal bir benzerlik hesaplanmaktadır. Hesaplanan bu benzerlikler daha sonra dokümanlar arasındaki benzerlik hesaplanırken kullanılmaktadır ve son olarak da hiyerarşik öbekleme yöntemi uygulanarak sonuç öbekleri elde edilmektedir.

Geliştirilen yöntem ile birlikte farklı vektör tabanlı yöntemler test edilmiş ve sonuçlar klasik kelime çantası (BoW) yöntemi ile karşılaştırılmıştır. Sonuçlar bağlı veri kaynağı tabanlı anlamsal benzerlik yönteminin (LDSS), kelime çantası ve diğer yöntemlerden oldukça iyi olduğunu göstermektedir. Ayrıca LDSS yönetimine kategori ağırlıkları ekleyerek geliştirdiğimiz diğer bir anlamsal benzerlik yöntemi LDSS-CW'nin LDSS'e göre daha da iyi sonuçlar verdiği gözlenmiştir. Bu sonuçlarla, bağlı veri kaynakları kullanılarak yapılan anlamsal benzerlik hesaplamalarının, sadece kelime benzerliklerine bakılarak yapılan benzerlik yöntemlerinden daha doğru sonuçlar verdiği gösterilmiştir.

Bağlı veri kaynakları oldukça büyük yapılandırılmış veri içermekte ve sürekli artmaktadır. Bağlı veri kaynakları farklı makine öğrenmesi işlemleri için kullanılabilmektedir. Zaman geçtikçe bu tarz yöntemlerin uygulamaları daha fazla görülebilecektir. Elde edilen sonuçlarda en iyi F1 puanı olarak 0,88 elde edilmiştir (BBC News veri seti ve LDSS-CW yöntemi). Bu daha gidilecek yol olduğunu göstermektedir. Doküman benzerliği konusundaki çalışmalara bağlı veri kaynaklarının başka özelliklerinin de kullanılması ile geliştirmelere devam edilebilir.

Doküman öbeklenmesi konusunda dış veri kaynağı kullanan birçok çalışma bulunmaktadır. Bu konuda geliştirilmesi gereken önemli bir diğer konu ise geliştirilen yöntemlerin hızlarıdır. Gelecekteki çalışmalar geliştirilen yöntemlerin daha hızlı çalışması üzerine olacaktır. Dış veri kaynaklarının içerik olarak oldukça büyük oldukları düşünüldüğünde, yöntemlerin hızlı çalışması için düşünülmesi gereken konular bu kaynaklar üzerindeki veriye erişim hızını artırılması ve daha verimli benzerlik hesaplama yöntemleri geliştirilmesi olmalıdır.

KAYNAKLAR

[1] **Aggarwal, C.C. ve Zhai, C.** (2012). "A survey of text clustering algorithms". In: *Mining Text Data*. Springer, pp. 77–128.

[2] **Agrawal, R. ve Phatak, M.** (2013). "A novel algorithm for automatic document clustering". In: *Advance Computing Conference (IACC), 2013 IEEE 3rd International*. IEEE, pp. 877–882.

[3] **Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. ve Ives, Z.** (2007). "Dbpedia: A nucleus for a web of open data". In: *The semantic web*. Springer, pp. 722–735.

[4] **Baeza-Yates, R., Ribeiro-Neto, B. vd.** (1999). *Modern information retrieval*. Vol. 463. ACM press New York.

[5] **Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M. ve Etzioni, O.** (2007). "Open Information Extraction from the Web." In: *IJCAI*. Vol. 7, pp. 2670–2676.

[6] **Benik, J., Chang, C., Raschid, L., Vidal, M., Palma, G. ve Thor, A.** (2012). "Finding cross genome patterns in annotation graphs". In: *International Conference on Data Integration in the Life Sciences*. Springer, pp. 21–36.

[7] **Berkhin, P.** (2006). "A survey of clustering data mining techniques". In: *Grouping multidimensional data*. Springer, pp. 25–71.

[8] **Berners-Lee, T., Hendler, J., Lassila, O. vd.** (2001). "The semantic web". In: *Scientific american* 284.5, pp. 28–37.

[9] **Bizer, C., Heath, T., Idehen, K. ve Berners-Lee, T.** (2008). "Linked data on the web (LDOW2008)". In: *Proceedings of the 17th international conference on World Wide Web*. ACM, pp. 1265–1266.

[10] **Bizer, C., Heath, T. ve Berners-Lee, T.** (2009). "Linked data-the story so far". In: *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pp. 205–227.

[11] **Bollacker, K., Evans, C., Paritosh, P., Sturge, T. ve Taylor, J.** (2008). "Freebase: a collaboratively created graph database for structuring human knowledge". In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, pp. 1247–1250.

[12] **Bouras, C. ve Tsogkas, V.** (2012). "A clustering technique for news articles using WordNet". In: *Knowledge-Based Systems* 36, pp. 115–128.

[13] **Bouras, C. ve Tsogkas, V.** (2010). "W-kmeans: clustering news articles using wordnet". In: *International Conference on Knowledge Based and Intelligent Information and Engineering Systems*. Springer, pp. 379–388.

[14] **Cano, A.E., Varga, A., Rowe, M., Ciravegna, F. ve He, Y.** (2013). "Harnessing linked knowledge sources for topic classification in social media". In: *Proceedings of the 24th ACM Conference on Hypertext and Social Media*. ACM, pp. 41–50.

[15] **Cimiano, P., Lopez, V., Unger, C., Cabrio, E., Ngomo, A.N. ve Walter, S.** (2013). "Multilingual question answering over linked data (qald-3): Lab overview". In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, pp. 321–332.

[16] **Dostal, M., Nykl, M. ve Ježek, K.** (2014). "Exploration of document classification with linked data and pagerank". In: *Intelligent Distributed Computing VII*. Springer, pp. 37–43.

[17] **Fader, A., Soderland, S. ve Etzioni, O.** (2011). "Identifying relations for open information extraction". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1535–1545.

[18] **Forsati, R., Mahdavi, M., Shamsfard, M. ve Meybodi, M.R.** (2013). "Efficient stochastic algorithms for document clustering". In: *Information Sciences* 220, pp. 269–291.

[19] **Fraley, C. ve Raftery, A.E.** (1998). "How many clusters? Which clustering method? Answers via model-based cluster analysis". In: *The computer journal* 41.8, pp. 578–588.

[20] **Gabrilovich, E. ve Markovitch, S.** (2007). "Computing semantic relatedness using Wikipedia-based explicit semantic analysis." In: *International Joint Conference on Artificial Intelligence (IJcAI)*. Vol. 7, pp. 1606–1611.

[21] **Hakimov, S., Oto, S.A. ve Dogdu, E.** (2012). "Named entity recognition and disambiguation using linked data and graph-based centrality scoring". In: *Proceedings of the 4th international workshop on semantic web information management*. ACM, p. 4.

[22] **Hofmann, T.** (2001). "Unsupervised learning by probabilistic latent semantic analysis". In: *Machine learning* 42.1-2, pp. 177–196.

[23] **Hu, J., Fang, L., Cao, Y., Zeng, H., Li, H., Yang, Q. ve Chen, Z.** (2008). "Enhancing text clustering by leveraging Wikipedia semantics". In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 179–186.

[24] **Huang, A.** (2008). "Similarity measures for text document clustering". In: *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, pp. 49–56.

[25] **Hulpus, I., Hayes, C., Karnstedt, M. ve Greene, D.** (2013). "Unsupervised graph-based topic labelling using dbpedia". In: *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, pp. 465–474.

[26] **Jain, A.K.** (2010). "Data clustering: 50 years beyond K-means". In: *Pattern recognition letters* 31.8, pp. 651–666.

[27] **Jain, A.K., Murty, M.N. ve Flynn, P.J.** (1999). "Data clustering: a review". In: *ACM computing surveys (CSUR)* 31.3, pp. 264–323.

[28] **Jain, A.K. ve Dubes, R.C.** (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

[29] **Jardine, N. ve van Rijsbergen, C.J.** (1971). "The use of hierarchic clustering in information retrieval". In: *Information storage and retrieval* 7.5, pp. 217–240.

[30] **Jiang, Y., Bai, W., Zhang, X. ve Hu, J.** (2016). "Wikipedia based information content and semantic similarity computation". In: *Information Processing & Management*.

[31] **Jiang, Y., Zhang, X., Tang, Y. ve Nie, R.** (2015). "Feature-based approaches to semantic similarity assessment of concepts using Wikipedia". In: *Information Processing & Management* 51.3, pp. 215–234.

[32] **Johnson, S.C.** (1967). "Hierarchical clustering schemes". In: *Psychometrika* 32.3, pp. 241–254.

[33] **Kaufman, L. ve Rousseeuw, P.J.** (2009). *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons.

[34] **Kaur, M. ve Kaur, U.** (2013). "Comparison between k-means and hierarchical algorithm using query redirection". In: *International Journal of Advanced Research in Computer Science and Software Engineering* 3.7.

[35] **Kim, C. ve Park, S.** (2013). "Enhancing Text Document Clustering Using Non-negative Matrix Factorization and WordNet". In: *Journal of information and communication convergence engineering* 11.4, pp. 241–246.

[36] **Kim, H., Hong, K. ve Chang, J.Y.** (2015). "Semantically enriching text representation model for document clustering". In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. ACM, pp. 922–925.

[37] **Klyne, G. ve Carroll, J.J.** (2006). "Resource description framework (RDF): Concepts and abstract syntax". In:

[38] **Kuralenok, I. ve Nekrest'yanov, I.** (2000). "Automatic document classification based on latent semantic analysis". In: *Programming and Computer Software* 26.4, pp. 199–206.

[39] **Landauer, T.K., Foltz, P.W. ve Laham, D.** (1998). "An introduction to latent semantic analysis". In: *Discourse processes* 25.2-3, pp. 259–284.

[40] **Leal, J.P., Rodrigues, V. ve Queirós, R.** (2012). "Computing semantic relatedness using dbpedia". In: *OpenAccess Series in Informatics*. Vol. 21. Schloss Dagstuhl Leibniz Zentrum fuer Informatik.

[41] **Li, Y., Chung, S.M. ve Holt, J.D.** (2008). "Text document clustering based on frequent word meaning sequences". In: *Data & Knowledge Engineering* 64.1, pp. 381–404.

[42] **Liu, T., Chen, Z., Zhang, B., Ma, W. ve Wu, G.** (2004). "Improving text classification using local latent semantic indexing". In: *Data Mining, 2004. ICDM'04. 4th IEEE International Conference on*. IEEE, pp. 162–169.

[43] **Menzel, J.** (2010). "Deeper Understanding with Metaweb". In: *Official Google Blog, July*.

[44] **Meymandpour, R. ve Davis, J.G.** (2015). "Enhancing Recommender Systems Using Linked Open Data-Based Semantic Analysis of Items". In: *Proceedings of the 3rd Australasian Web Conference (AWC 2015)*. Vol. 27, p. 30.

[45] **Miller, G.A.** (1995). "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11, pp. 39–41.

[46] **Ming, Z., Wang, K. ve Chua, T.** (2010). "Prototype hierarchy based clustering for the categorization and navigation of web collections". In: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 2–9.

[47] **Montalvo, S., Martínez, R., Casillas, A. ve Fresno, V.** (2007). "Multilingual news clustering: Feature translation vs. identification of cognate named entities". In: *Pattern Recognition Letters* 28.16, pp. 2305–2311.

[48] **Montalvo, S., Martínez, R., Casillas, A. ve Fresno, V.** (2007). "Bilingual news clustering using named entities and fuzzy similarity". In: *International Conference on Text, Speech and Dialogue*. Springer, pp. 107–114.

[49] **Murtagh, F. ve Legendre, P.** (2014). "Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion?" In: *Journal of Classification* 31.3, pp. 274–295.

[50] **Naik, M.P., Prajapati, H.B. ve Dabhi, V.K.** (2015). "A survey on semantic document clustering". In: *Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on*. IEEE, pp. 1–10.

[51] **Ni, Y., Xu, Q.K., Cao, F., Mass, Y., Sheinwald, D., Zhu, H.J. ve Cao, S.S.** (2016). "Semantic Documents Relatedness using Concept Graph Representation". In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, pp. 635–644.

[52] **Nunes, B.P., Fetahu, B., Kawase, R., Dietze, S., Casanova, M.A. ve Maynard, D.** (2015). "Interlinking Documents Based on Semantic Graphs with

an Application". In: *Knowledge-Based Information Systems in Practice*. Springer, pp. 139–155.

[53] **Oto, S. A.** (2012). "Varlık İsimleri Arasındaki İlişkiler Kullanılarak Haberlerin Öbeklenmesi". MA thesis. TOBB University of Economy and Technology.

[54] **Pasca, M. ve Harabagiu, S.** (2001). "The informative role of WordNet in open-domain question answering". In: *Proceedings of NAACL-01 Workshop on WordNet and Other Lexical Resources*, pp. 138–143.

[55] **Patil, L.H. ve Atique, M.** (2013). "A Semantic approach for effective document clustering using WordNet". In: *arXiv preprint arXiv:1303.0489*.

[56] **Paul, C., Rettinger, A., Mogadala, A., Knoblock, C.A. ve Szekely, P.** (2016). "Efficient Graph-Based Document Similarity". In: *International Semantic Web Conference*. Springer, pp. 334–349.

[57] **R Core Team** (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria.

[58] **Radev, D.R., Blair-Goldensohn, S., Zhang, Z. ve Raghavan, R.S.** (2001). "Newsinessence: A system for domain-independent, real-time news clustering and multi document summarization". In: *Proceedings of the first international conference on Human language technology research*. Association for Computational Linguistics, pp. 1–4.

[59] **Ravichandran, D. ve Hovy, E.** (2002). "Learning surface text patterns for a question answering system". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pp. 41–47.

[60] **Salton, G. ve Buckley, C.** (1988). "Term-weighting approaches in automatic text retrieval". In: *Information processing & management* 24.5, pp. 513–523.

[61] **Schuhmacher, M. ve Ponzetto, S.P.** (2014). "Knowledge-based graph document modeling". In: *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, pp. 543–552.

[62] **Sidhu, N.K. ve Kaur, R.** (2013). "Clustering In Data Mining". In: *International Journal of Computer Trends and Technology (IJCTT)* 4, pp. 710–714.

[63] **Song, W., Li, C.H. ve Park, S.C.** (2009). "Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures". In: *Expert Systems with Applications* 36.5, pp. 9095–9104.

[64] **Spanakis, G., Siolas, G. ve Stafylopatis, A.** (2012). "Exploiting Wikipedia knowledge for conceptual hierarchical clustering of documents". In: *The Computer Journal* 55.3, pp. 299–312.

[65] **Sree, P.K. ve Babu, I.R.** (2014). "Improving quality of clustering using cellular automata for information retrieval". In: *preprint arXiv:1401.2684*.

[66] **Steinbach, M., Karypis, G., Kumar, V. vd.** (2000). "A comparison of document clustering techniques". In: *KDD workshop on text mining*. Vol. 400. 1. Boston, pp. 525–526.

[67] **Suchanek, F.M., Kasneci, G. ve Weikum, G.** (2007). "Yago: a core of semantic knowledge". In: *Proceedings of the 16th international conference on World Wide Web*. ACM, pp. 697–706.

[68] **Sun, J., Chen, Z., Zeng, H., Lu, Y., Shi, C. ve Ma, W.** (2004). "Supervised latent semantic indexing for document categorization". In: *Data Mining, 2004. ICDM'04. 4th IEEE International Conference on*. IEEE, pp. 535 –538.

[69] **Szczuka, M. ve Janusz, A.** (2013). "Semantic clustering of scientific articles using explicit semantic analysis". In: *Transactions on Rough Sets XVI*. Springer, pp. 83–102.

[70] **Unger, C., Bühmann, L., Lehmann, J. vd.** (2012). "Template-based question answering over RDF data". In: *Proceedings of the 21st international conference on World Wide Web*. ACM, pp. 639–648.

[71] **Unger, C., Forascu, C., Lopez, V., Ngomo, A.N., Cabrio, E., Cimiano, P. ve Walter, S.** (2014). "Question answering over linked data (QALD-4)". In: *Working Notes for CLEF 2014 Conference*.

[72] **Ward Jr, J.H.** (1963). "Hierarchical grouping to optimize an objective function". In: *Journal of the American statistical association* 58.301, pp. 236–244.

[73] **Wei, T., Lu, Y., Chang, H., Zhou, Q. ve Bao, X.** (2015). "A semantic approach for text clustering using WordNet and lexical chains". In: *Expert Systems with Applications* 42.4, pp. 2264–2275.

[74] **Weld, D.S., Hoffmann, R. ve Wu, F.** (2009). "Using wikipedia to bootstrap open information extraction". In: *ACM SIGMOD Record* 37.4, pp. 62–68.

[75] **Wu, F. ve Weld, D.S.** (2010). "Open information extraction using Wikipedia". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 118–127.

[76] **Yu, B., Xu, Z. ve Li, C.** (2008). "Latent semantic analysis for text categorization using neural network". In: *Knowledge-Based Systems* 21.8, pp. 900–904.

[77] **Zhang, Z., Gentile, A.L. ve Ciravegna, F.** (2013). "Recent advances in methods of lexical semantic relatedness–a survey". In: *Natural Language Engineering* 19.04, pp. 411–479.

[78] **Zhu, G. ve Iglesias, C.A.** (2016). "Computing Semantic Similarity of Concepts in Knowledge Graphs". In: *IEEE Transactions on Knowledge and Data Engineering*.

ÖZGEÇMİŞ

Ad-Soyad : Mehmet Mert Yücesan

Uyruğu : T.C.

Doğum Tarihi ve Yeri : 24.08.1989 Ankara

E-posta : mertyucesan57@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2012, Bilkent Üniversitesi, Bilgisayar Mühendisliği
- **Lisans** : 2016, Anadolu Üniversitesi, İşletme

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	**Yer**	**Görev**
2012 - Halen	Tork Yazılım	Kurucu Ortak
2014 - 2016	TOBB ETÜ	Burslu Yüksek Lisans Öğrencisi

YABANCI DİL: İngilizce

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- Mehmet Mert Yucesan, Erdogan Dogdu. News Clustering Using Linked Data Resources and Their Relationships. Proc. of the International Conference on Artificial Intelligence and Data Processing (IDAP), Sep 17-18, 2016, Malatya, Turkey

DİĞER YAYINLAR, SUNUMLAR VE PATENTLER:

- M. Akif, Agca, Senol Atac, M. Mert Yucesan, Gokhan Y. Kucukayan, A. Murat Özbayoglu and Erdogan Dogdu, ”Opinion Mining of Microblog Texts on Hadoop Ecosystem“, International Journal of Cloud Computing, vol.5:1, pp.79-90 (2016)
- Muhammed Akif Agca, Senol Atac, M. Mert Yucesan, Gokhan Y. Kucukayan, A. Murat Özbayoglu, Erdogan Dogdu, “Opinion Mining of Microblog Texts on Hadoop Ecosystem”, 2nd IBM Cloud Academy Conference, ICA CON 2014.