

**T.C.
KASTAMONU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**DERİN ÖĞRENME YAKLAŞIMLARINI KULLANARAK
MAKALE ATIFLARININ SEMANTİK ANALİZİ**

Nabila Elmukhtar Mohamad ALBANNAI

**Danışman
Jüri Üyesi
Jüri Üyesi**

**Dr. Öğr. Üyesi Yasemin GÜLTEPE
Dr. Öğr. Üyesi Can Doğan VURDU
Dr. Öğr. Üyesi Cevat RAHEBİ**

**YÜKSEK LİSANS TEZİ
MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI**

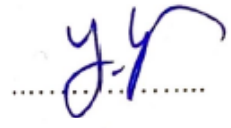
KASTAMONU – 2019

TEZ ONAYI

Nabila Elmukhtar Mohamad ALBANNAI tarafından hazırlanan “**Derin Öğrenme Yaklaşımlarını Kullanarak Makale Atıflarının Semantik Analizi**” adlı tez çalışması aşağıdaki jüri üyeleri önünde savunulmuş ve **oy birliği** ile Kastamonu Üniversitesi Fen Bilimleri Enstitüsü **Malzeme Bilimi ve Mühendisliği Ana Bilim Dalı**’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman

Dr. Öğr. Üyesi Yasemin GÜLTEPE
Kastamonu Üniversitesi

.....


Jüri Üyesi

Dr. Öğr. Üyesi Can Doğan VURDU
Kastamonu Üniversitesi

.....


Jüri Üyesi

Dr. Öğr. Üyesi Cevat RAHEBI
Altınbaş Üniversitesi

.....


13/12/2019

Enstitü Müdürü

Doç. Dr. Nur BELKAYALI

.....


TAAHHÜTNAME

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 bildirir ve taahhüt ederim.

Nabila Elmukhtar Mohamad ALBANNAI



ÖZET

Yüksek Lisans Tezi

DERİN ÖĞRENME YAKLAŞIMLARINI KULLANARAK MAKALE ATIFLARININ SEMANTİK ANALİZİ

Nabila Elmukhtar Mohamad ALBANNAI
Kastamonu Üniversitesi
Fen Bilimleri Enstitüsü
Malzeme Bilimi ve Mühendisliği Ana Bilim Dalı

Danışman: Dr. Öğr. Üyesi Yasemin GÜLTEPE

Anlamsal yayıncılık; yayınlanan bir dergi makalesinin anlamını geliştirmek, otomatik keşfini kolaylaştırmak, anlamsal olarak ilgili makalelere bağlanmasını sağlamak, makale içindeki verilere işlem yapılabilir biçimde erişim sağlamak ve makaleler arasında verilerin bütünlüğünü kolaylaştırmak için anlamsal web teknolojilerinin kullanımı ile sağlanmaktadır. Anlamsal web teknolojileri ve bilgi gösterimi, bilginin yeniden yapılandırılmasını, yapılandırılmış ve makine tarafından okunabilir bir şekilde paylaşımını artırır. Anlamsal web teknolojileri ve derin öğrenme, akıl yürütme, onaylama ve tahmin etme gibi insan zekasını bilgisayar aracılığıyla taklit eden akıllı yapay nesnelere yaratma hedefini paylaşır.

Yapılan bu tez çalışmanın amacı; Anlamsal web teknolojileri kullanılarak geliştirilen atıf ontolojisi ile birlikte derin öğrenme uygulamaları kullanılarak hem anlamsal web teknolojilerinin hem de derin öğrenme uygulamalarının avantajlarını sunan derin öğrenme tabanlı atıf ağında semantik arama yöntemi geliştirmektir.

Çalışma kapsamında geliştirilen atıf ontoloji, dergi makalelerine yapılan atıflar hakkında bilgileri sağlamaktadır. Atıf ontoloji, Protégé ontoloji geliştirme editörü kullanılarak yaratılmıştır. Protégé ontoloji geliştirme editörünün grafik arayüzü sayesinde atıf ile ilgili konular görsel olarak tanımlanmakta ve böylelikle tanımlanmak istenilen alan modellenmektedir.

Tez çalışmasından önerilen atıf ağının semantik analizinde derin öğrenme yöntemleri kullanılarak geliştirilen sistem sonucunda eşleştirme işlemlerinde kesin eşleşmelerden daha fazla benzer dergi makalesi bularak önerilen yöntemin dergi makalesi eşleştirme başarımını artırmaya yönelik etkileri tespit edilmiştir.

Anahtar Kelimeler: Anlamsal web, ontoloji, derin öğrenme, makale, atıf

2019, 51 sayfa

Bilim Kodu: 101

ABSTRACT

MSc. Thesis

SEMANTIC ANALYSIS OF ARTICLE CITES USING DEEP LEARNING APPROACHES

Nabila Elmukhtar Mohamad ALBANNAI
Kastamonu University
Graduate School of Natural and Applied Sciences
Department of Material Science and Engineering

Supervisor: Assist. Prof. Dr. Yasemin GÜLTEPE

Abstract: Semantic publishing; the use of web and semantic web technologies to improve the meaning of a published journal article, facilitate its automatic discovery, connect to semantically related articles, provide processable access to data within the article, and facilitate the integrity of data between articles. Semantic web technologies and information display enhance information restructuring, sharing in a structured and machine-readable way. It shares the goal of creating semantic web technologies and intelligent artificial objects that imitate human intelligence via computer, such as deep learning, reasoning, validation and prediction.

The aim of this thesis is to develop a deep learning based search method in citation network that provides the advantages of both semantic web technologies and deep learning applications by using deep learning applications together with citation ontology developed using semantic web technologies.

The citation ontology developed within the scope of the study provides information about citations to journal articles. Citation ontology was created using the Protégé ontology development editor. The protégé ontology development editor's graphical interface allows visual identification of citation issues, so that the desired area can be modeled.

In the semantic analysis of the citation network proposed in the thesis study, the system developed by using deep learning methods found more similar journal articles in the matching process than the exact matches, and the effects of the proposed method to increase the journal article matching performance were determined.

Key Words: Semantic web, ontology, deep learning, article, citation

2019, 51 pages

Science Code: 101

TEŞEKKÜR

Öncelikle bu tez konusu üzerinde bana çalışma imkânı sunan tez danışmanım Dr. Öğr. Üyesi Yasemin GÜLTEPE'ye tez çalışmam süresince deneyimleri, bilgileri ve önerileriyle araştırma ve geliştirmeyi yönlendirmesi ve sağladığı kaynaklarla destek olmasından dolayı teşekkürü bir borç bilirim.

Tez izleme toplantılarında jüri olarak görev alan Dr. Öğr. Üyesi Can Doğan VURDU'a ve Dr. Öğr. Üyesi Cevat RAHEBI'a değerli önerileri ve çalışmaya olan katkıları için teşekkürlerimi sunarım.

Ayrıca tez çalışmaları sırasında fikirlerine başvurduğum Kastamonu Üniversitesi Bilgisayar Mühendisliği'nin değerli öğretim üyelerine ve elemanlarına teşekkürlerimi sunarım.

Her zaman yanımda olan sevgili aileme ve yakınlarıma minnettar olduğumu belirtmek isterim.

Kastamonu Üniversitesinde aldığım eğitim için yaptıkları destekten dolayı Libya Hükümetine ve Türkiye'deki Libya Büyükelçiliğine teşekkür ederim.

Nabila Elmukhtar Mohamad ALBANNAI
Kastamonu, Aralık, 2019

İÇİNDEKİLER

	Sayfa
TEZ ONAYI.....	ii
TAAHHÜTNAME.....	iii
ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER.....	vii
SİMGELER VE KISALTMALAR DİZİNİ.....	viii
ŞEKİLLER DİZİNİ.....	ix
TABLolar DİZİNİ.....	x
1. GİRİŞ.....	1
2. DERİN ÖĞRENME.....	5
2.1. Derin Öğrenme Topolojisi ve Uygulamaları.....	7
2.2. Derin Öğrenme Mimarisi İçin Gerekli Bazı Katmanlar.....	8
2.2.1. Tam Bağlı Katmanlar.....	8
2.2.2. Evrişim Katmanı.....	8
2.2.3. Havuzlama Katmanı.....	9
2.2.4. Düzeltilmiş Doğrusal Katman.....	11
3. ANLAMSAL DERİN ÖĞRENME.....	12
3.1. Anlamsal Web Teknolojileri.....	12
3.1.1. RDF.....	12
3.1.2. Web Ontoloji Dili.....	13
3.1.3. SPARQL.....	14
3.1.4. Protégé.....	15
3.2. Anlamsal Derin Öğrenme.....	15
4. MATERYAL VE METOT.....	21
4.1. Atıf Ontolojisi.....	22
4.2. Anlamsal Arama Modeli.....	25
4.3. Kelime Vektörlerinin Çıkarılması.....	26
4.4. Sistemin Arayüzleri.....	30
4.5. Araştırma Bulguları.....	34
4.5.1. SPARQL Kullanımı.....	35
4.5.2. Kelime Vektörlerinin Kullanımı.....	37
4.5.3. Değerlendirme.....	41
5. SONUÇLAR.....	44
KAYNAKLAR.....	46
ÖZGEÇMİŞ.....	51

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

YSA	Yapay Sinir Ağları
BoW	Kelime/Sözcük Çantası
CNN	Evrişimsel Sinir Ağları
DL	Açıklama Mantığı
DOI	Sayısal Nesne Tanımlayıcı
ML	Makine Öğrenmesi
NLP	Doğal Dil İşleme
OWL	Web Ontoloji Dili
RDF	Kaynak Tanımlama Çerçevesi
RDFS	Kaynak Tanımlama Çerçevesi Şeması
ReLU	Düzeltilmiş Doğrusal Katman
SG	Atlama Gram
SPARQL	Protokol ve RDF Sorgu Dili
SW	Semantik Web
URI	Standart Kaynak Tanımlayıcısı
URL	Standart Kaynak Bulucu
URN	Hiper Metin Ön İşleme
Word2Vec	Kelime Vektörleri
WWW	Dünya Çağında Ağ
XML	Genişletilmiş İşaretleme Dili

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. İlk Derin Ağ Mimarisi	7
Şekil 2.2. Tam Bağlantılı Katmanlardaki Nöronlar ve Ağırlıklar.....	8
Şekil 2.3. Havuzlama Katmanı Örneği	9
Şekil 2.4. Doğrultulmuş Doğrusal Birim Etkinleştirme Fonksiyonu.....	11
Şekil 3.1. Bir Basit RDF Örneği	13
Şekil 3.2. RDF Tabanlı Graf Örneği	13
Şekil 3.3. Makine Öğrenmesinde RDF İşlemi için Genel Bir Boru Hattı	16
Şekil 3.4. Kelime Yerleştirme Vektörleri Örneği	19
Şekil 4.1. Atıf Ontolojisi Nesne Özellikleri	23
Şekil 4.2. Atıf Ontolojisi Veri Tipi Özellikleri	23
Şekil 4.3. Atıf Ontolojisi Sınıf ve Özellikleri	24
Şekil 4.4. Anlamsal Arama Yaklaşımının İşlem Adımları	25
Şekil 4.5. Giriş Ekranı	31
Şekil 4.6. Yazar Ekleme Ekranı	32
Şekil 4.7. Makale Ekleme Ekranı.....	32
Şekil 4.8. Dergi Ekleme Ekranı	33
Şekil 4.9. SPARQL Sorgu Ekranı	33
Şekil 4.10. Örnek Anlamsal Arama Sonuçları.....	34
Şekil 4.11. SPARQL ile Yapılan Aramaların Benzerlik Puanları	37
Şekil 4.12. Önerilen Anlamsal Arama Şeması ile Yapılan Aramaların Benzerlik Puanları	39
Şekil 4.13. Gerçekleştirilen Deneylerde Kullanılan Arama Cümleleriyle Eşleşen Makale Sayıları	41
Şekil 4.14. Eşleşen Makalelerin Arama Zamanlarının Karşılaştırılması	42

TABLULAR DİZİNİ

	Sayfa
Tablo 4.1. Atıf Ontolojisindeki Yazar Sınıfına Ait Özellikler.....	24
Tablo 4.2. Makale Sınıfına Ait Bir Örnek Olarak Verilen Makalenin Özellikleri.....	29
Tablo 4.3. Makale Sınıf Örneğine Ait Bileşenler ile Arama İfadesi Arasındaki Benzerlikler	30
Tablo 4.4. Aranılan Cümle ile Örnek Makale Arasındaki Benzerlik Sonuçları....	30
Tablo 4.5. Açık Akademik Topluluk Atıf Veri Kümesi Özellikleri	35
Tablo 4.6. SPARQL Sorgu Dilini Kullanan Arama Sonuçları	36
Tablo 4.7. Önerilen Arama Şemasını Kullanarak Arama Sonuçları.....	37
Tablo 4.8. Seçilen Arama Cümleleri İçindeki Kelimelerin Anlamsal Olarak Benzer Kelimeler Arasındaki Benzerlikler	39

1. GİRİŞ

İnternette artan dijital bilgi hacmi, dijital bilgilere erişim yöntemlerini yeniden şekillendirmektedir. Bu web tabanlı ortamda, bilgi kaynaklarına ve hizmetlerine gerektiğinde erişilebilir ve sunulabilir. Bu nedenle kütüphanelerin hizmetleri dört duvarla sınırlı olmayıp yerel, bölgesel, ulusal ve uluslararası ağlarla bütünleştirilebilir. Web ortamında dijital kütüphaneleri dijital içeriğin işlenmesini ve kullanılabilirliğini daha etkili bir şekilde geliştirmek için anlamsal web'i kullanması için yeni teknolojiler ve stratejiler sunulmaktadır. Web teknolojisinin potansiyeli ile birlikte dijital kütüphane araştırmacılarına web uygulamalarından yararlanma ve kullanıcılar için sınırsız dijital içerik sunma için açık bir alan sunulmaktadır.

Dijital kütüphaneler bilgi dijital formatlarda (yazdırma, mikro form veya diğer ortamların aksine) depolanarak bilgisayarlar tarafından erişilebilen bir bilgi sistemi temelindedir. Dijital kütüphane ortamında; bilgi kaynaklarının etkili bir şekilde aranmasını, erişilmesini ve alınmasını sağlamak için, dijital kayıtlar için farklı üst veri şemalarının yanında, klasik yöntemlere, yaklaşımlara ve araçlara dayanan bibliyografik sınıflandırma şemaları gibi çeşitli üst veri şemaları kullanılabilir. Ayrıca her belgeye, belgenin içeriğini veya bilgi amacını yansıtan uygun bir sınıflandırma sistemi endeksi eklenebilir ve böylelikle belgeleri (hem fiziksel hem de sanal olarak) konuya ve tematik olarak bir araya getirmek için kullanılabilir (Solodovnik, 2011).

Kütüphanelerin bakış açısından, geleneksel kütüphane içeriklerini endekslemenin ötesine geçen ve aynı zamanda web'teki içeriği de içeren çözümler sunmak önemlidir (Lewandowski ve Mayr, 2006; Lossau, 2004). Kullanıcı deneyimiyle ilgili olarak, kütüphanelerde ayrıca web arama motorları da kullanılmaktadır (Sadeh, 2007).

Son yıllarda sayısal kütüphanelerde yayımlanan bilimsel makale sayısı hızla artmıştır. İstatistiksel yöntemler kullanılarak makaleler üzerinde sorgulama işlemi sonucunda sınırlı bilgiler elde edilir. Çünkü sadece makaleler ile ilgili sözcükler ve bunları temsil eden etiketler arasındaki ilişkiler ele alınmaktadır. Bununla birlikte

cümlenin gerçek anlamını yorumlayarak kelimenin konumunun cümle anlamına etkisi tespit edilemez. Genel olarak makaleler, doğal dil kullanılarak yazılmış olduğundan makalenin anlamsal gösterimini elde etmek için makaleye ait anlamsal bilgiler ile birlikte özel tekniklerin kullanımına gereksinim duyulmaktadır (Hirschberg ve Manning, 2015). Mevcut yayın sisteminin en önemli eksikliği, bilimsel makalelerin otomatik bir şekilde işlenebilecek, bir araya getirilebilecek ve yorumlanabilecek biçimsel anlamsallıkları içermemesidir.

Anlamsal yayıncılık (Shotton, Portwin, Klyne ve Miles, 2009; Shotton, 2009; Waard, 2010) anlamsal web ve ilgili alanların kavram ve araçlarını kullanarak bu bilimsel iletişim sorununu çözmek için genel bir yaklaşımdır. Semantik tabanlı kütüphaneler geliştirmek için RDF ve web ontolojileri kullanılmalıdır ve semantik arama sunarak web’de mevcut bilgilere erişim sağlamalıdır (Garcia, Gomez, Colomo ve Garcia, 2011). Anlamsal Web (Semantic Web, SW) Teknolojileri ve Derin Öğrenme ile akıllı yapay nesnelere yaratmayı hedeflemektedir. Her iki teknoloji, veri temsiline olduğu kadar veri ve bilgi analizinde de önemli sonuçlar sağlamaktadır. Çünkü ifade edilebilir dilsel olayları modellemek ve anlamsal olarak karmaşık problemleri çözmek için iki tamamlayıcı yol oluşturmaktadır. Anlamsal yayıncılığın amacı, bir belgenin anlamsal olarak, makine içinde okunabilir biçimde, argümanların metinde nasıl modelleneceğini göstermektir.

Anlamsal web, web sayfalarının içeriğine daha anlamlı ve anlaşılabilir bir yapı getirerek bilgisayarların insanlar için karmaşık görevleri yerine getirebilmesini sağlayacaktır (Berners-Lee, Hendler ve Lassila, 2001). Anlamsal web’in temel bileşeni olan ontolojiler, varlıkları temsil etmek ve varlıklar arasındaki anlamsal ilişkileri ve ilişkilerin çeşitliliğini tanımlamaktadır. Anlambilimsel ilişkiler, iki veya daha fazla kavram arasında veya; varlık veya varlık kümeleri arasındaki anlamlı ilişkilere (Khoo ve Na, 2006). Manuel olarak ontoloji oluşturma maliyetlidir. Bu nedenle verilerden (genellikle doğal dil metni) ontolojileri otomatik olarak çıkarmak için bir dizi sistem önerilmiştir. Bu sistemler ontoloji mühendislerine öğrenilmiş ontolojiler sağlayarak ontoloji yapma sürecini başlatır. Ontoloji öğrenme sistemi genellikle terimler, eş anlamlılar, kavramlar, sınıflandırmalar vb. verilerden

ayıklamak için farklı veri kaynakları ve teknikleri kullanan büyük ve karmaşık çerçevelerdir.

Kelime vektörleri; her bir kelimenin dağılım hipotezine göre kelimelerin bir vektörle temsil edilmesini sağlar. Kelime vektörlerinin çok düşük boyutlu olması (low dimensionality) ilgisiz kelimelerin bir birine yakın dağılımına ve semantik anlamın kaybolmasına, çok fazla boyut olması (high dimensionality) ise hesaplama karmaşıklığının artmasına sebep olmaktadır. Kelime yerleştirme; kelimelerin vektör temsillerinin semantik anlamını kaybetmeden, boyutun indirgenmiş olarak temsil edilebilmelerine imkân verir. Bilgi çıkarımı, doküman sınıflandırma, soru-cevap sistemleri, varlık ismi tanıma sistemleri gibi Doğal Dil İşleme (Natural Language Processing - NLP) uygulamalarında kelime yerleştirme sıkça kullanılmaktadır.

Kelime vektörü algoritması olan Word2Vec, büyük metin veri kümeleri için sürekli vektör temsillerini hesaplamaktadır. Word2Vec, sözdizimsel ve anlamsal benzerlikleri ölçmek için yüksek performans sağlar (Mikolov, Chen, Corrado, and Dean, 2013a).

Makale bilgileri anlamsal olarak ontolojilerde saklanabilir ve metin verilerini cümle içindeki konumlarını göz önüne alarak kabiliyetine göre analiz etmek için derin öğrenme yöntemleri kullanılmaktadır. Böylelikle anlamsal bilgiler, cümle içerisinden çıkarılabilir (Cambria, 2016; LeCun, Bengio ve Hinton, 2015; Majumder, Poria, Gelbukh ve Cambria, 2017; Tang, Quin ve Liu, 2015). Bunun sonucunda dijital kütüphaneler üzerinde anlamsal tabanlı bilgi elde edilmesi son kullanıcılara daha iyi hizmetler sağlanacaktır.

Derin öğrenme mimarisinde kullanılmak üzere hazırlanmış ontoloji tabanlı atıf veri kümesi çok az sayıdadır. Tez çalışmasında ilk olarak amaçlanan Kastamonu Üniversitesi'nde görev yapan akademik personel tarafından yayınlanan makaleleri kapsayan atıf ontolojisi tanımlamak için en uygun yöntemi seçerek anlamsal bilgileri bilimsel makalelerden elde etmek için kullanılan mevcut derin öğrenme teknikleri incelenmiştir. Ayrıca bu makaleleri ve derin öğrenme yaklaşımları kullanılarak elde edilen bilgileri depolamak için bir ontoloji uygulanmıştır. Bu ontoloji üzerinde

anlamsal sorgular gerekleřtirilmiřtir. Geliřtirilen sistem sayesinde anlamsal olarak ilgili makaleler aranarak ve mevcutta yayımlanmıř farklı makaleler ve yazarlar arasındaki iliřkileri kullanarak Kastamonu niversitesi'ndeki tm arařtırmacılara nemli katkılar saęlayacaktır.

Birinci blmde tez alıřmasının fikir temelleri ve yapılan alıřmanın kısaca ifadesini ieren giriř blm bulunmaktadırdır. Tezin ikinci blmnde alıřma iin gerekli derin ęrenme teknolojileri hakkında bilgi verilmektedir. Tezin kapsamına uygun olarak derin ęrenme topolojisi ve uygulamaları, derin ęrenme mimarisi iin gerekli bazı katmanlar ile ilgili literatr taraması yapılarak bilgi verilmektedir. nc blmde teze eřlik eden, anlamsal web teknolojileri ile derin ęrenme uygulamaları birleřtirilerek anlamsal derin ęrenme anlatılmıřtır. Drdnc blmde materyal ve metot blm verilmiřtir. Bu blmde RDF tabanlı atıf ontolojisi geliřtirilmiř ve anlamsal arama modeli oluřturulmuřtur. Oluřturulan anlamsal arama modeli temelinde anlamsal arama vektr ıkarılmasıdır. Ayrıca bu iřlemler iin sistem tasarlanmıřtır. Bu sistem zerinde hem SPARQL hem de kelime vektrlerinin kullanımı ile ilgili sonulara ait bir deęerlendirme sunulmuřtur. Beřinci blm olan son blmde ise sonu ve alıřmayı ileriye gtrebilecek neriler yer almaktadır.

2. DERİN ÖĞRENME

Araştırmacıların manuel olarak inşa etmek zorunda oldukları anlambilimsel bilgilerinin bir kısmı istatistiksel ve makine öğrenme yöntemlerinin hesaplama gücünden yararlanılarak otomatik olarak öğrenebilen ve anlambilimsel olarak web verilerini anlama ve işleme ölçeği büyük ölçüde artan sistemler geliştirilmektedir (Bernstein, Hendler ve Noy, 2016).

Bir makine öğrenmesi algoritması, verilerden öğrenebilen bir algoritmadır. Öğrenme, bir görevi yerine getirme yeteneğini elde etmenin bir yoludur. Makine öğrenmesi (Machine Learning, ML), geliştiriciler tarafından yazılan ve tasarlanan, sabit programlarla çözülmesi çok zor olan görevleri tamamlamayı sağlar. ML görevleri genelde, sistemin bir örneği nasıl işleyeceği açısından açıklanmaktadır. Veriler içerisindeki her bir örnek, öznitelik topluluğudur. Bir görüntünün öznitelikleri, görüntü içerisindeki piksellerin değeri olarak nitelendirilebilir. Birçok problem makine öğrenmesi sayesinde çözülebilmektedir. Bunlardan birkaçı sınıflandırma, regresyon, makine öğrenme ve yapısal çıktı problemleridir.

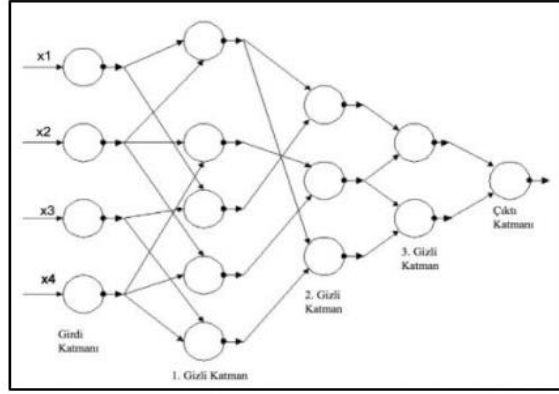
Makine öğrenmesi, girdi ve çıktılarının çok büyük olduğu yüksek boyutlu verileri işleme yeteneği konusunda sınırlıdır. Bu verileri işlemek problem karmaşıklığı arttıkça performans ve doğruluk açısından daha da zorlu hale gelmektedir. Bu aşamada derin öğrenme, görüntü ve metinle ilgili karmaşık gerçek dünya problemlerine çözüm getirmektedir. Görev karmaşılaştıkça nöronlar daha karmaşık bir ağ yapısı oluşturarak bilgi aktarımı sağlarlar. Derin öğrenme yöntemleri, veri setinden daha karmaşık ilişkiler çıkarma yeteneğine göre, diğer makine öğrenme tekniklerine kıyasla bilgi elde etmede önemli ölçüde daha iyi performans göstermiştir (Cortes, Gonzalvo, Kuznetsov, Mohri ve Yang, 2016; Esfe, Ahangar, Rejvani, Toghraie ve Hajmohammad, 2016). Derin öğrenme algoritmaları Yapay Sinir Ağlarının (YSA) yapısal olarak daha karmaşık hali olarak düşünülebilir (Rende, Bütün ve Karahan, 2016).

Derin öğrenme teknikleri, bir insan beynindeki biyolojik nöronlar arasında değiştirilen elektrik sinyallerini matematiksel olarak benzetim yapan yapay sinir

ağlarına dayanmaktadır. İnsan beynine benzer şekilde, yapay sinir ağları da birbirlerine bağlanan nöronlardan oluşur; bu bağlantıların sinir ağları tarafından alınan kararları girdi değerlerine göre tanımlamaktadır (Jain, Mao ve Mohiuddin, 1996). Her bağlantı, bir nöronun diğerinin girişine iletilen çıktının etkisini ayarlayan matematiksel bir değer olan ağırlık olarak bilinir. Her ağ, her katmanın bir dizi nöron içerdiği bir katman kümesinden oluşur. Bir sinir ağındaki ilk katman, bu katmandaki nöron sayısını dış dünyadan toplanan girdi sayısına eşit olması gereken girdi katmanı (input layer) olarak bilinir. Ağdaki son katman çıkış katmanı (output layer), ağda hesaplanan çıktıyı dış dünyaya iletir. Çıkış katmanındaki nöronların sayısı, sinir ağından istenen çıkış sayısına eşit olmalıdır. Daha fazla esneklik için, ikisinin arasındaki bilgi akışını sağlayan saklı katmandan (hidden layer) oluşmaktadır. Derin öğrenme mimarisinde giriş ve çıkış katmanları arasında birden fazla gizli katman bulunmaktadır (Deng ve Yu, 2014; Schmidhuber, 2015; Wang, Mao ve Yi, 2017).

Yüksek boyutlu veri analizlerinde sınıflandırıcının işini kolaylaştıran özniteliklerin bulunması için öznitelik seçme yöntemleri geliştirilmiştir. Öznitelik seçimi, hesaplama süresini azaltan, öğrenme doğruluğunu artıran ve öğrenme modeli veya verileri için anlamlandırmayı kolaylaştıran alakasız ve gereksiz verileri ayıklamak için etkili bir yol sağlar (Bengio, 2009).

Sınıflandırma ve tahminleme için derin öğrenme yöntemleri, her ikisi de performansta iyi sonuçlar gösteren bilgisayar görme ve doğal dil işleme dâhil olmak üzere çeşitli alanlarda uygulanmıştır (Ngiam, Khosla, Kim, Nam, Lee ve Ng, 2011). Denetimli derin ileri beslemeli çok katmanlı perceptron tipinde ilk DL sistemi, Ivakhnenko ve Lapa tarafından 1965 yılında yayınlanmıştır (Ivakhnenko ve Lapa, 1965). Bu çalışmada, her katmanda, en iyi özellikler matematiksel işlemlerle ayrıştırılıp bir sonraki katmana iletilmektedir. Ağları uçtan uca eğitmek için geri yayımlı algoritma yerine önceki katmanlardan sonraki katmanlarda en küçük kareler yöntemi kullanılmıştır. Şekil 2.1'de Ivakhnenko tarafından eğitilen ve bilinen ilk derin ağ mimarisi gösterilmiştir (Dettmers, 2015).



Şekil 2.1. İlk derin ağ mimarisi

2.1. Derin Öğrenme Topolojisi ve Uygulamaları

Derin öğrenme, insan beyninin karmaşık problemleri çözmek için kullandığı yöntem ve yeteneklerini örnek olarak alan, büyük miktarda veriden faydalanarak özellik çıkarma, sınıflandırma ve dönüştürme işlemlerini yapma yeteneğine sahip bir makine öğrenmesi tekniğidir. Makine öğrenme tekniklerine benzer şekilde derin öğrenme; denetimli, denetimsiz ve yarı-denetimli problemleri çözmek için uygulanabilir (Rasmus, Berglund, Honkala, Valpola ve Raiko, 2015). Derin öğrenme, büyük miktarda denetimsiz veri kullanarak sınıflandırma problemlerinde özellik çıkarımı maliyetini ortadan kaldırmış, yapay sinir ağlarının özelleştirilmiş pek çok gizli katmandan ve işlem elemanından oluşan bir çeşididir.

Her problem derin öğrenme ile çözülmeye uygun olmayabilir. Bunun için öncelikle problemin belirlenmesi ve derin öğrenme algoritmalarının kullanımına uygunluğunun değerlendirilmesi gerekir. Eğer uygun ise derin öğrenme için belirli süreçlerin sırasıyla yapılması gerekir. Derin öğrenme süreçleri aşağıdaki şekilde verilmiştir (Rasmus vd. 2015):

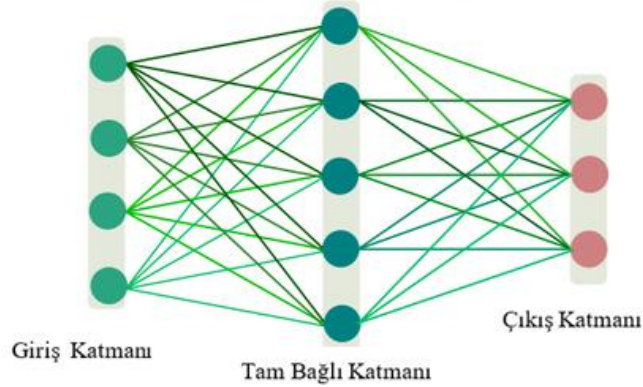
- Problemin tanımı ve derin öğrenme ile çözümünün uygun olup olmadığının tespiti.
- İlgili veri kümelerinin tanımı ve analize hazır hale getirilmesi.
- Uygun derin öğrenme mimarisinin seçilmesi.
- Veri seti ve seçilen derin öğrenme mimarisi kullanılarak sistemin eğitilmesi.

- Eğitilen sistemin eğitiminde kullanılmayan test verileriyle performansının test edilmesi.

2.2. Derin Öğrenme Mimarisi İçin Gerekli Bazı Katmanlar

2.2.1. Tam Bağlı Katmanlar

Bir nöronun tamamen bağlı bir tabakadaki girdileri, önceki tabakadaki nöronların çıktılarından toplanır. Şekil 2.2’de gösterildiği gibi, bir nöronun çıktısı, bir sonraki katmandaki bir nöronun girişi olarak verilmeden önce belirli bir ağırlık değeri (w_i) ile çarpılır (Albawi, Mohammed ve Alzawi, 2017). Her katmandaki nöronlar, daha karmaşık bir özellik saptamak için önceki katmanda tespit edilen özellikleri, çoklu nöronlardan birleştirir. Yapay sinir ağının çoğu, özellikle ağdan tek boyutlu bir çıktı gerektiğinde, başka tabaka türleri kullanılsa bile, bu tür tabakaları içerir. Tamamen bağlı katmanlardan oluşan bir sinir ağı, genellikle her bir örneğin sabit boyutta bir öznitelik değerleri vektörü kullanılarak açıklanan verileri işlemek için kullanılır (Sainath, Vinyals, Senior ve Sak, 2015).



Şekil 2.2. Tam bağlantılı katmanlardaki nöronlar ve ağırlıklar

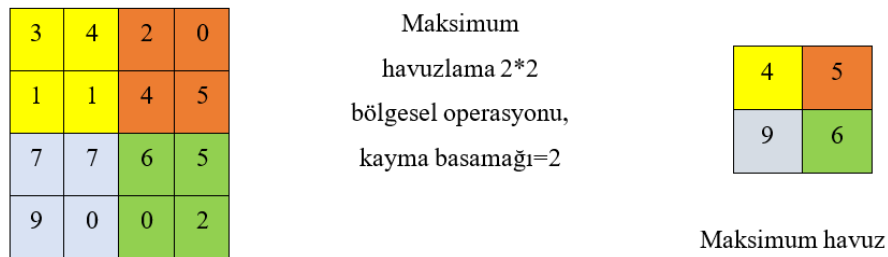
2.2.2. Evrişim Katmanı

Evrişimli bir tabakada, bir nöronun ağırlıkları, iki boyutlu girdi boyunca kıvrılmış iki boyutlu bir filtreden yayılır. Her bir evrişim başına, nöronun çıkışı, evrişimdeki filtrede bulunan giriş değerlerine göre hesaplanır. Bu nedenle, tüm girdi üzerinde katlanmalar tamamlandıktan sonra o nöronun elde edilen çıktı iki boyutlu bir

matristir. Evrişimli bir tabakadaki nöronların, yerel filtrelemedeki değerleri arasındaki ilişkilerin girdi boyutlarından ziyade birbirlerine göre tanımlandıkları yerel iki boyutlu özellikleri tespit etmeleri sağlanmaktadır. Bir evrişimli katmandaki nöronların filtrelerinin boyutuna ek olarak, filtrenin her iki ekseninde, adım olarak bilinen, hareket ettiği değerlerin sayısı tanımlanmalıdır. Böylece nöron tüm girişi tarayabilir ve çıktıyı hesaplayabilir. Adımların değerlerini artırmak, nöronun girdi boyutlarıyla karşılaştırıldığında çıktı boyutlarını azaltır. Bununla birlikte, bu artış, belirli pozisyonlarda belirli özelliklerin eksik olma ihtimalini de artırır. Bu nedenle, evrişimsel bir tabakadaki çıkışların boyutsallığını azaltmak için, evrişimsel tabakadaki nöronların basamak değerlerini arttırmadan başka işlevler kullanılır (Wei vd. 2016). Bu işlevler havuzlama işlevleri olarak bilinir ve ağırlık değerleri yoktur, yani eğitilemezler.

2.2.3. Havuzlama Katmanı

Havuzlama katmanları genel olarak verinin boyutunu indirgemeye yönelik katmanlardır. Genel olarak havuzlama katmanında kullanılan yöntem belirlenen çerçevede maksimum değeri seçme işlemidir. Buna ek olarak belirlenen çerçevede ortalama değeri alma, minimum değeri seçme gibi işlemleri gerçekleştiren havuzlama katmanları bulunmaktadır. Şekil 2.3’de, örnek bir havuzlama katmanı işlemleri gösterilmiştir. İlgili şekilde maksimum değeri seçen bir havuzlama işlemi tercih edilmiştir (Szegedy, Vanhoucke, Ioffe, Shlens ve Wojna, 2016).



Şekil 2.3. Havuzlama katmanı örneği

Ayrıca evrişimli tabakalar, bu girdilerin her biri iki boyutlu ayrı ayrı tabakalar olarak kabul edildiği üç boyutlu girdilerin işlenmesi yeteneğine de sahiptir. Bu katmanlar arasındaki özellikler, son katlamalı katmanın çıktısının düzeltilmesinden sonra

birleştirilebilir ve farklı katmanlardaki özelliklere sahip nesnelere tespit etmek için tam bağlantılı bir katmana bağlanabilir. Bununla birlikte, üç boyutlu girişlerde girişler görüntüler ise her görüntünün üç ana renk kanalı kullanılarak tanımlandığı evrişimsel sinir ağları (Convolutional Neural Network, CNN) ile yaygın olarak kullanılır. Metin işlemede, girdiler iki boyutlu diziler olarak evrişimsel sinir ağına iletilmektedir. Burada evrişimsel katmanların yerel özelliklerini saptama kabiliyeti girdilerdeki metnin anlamsal analizine izin verir (Long, Shelhamer ve Darrell, 2015).

Zhang vd. çalışmasında karakter düzeyinde metin sınıflandırması için derin bir kıvrımlı sinir ağı uygulamışlardır (Zhang, Zhao ve LeCun, 2015). Uygulanan sinir ağındaki ilk katmanlar, bir sonraki katmanlardaki karakterleri tespit etmek için birleştirilebilecek özellikleri algılamaktadır. Algılanan harfler daha sonra her bir nöronun anlamsal olarak benzer anlamı olan bir veya daha fazla kelimeyi tespit edebildiği sonraki katmanlardaki kelimeyle birleştirilir. Daha derin evrimleştirilmiş katmanlar, bu sözcükleri sonunda cümle halinde bir araya getirilen öbekleri birleştirmeyi öğrenir. Bu işlemler, ilk iki ve son katlanma katmanından sonra yerleştirilen en fazla üç havuzlama katmanının yanı sıra altı katlamalı katmanda uygulanır. Son maksimum havuzlama katmanının çıktısı daha sonra düzleştirilir ve tam olarak bağlanmış üç katmana iletilir; buradaki çıkış katmanındaki nöronların sayısı, sınıflandırıcıdan istenen çıkış sayısına bağlıdır. Çalışmanın sonuçları, evrişimli sinir ağlarının metin sınıflandırmadaki performansının istatistiksel tekniklerle ve diğer derin öğrenme temelli yöntemlerle rekabet ettiğini göstermektedir. Ayrıca, çalışma aynı zamanda eğitim için kullanılan verilerin boyutunun sinir ağının performansı üzerinde önemli bir etkisi olduğunu göstermektedir.

Poria vd. çalışmasında kısa cümlelerden fikir madenciliği için evrişimli bir sinir ağı kullanılmıştır (Poria, Cambria ve Gelbukh, 2016). Cümle içindeki her bir kelime için, evrişimli sinir ağı önceki ve sonraki kelimeleri, beş kelimedenden oluşan bir pencere kullanarak, yani önceki kelimeye ve orta kelimeye ilaveten iki kelimeyi kullanarak toplanır. Bu kelimeler, düzleştirilmiş ve tek bir tam bağlı katmana bağlanmadan önce, her biri bir maksimum havuzlama katmanı izleyen iki evrişimli katman kullanılarak işlenir. Ağı çıktısı katmanı, giriş metninin görünüşünü temsil etme

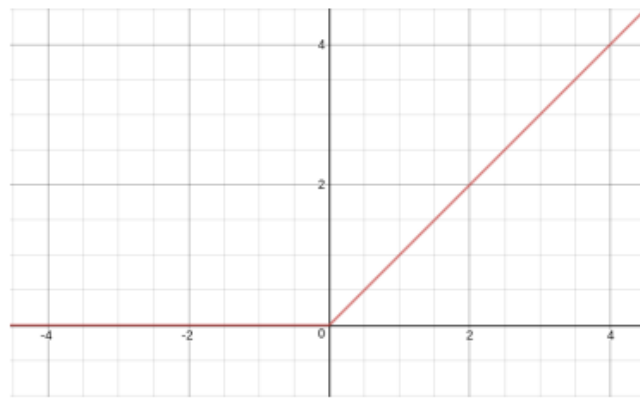
olasılığını belirleyen tek bir nörondan oluşur. Sonuçlar kelimelerin yerleşiminin cümlelerin genel semantik anlamı üzerindeki önemine göre, istatistiksel tekniklerle karşılaştırıldığında bu tür sinir ağlarının üstünlüğünü göstermektedir.

2.2.4. Düzeltilmiş Doğrusal Katman

Düzeltilmiş Doğrusal Katman (Rectified Linear Units, ReLU), geleneksel olarak derin bir sinir ağındaki gizli katmanlar için bir aktivasyon fonksiyonu olarak kullanılır (Agarap, 2018). 2011 yılında derin sinir ağlarının eğitimini daha geliştirdiği gösterilmiştir. Denklem 2.1’de gösterildiği şekilde değerleri 0’a eşleştirerek çalışır.

$$f(x)=\max(0, x) \quad (2.1)$$

Denklem 2.1’de $x < 0$ olduğunda 0, tersine $x \geq 0$ olduğunda doğrusal bir fonksiyon vermektedir.



Şekil 2.4. Doğrultulmuş doğrusal birim etkinleştirme fonksiyonu

Şekil 2.4’de, Denklem 2.1’e uygun olarak $x < 0$ olduğunda fonksiyon çıktısı olarak 0, sonrasında $x \geq 0$ olduğunda eğitim 1 olan bir doğrusal üretmektedir.

3. ANLAMSAL DERİN ÖĞRENME

3.1. Anlamsal Web Teknolojileri

İnternette mevcut olan çok sayıda farklı veriyi paylaşmak için birleşik bir yönteme duyulan ihtiyaç, bu verileri internet üzerinden farklı uygulamalar arasında değiş tokuş etmek için standartların ve uygulamaların tanımlanması ihtiyacını getirmiştir. Anlamsal web, uygulamaların verilere ve aralarındaki ilişkilere erişmesini sağlayan bu standartları ve uygulamaları tanımlar. Yapısal tablolarda verileri temsil etmek yerine, anlamsal web, Web Ontoloji Dili (Web Ontology Language, OWL) kullanılarak depolanan Kaynak Tanımlama Çerçevesi (Resources Description Framework, RDF) kullanır. Veriler, SPARQL (Query Language for RDF) olarak bilinen özel bir sorgu dili kullanılarak RDF'lerden sorgulanmaktadır (W3C, 2019; Wood, Zaidman, Ruth ve Hausenblas, 2014).

3.1.1. RDF

RDF, World Wide Web (Dünya Çağında Ağ, WWW) konsorsiyum tarafından geliştirilmiştir (Koivunen ve Miller, 2001). RDF, özne, yüklem ve bir nesneden oluşan bir üçlü olarak ifade edilir. Bu tür üçlü gruplara RDF grafiği denir. RDF üçlüleri tüm yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış içeriğe eşit olarak uygulanabilir. Yeni türler ve tahminler tanımlayarak, RDF içinde daha anlamlı kelimeler oluşturmak mümkündür. Bu ifade RDF'nin tam anlambilim ile kontrollü kelimeleri tanımlamasını sağlar. Bu özellikler RDF'i farklı veri kümelerinde veri birleştirme ve birlikte çalışabilirlik için güçlü bir veri modeli ve dili yapar (de Vries, 2013). RDF belgeleri XML (eXtensible Markup Language) kullanılarak ifade edilir (Chong, Das, Eadon ve Srinivasan, 2005). RDF ile kullanılan XML, RDF/XML olarak isimlendirilir. Şekil 3.1'de basit bir RDF örneği verilmiştir.

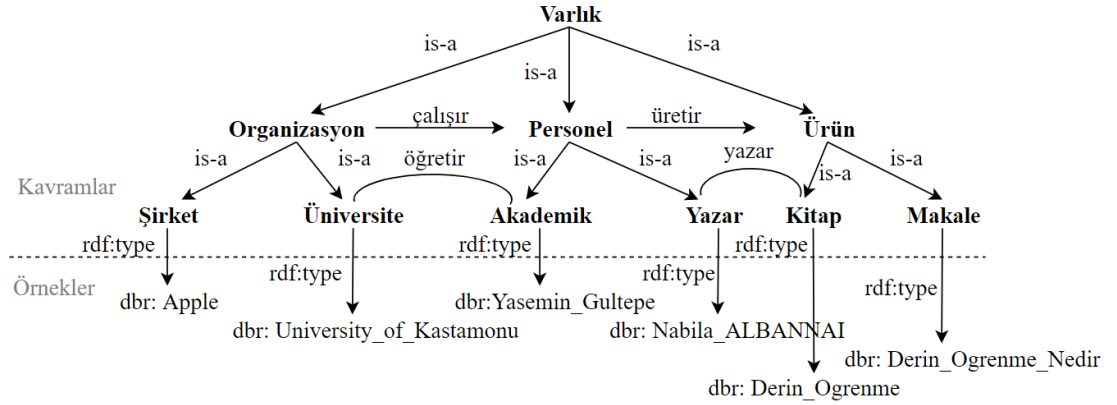
RDF'deki her kaynağa, kaynağın tanımlandığı konumların yolunu tanımlayan Tekdüzen Kaynak Konumlandırıcı (Uniform Resource Locator, URL)'dan ve Evrensel Kaynak Adı'ndan (Uniform Resource Name, URN) oluşan Evrensel Kaynak Tanımlayıcısı (Uniform Resource Identifier, URI) atanır. URL tarafından

tanımlanan konumdaki kaynak için benzersiz bir tanımlayıcı içerir. URI kullanarak, bir uygulama bu kaynağı semantik olarak yorumlamak için bir kaynağın özelliklerini ve bu özelliklerin her biri için değerleri alabilir.

```
<?xml verison="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:uni="http://www.akademik.fake/aakdemik#">
<rdfs:Class rdf:ID="personel">
<rdfs:comment>Personel sınıfı</rdfs:comment>
</rdfs:Class>
<rdfs:Class rdf:ID="akademikPersonel">
<rdfs:comment>Akademik personel sınıfı</rdfs:comment>
<rdfs:subClassof rdf:resource="personel"/>
</rdfs:Class>
<rdfs:Class rdf:ID="OgretimUyesi">
<rdfs:comment>Öğretim Üyesi sınıfıdır. Bütün öğretim üyeleri, akademik personel üyesidir.
</rdfs:comment>
<rdfs:subClassof rdf:resource="akademikPersonel"/>
</rdfs:Class>
<rdfs:Class rdf:ID="ders">
<rdfs:comment>Derslerin sınıfıdır</rdfs:comment>
</rdfs:Class>
<rdf:Property rdf:Id="ogretim">
<rdfs:comment>Öğretim üyelerine dersler atanır.</rdfs:comment>
<rdfs:domain rdf:resource="#OgretimUyesi"/>
<rdfs:range rdf:resource="#ders"/>
</rdf:Property>
</xmlns>
```

Şekil 3.1. Bir basit RDF örneği

Şekil 3.2.'de Öğretim Üyesinin Akademik Personel sınıfının alt sınıfı ve ders ile *öğretim* ilişkisi olduğunu göstermektedir.



Şekil 3.2. RDF tabanlı graf örneği

3.1.2. Web Ontoloji Dili

Anlamsal web'in temel bileşeni ontolojiler, bir alana ait kavramlar kümesini ve kavramlar arasındaki ilişkileri biçimsel olarak tanımlamaktadır. Bir ontoloji herhangi

bir alanda standart olarak kullanılacak ortak ve paylaşılan sözcük kümelerini belirler (McCrae vd. 2012).

OWL; XML, RDF ve RDFS tarafından desteklenen web içeriğinin yazılımlar tarafından daha iyi yorumlanabilmesini, biçimsel bir anlambilimi ile birlikte ek sözcük kümeleri sunarak kolaylaştırır. OWL, RDF/XML sözdizimin temeline dayanan bir dildir (Pascal, Krötzsch, Parsia, Patel-Schneider ve Rudolph, 2012).

OWL, özellikleri ve sınıfları tanımlayabileceğimiz daha geniş bir kelime hazinesi sağlamaktadır. Ontolojilerde en üst sınıf varsayılan olarak Türkçe’de “şey” anlamına gelen “thing”dir. Sınıfların özellikleri tanımlanarak terimlerin dâhili yapılarının meydana gelmesi sağlanmaktadır. Özellikler, sınıflara ait kavramların birbiri ile olan ilişkilerini nitelemek için kullanılmaktadır. Ontolojilerde çeşitli özellikler ve bu kavramlar arasında anlamsal ilişkiler tanımlanmaktadır. İki çeşit özellikleri bulunmaktadır; Nesne özellikleri (object property) ve veri tipi (datatype property) özellikleri. Ontolojilerde iki sınıfın örnekleri arasındaki ilişkiler nesne özellikleri kullanılarak gösterilmektedir. Nesne özellikleri simetrik, fonksiyonel, ters fonksiyonel veya geçişli olabilir.

3.1.3. SPARQL

Ontolojiler üzerinde veri sorgulama işlemi için W3C semantik aktivitesi tarafından da tanımlanan SPARQL yaygın olarak kullanılmaktadır. SPARQL’de, gerekli verilerin modellerini temsil etmek için kullanılan desenler, ayrıca, üçlülerin her bir parçasının, bir URI, değişken olabilen, değişken adının önünde bir soru işareti kullanılarak gösterilen değişkendir. İstenilen şablon, her biri bir nokta ile sonlandırılan, belirli bir üçlüden bir değişken için toplanan değerlerin, bu değişkenin aşağıdaki üçlülerde görüldüğü yerde kullanıldığı, birden fazla üçlü kullanılarak tanımlanabilir (DuCharme, 2013). SPARQL kullanılarak gerçekleştirilebilecek dört ana sorgu türü şunlardır:

- Seçme sorguları: Bu sorgular, seçilen ifadede de belirtilen tanımlanmış kalıplardaki değişkenler için toplanan değerleri döndürür.

- Sorgu tanımlama: Bu sorguların sonuçları, yapı ifadesinde bu üçlülerin tanımına göre üçlü formda döndürülür.
- Sorgu doğrulama: Bu sorgu türü, sorgulanan kalıbın mevcut olup olmadığını belirten Boolean sonuçları döndürür.
- Sorgu etiketleme: Bu sorgular, değişken için döndürülen nesnelere tüm tanımlarını açıklama (annotation) deyiminde tanımlayan üçlülerini döndürür.

Bir sorgunun sonuçları, sorgulama uygulamasının gerekliliklerine göre, toplam, sayım, maksimum, minimum, ortalama ve birleştirme gibi fonksiyonlarda kullanılarak özetlenebilir. Ayrıca, sorguyu tanımlayan kalıplara, mantıksal, matematiksel veya varoluş cümlelerini kullanarak, süzgeç ifadesindeki kurallara uymayan sonuçların sorgu tarafından döndürülen sonuçların dışında bırakıldığı yerlerde filtreler uygulanabilir. SPARQL aracılığıyla veri oluşturma, okuma, güncelleme ve silme işlemleri yapılabilir.

3.1.4. Protégé

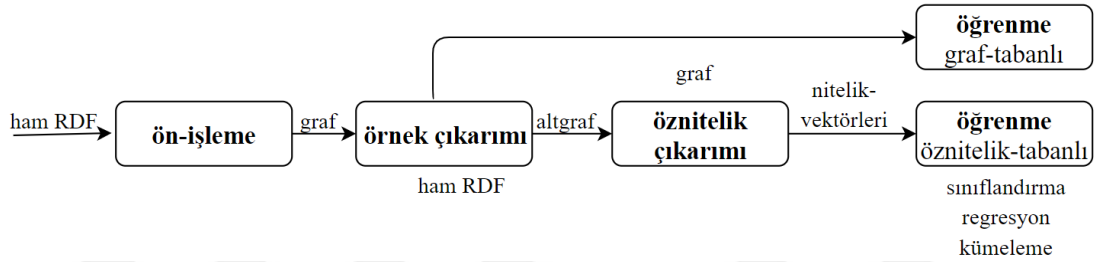
Protégé, Stanford University ile University of Manchester tarafından ortaklaşa olarak Java platform üzerinde geliştirilen bir ontoloji editörüdür. OWL DL düzeyinde destek sağlayan program ile internet üzerinde ulaşılabilecek ontolojiler incelenebilirken sıfırdan bir ontoloji oluşturmak da mümkündür. Pellet ve Fact++ isimli yorumlama motorları ile ontoloji üzerinde açıkça belirtilmemiş olan ilişkileri de kullanıcının dikkat ve kullanımına sunar (Stanford Medical Informatics, 2006). İnternet üzerinden ulaşılan ontolojiler üzerinde çalışabileceği gibi yerel disk sistemi üzerinde de çalışmalar yapılabilir. Birden fazla sayıda ontoloji aynı anda açılıp bağlantılar kullanılarak çalışma yapılabilir. Protégé, eklentilere açık bir mimari ile tasarlanmıştır. Bu eklentilerden biri olan OwlViz eklentisi ile çalışılan ontolojinin görsel olarak ortaya konulması mümkün olmaktadır.

3.2. Anlamsal Derin Öğrenme

Veri madenciliği ve makine öğrenimi tekniklerinin, zengin veriler içeren semantik öğrenmenin performansını artırmak için önemli olduğu bulunmuştur. Bu teknikler temel olarak bu verilerdeki görüntüleri tanıma ve anlambilimsel bilgileri çıkarma

gibi önemli anlambilimsel tanımlama için kullanılır. Ayrıca, verilerin anlambilimi, öneriler, sınıflandırma ve kümeleme gibi diğer uygulamalarda da önemli bir role sahiptir.

Anlamsal verileri derin öğrenme kullanarak işlemek için, özellikler Şekil 3.3'de gösterilen Bloem ve Vries (2014) tarafından sunulan boru (pipe) hattında gösterildiği gibi bir bilgi grafiği kullanılmaktadır (Bloem ve Vries, 2014). Bu boru hattına göre, örnekler bilgi grafiğinden çıkarılır ve ardından bir dizi özellik kullanılarak açıklanır. Her özelliğin değeri, örnek ile grafikteki başka bir nesne arasında tanımlı olan bir ilişkiden elde edilir. Özellik çıkarımının çıktısı, derin öğrenme teknikleriyle kullanılabilir olan özellik vektörleridir.



Şekil 3.3. Makine öğrenmesinde RDF işlemi için genel bir boru hattı

Bilgi grafiğini (Knowledge Graph) özellik vektörlerine dönüştüren ana üç adım bulunmaktadır. Böylece derin öğrenme de dâhil olmak üzere özellik tabanlı makine öğrenme teknikleriyle işlenebilir. Bunlar aşağıdaki şekilde sıralanmaktadır:

- **Ön-İşleme (Preprocessing):** Derin öğrenme tekniğiyle sağlanan bilginin etkinliği, bilgi grafiğinden elde edilen daha az alakalı bilgilerin kaldırılmasıyla önemli ölçüde iyileştirilebilir. Bilgi grafiğindeki bilginin ayrıntılı niteliği, bu işlemin önemini artırır. Çünkü bu tür ayrıntılı bilgiler normalde derin öğrenme uygulaması için önemli bir bilgi içermeyen önemli miktarda bilgi içerir. Derin öğrenme tekniklerinin yalnızca göreceli bilgileri dikkate alma ve bu özelliklerden bilgi çıkarma yeteneğine rağmen, vektörlerin boyutunun azaltılması, eğitim sürecini hızlandırabilir ve daha küçük modeller için kullanılan daha basit modellere göre, her girişi işlemek için gereken süreyi azaltabilir. Yanlış bir özellik seçimi, seçilen özelliklerin kullanımıyla

karakterize edilemeyen önemli bir sorun olan bilgi grafiğinden boş düğümler üretebilir. Bu ilgili özelliklerin seçiminde kaçınılması gereken önemli bir konudur.

- Örnek Çıkarımı (Instance Extraction): Bilgi grafiğindeki örnekler, bu grafikteki diğer örneklerle ilişkilerine dayanarak tanımlandığından, örnek kaynakların diğer örneklerle ilişkileri göz önünde alınmadan kullanımı, derin öğrenme uygulamasının gerektirdiği bilgiden yoksundur. Bu nedenle, örnek ile grafikteki diğer örneklerdeki birinden diğerine olan ilişkiler, örneği tanımlayan özelliklerin değerlerini çıkarmak için araştırılmalıdır. Bununla birlikte, araştırılmakta olan örnekle ilgili örnekler, grafikteki diğer örneklerle ilişkileriyle de karakterize edilir; bu, elde edilen mevcut örneği karakterize etmek için bu örnekleri inceleme gereği getirir. Bu nedenle, örnek çıkarma aşaması, grafiği, örneğin konumundan belirli bir derinliğe kadar inceler (Lösch, Bloehdorn ve Rettinger, 2012; Vries, 2013).
- Öznitelik Çıkarımı (Feature Extraction): Bir örneğin özellik değerlerini içeren vektör, bilgi grafiğinden iki yaklaşım kullanılarak çıkarılabilir. Birinci yaklaşım, bu değerleri tensör çarpanlara ayırma yöntemini kullanarak doğrudan bilgi grafiğinden çıkarır (Nickel, Tresp ve Kriegel, 2011), ikinci yaklaşımda ise bu değerleri Weisfeiler-Lehman algoritması kullanarak önceki adımdan elde edilen örnekten çıkarır (Vries, 2013). Bu özelliklerin bilgi grafiğinden çıkarılan örneklerden çıkarılması, bu örnekler için uygun bir yapı bilindiğinde grafikteki ilgili bilgiler olduğunda daha etkilidir.

Bir modelde, önceki n kelimenin vektör temsilleri girdi olarak alınırsa bu tür kelime vektörleri, kelime yerleştirme olarak bilinir. Kelime vektörleri; kelimeler, cümleler ve belgeler arasındaki ilişkileri analiz etme yeteneğini artırmaktadır. Bunun için makineler kelime hakkında daha önce geleneksel kelimelerin temsillerini kullanarak mümkün olandan çok daha fazla bilgi sağlayarak teknolojiyi ilerletiyorlar.

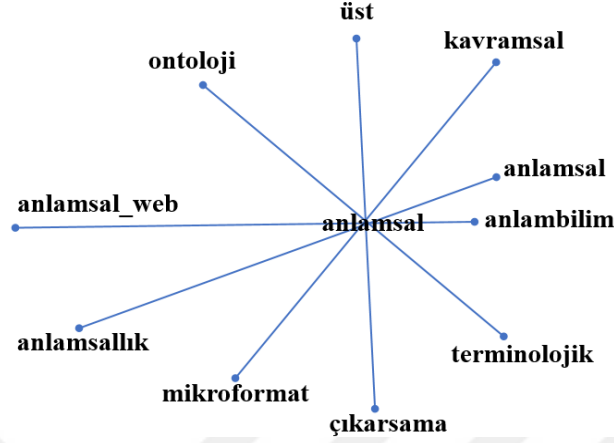
Kelime vektörleri, kelimeleri semantik olarak benzer kelimelerin geometrik uzayda yaklaşık noktalara eşlendiği çok boyutlu sürekli kayan nokta sayıları olarak temsil eder. Daha basit bir ifadeyle, bir kelime vektörü, her noktanın kelimenin anlamının bir boyutunu yakaladığı ve anlamsal olarak benzer kelimelerin benzer vektörlere

sahip olduđu bir dizi gerek deęerli sayılardır (Zhou, Sedoc ve Rodu, 2019). Kelime vektörlerinin ok düşük boyutlu olması (low dimensionality) ilgisiz kelimelerin bir birine yakın daęılımına ve semantik anlamın kaybolmasına, ok fazla boyut olması (high dimensionality) hesaplama karmaşıklığının artmasına sebebiyet verebilir. Örnek olarak tekerlek ve motor gibi kelimelerin araba kelimesine benzer kelime vektörlerine sahip olması gerektięi anlamına gelir (anlamlarının benzerlięi nedeniyle), oysa muz kelimesi anlamsal olarak oldukça uzak olmalıdır. Farklı bir ifadeyle, benzer bir bağlamda kullanılan kelimeler, yakın bir vektör uzayına eşlenecektir.

Kelime vektörleri metin analizi gerekleřtiren özellikle yapay sinir aęlarını temel alan doęal dil iřleme sistemlerinde girdi olarak kullanılır (Wang, Wang, Chen, Wang ve Kuo, 2019). Bu kelime yerleřtirmeleri bir gizli katmanı beslemek üzere bir araya getirilir. Gizli katmanın ıktısı ise ıktı katmanının girdisi olur.

Kelime benzerlik deęerlendiricisi, kelime vektörleri ve insan tarafından algılanan anlamsal benzerlik arasındaki mesafeyi ilişkilendirir. Ama, insan tarafından algılanan benzerlik kavramının kelime vektörü gösterimleri tarafından ne kadar iyi yakalandığını ölçmek ve kelimelerin anlamının içinde buldukları bağlamla ilişkili olduęu daęıtım hipotezini doęrulamaktır. İkincisi için, daęıtık semantik modellerin benzerlięi benzetme biçimi hala belirsizdir (Faruqui, Tsvetkov, Rastogi ve Dyer, 2016). Bu nedenle kelime yerleřtirme, bu kelimeleri sayısal vektörlere dönüřtürmek için kullanılır, böylelikle bu kelimeler arasındaki anlamsal karşılařtırmalar, bu vektörlerdeki sayısal deęerlere dayanarak yapılabilir. Vektörlerde daha benzer deęerlere sahip olan kelimeler, birbirlerinden daha büyük farklılıklara benzerdir. Dolayısıyla, vektörler arasındaki mesafenin hesaplanması benzerliklerini gösterir; burada daha az mesafe daha fazla benzerlięi gösterir ve bunun tersi de geçerlidir. Ancak, her kelime için vektör içindeki uygun özellik deęerleri, bu kelimeler için kesin bir düzen olmadığından tanınamaz. Bu nedenle, bu vektörleri üretmek için bir sinir aęı kullanılır, sinir aęını bir korpus kullanarak eęiterek, kelimelerin anlamsal anlamlarının cümle içindeki konumlarına dayanarak algılanmasını saęlar. Sonunda, sinir aęından ıkarılan örnek vektörler için, vektörlerin konumlandırılması,

kelimelerin belirli bir metindeki anlamsal olarak analiz edilmesi için kullanılabilir (Mikolov vd. 2013a).



Şekil 3.4. Kelime yerleştirme vektörleri örneği

Kelime yerleştirme tekniği ile üretilen vektörleri kullanarak, farklı kelimeler arasındaki Öklid mesafeleri, aralarındaki benzerlikleri ölçmek için hesaplanabilir. Şekil 3.4.'de gösterildiği gibi, “anlamsal” kelimesi ile diğer on sözcük arasındaki mesafeler ölçülür ve seçilen sözcük ile daha anlamsal olarak benzer sözcükler arasındaki mesafe kalan sözcüklerden daha azdır. Bu dağılıma göre, “anlambilim” kelimesi “anlamsal” kelimesinden “ontoloji” kelimesine daha benzer, “terminolojik” kelimesinden daha benzerdir.

Bir sözcük grubundan kelime düğümlerini öğrenmek için, genellikle V benzersiz öğelerden oluşan bir kelimedeki her kelimeyi bir vektördeki benzersiz bir dizine eşleyen bir sıcak kodlama (one hot encoding) ile başlayan birçok farklı yöntem vardır. Bu nedenle bir kelime, uygun pozisyonda bir (1) hariç tüm sıfırlardan (0) oluşan bir vektör ile temsil edilir. Bağlam temelli öngörmeyi öğrenerek, kelime gömme yöntemleri, bu sıcak vektörlerin her birini, boyutları tipik olarak kelime büyüklüğünden çok daha düşük olan ve öğeleri dil verilerinin gizli anlambilimini yakalayan yoğun temsillerle eşleştirir. Bunun mantığı, kelimelerin daha iyi temsil edilmesini öğrenerek daha iyi kelime tahmini yapılabilmesidir. Word2Vec, iki şekilde uygulanabilen tahmin tabanlı bir yöntemdir: Sürekli bir kelime torbası (Bag-of-Words, BoW) ve bir atlama-gram (Skip-Gram (SG)) (Mikolov vd. 2013a; Mikolov vd. 2013b; Mikolov, Yih ve Zweig, 2013c).

Zhao vd. (2015) tarafından Word2Vec modeline dayanan bir yöntem önerilmiştir. Önerilen yöntem ile NLP için yapay sinir ağlarını kullanan farklı teknikler arasında en iyi performans elde edilmiştir (Zhao, Lu ve Pourpart, 2015). Bu model Google'ın haber makalelerini kullanarak ve bir milyar kelimeyle eğitilmiştir. Korpusta 5'ten az tekrarlanan kelimeler ihmal edilir, çünkü bu kelimeler çok sınırlı kullanımda olabilir veya yazım hataları içerebilir. Bu nedenle eğitim veri seti, 692.000 kelimedenden oluşmaktadır. Her bir giriş sözcüğü için, bu sinir ağı matematiksel olarak o kelimeyi temsil eden 300 özelliğe sahip bir vektör çıkarır, böylece daha benzer vektörler, bu vektörler için kullanılan kelimelerin daha benzer duygusal anlamlara sahip olduğunu gösterir.



4. MATERYAL VE METOT

Atıf analizi, bilimsel yayınların niceliksel özelliklerini ve kurallarını ortaya koyan bibliometrik bir analiz tekniğidir. Dergi, makale, yazar ve diğer kaynaklardaki atıfları analiz etmek için matematiksel ve istatistiksel yöntemlerin kullanılmasını içerir. Atıf analizi, onlarca yıllık gelişimi boyunca önemli teorik ve pratik ilerleme kaydetmiştir ve bilimsel bilgiyi değerlendirmek, bilimsel modelleri tanımlamak ve bilim topluluğu tarafından araştırılan yeni sınırları keşfetmek için yaygın olarak uygulanmıştır (Xiao, Shi ve Wang, 2018).

Son yıllarda araştırmacılar, atıf verme davranışlarının ve motivasyonlarının tanımını iyileştirmek için ontoloji, bağlantılı veriler ve diğer teknolojilerden yararlanmak amacıyla atıf analizine anlamsal web teknolojilerini tanıtmaya başladılar.

Dijital kütüphane, ontoloji temelli bilgi hizmeti araştırmasının önemli bir uygulama alanıdır. Patkar (2011) çalışmasında ontolojinin bu dijital çağdaki kütüphanelerden bilgi almak için en yeni araçlardan biri olduğunu belirtmiştir. Bu çalışmada, bilgi yönetim araçlarındaki ilerlemeleri tartışmaktadır ve farklı alanlar arasındaki ontoloji uygulamalarını örneklenmiştir (Patkar, 2011).

Koutsomitropoulos vd. (2013) tarafından Dspace dijital veri havuzu sisteminin anlamsal arama servisini incelemişlerdir. Anlamsal arama sorgulamayı kolaylaştıran ve sistemin tasarımını, performansını ve ölçeklenebilirliğini geliştiren yapılandırılmış bir sorgulama mekanizması sunulmuştur (Koutsomitropoulos, Solomo ve Papatheodorou, 2013). Iorio ve Schaerf (2015) çalışmasında, kaynağa ait üst verilerini tanımlamak için örnek bir dijital kütüphanede tanımlanan bir anlamsal model önerilmiştir. Anlamsal model, üst veri nesnesi açıklama modelinden türetilir. Üst düzey bir kavramsal referans modeli, dijital kütüphane üst verileri için anlamsal web teknolojilerinin uygulanmasını desteklemektedir (Iorio ve Schaerf, 2015).

Ontoloji ve bağlantılı verilere dayanan herhangi bir atıf analizi yöntemi temel olarak şu üç adımı içerir: İlk olarak, bibliyografik atıf verilerine ve tam metin atıf bilgilerine göre atıf ontolojisi oluşturmak; ikincisi, atıf ontolojisini kullanarak referans

bilgilerini normalleştirmek ve verileri RDF modeline göre bağlantılı veriler olarak yayınlamak; ve, üçüncüsü, gerekli atıf bilgilerini çıkarmak için, bir atıf analizi boyutu için belirli bir SPARQL arama sorgusu yazmak ve arama sorgusunu yürütmek. Arama sonuçları daha sonra atıf analizi hedeflerine ulaşmak için görselleştirilir.

4.1. Atıf Ontolojisi

Bu bölümde atıf bilgilerini tanımlamak için göreve dayalı (task based) bir ontoloji oluşturmaktır. Bu tez çalışmasında, ontoloji oluşturulurken Noy ve McGuinness (2000)'ın çalışmasındaki ontoloji geliştirme metodolojisi esas alınarak sistemin ihtiyaçlarına cevap verebilen ontoloji geliştirme aşamaları izlenmiştir. Bu metodolojinin önemli bir özelliği ontoloji nesnelerinin yeniden kullanımını sağlamasıdır.

Ontoloji geliştirme süreçleri 7 (yedi) esas aşamada oluşmaktadır. Bu aşamalar şu şekilde sıralanmaktadır:

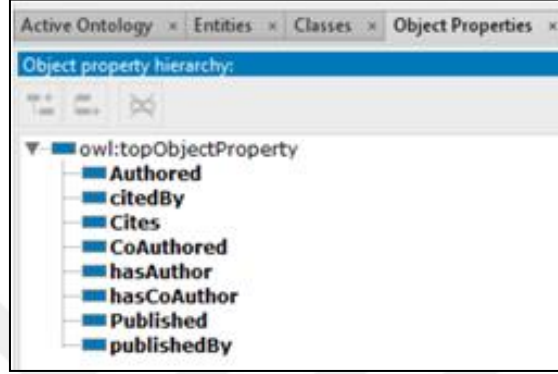
1. Ontoloji kapsamını ve etki alanını tanımlama
2. Ontolojinin yeniden kullanımını sağlama
3. Ontolojideki terimlerin ve terim tiplerinin belirlenmesi
4. Sınıfların tanımlanması ve sınıf sıra düzeninin oluşturulması
5. Sınıfların niteliklerinin tanımlanması
6. Niteliklerin özelliklerinin tanımlanması
7. Sınıf örneklerinin tanımlanması

Sınıfların özellikleri tanımlanarak terimlerin dâhili yapılarının meydana gelmesi sağlanmaktadır. Özellikler, sınıflara ait kavramların birbiri ile olan ilişkilerini nitelemek için kullanılmaktadır.

Ontolojilerde çeşitli özellikler ve bu kavramlar arasında anlamsal ilişkiler tanımlanmaktadır. İki çeşit özellikleri bulunmaktadır; Nesne özellikleri (object property) ve veri tipi (datatype property) özellikleri. Ontolojilerde iki sınıfın

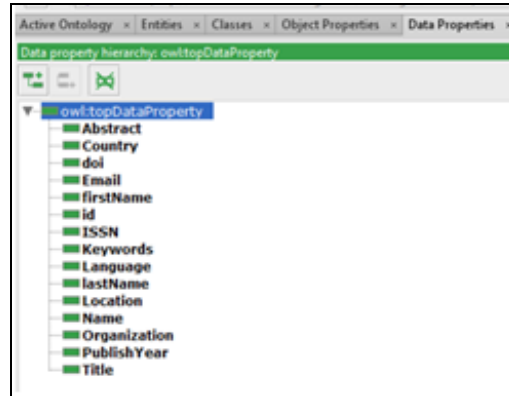
örnekleri arasındaki ilişkiler nesne özellikleri kullanılarak gösterilmektedir. Nesne özellikleri simetrik, fonksiyonel, ters fonksiyonel veya geçişli olabilir.

Nesne özelliği ve veri tipi özelliği tanımlanırken özelliği içerecek sınıf (domain) ve değer aralığı (range) belirtilmektedir. Şekil 4.1’de atıf ontolojisinin nesne özellikleri gösterilmiştir.



Şekil 4.1. Atıf ontolojisi nesne özellikleri

Basit bir veri tipinde değer içeren ve bir sınıfın belirli bir değer özelliğini gösteren özellikler veri tipi özelliği olarak belirlenmektedir. Şekil 4.2’de Atıf ontolojisinin veri tipi özellikleri gösterilmiştir.



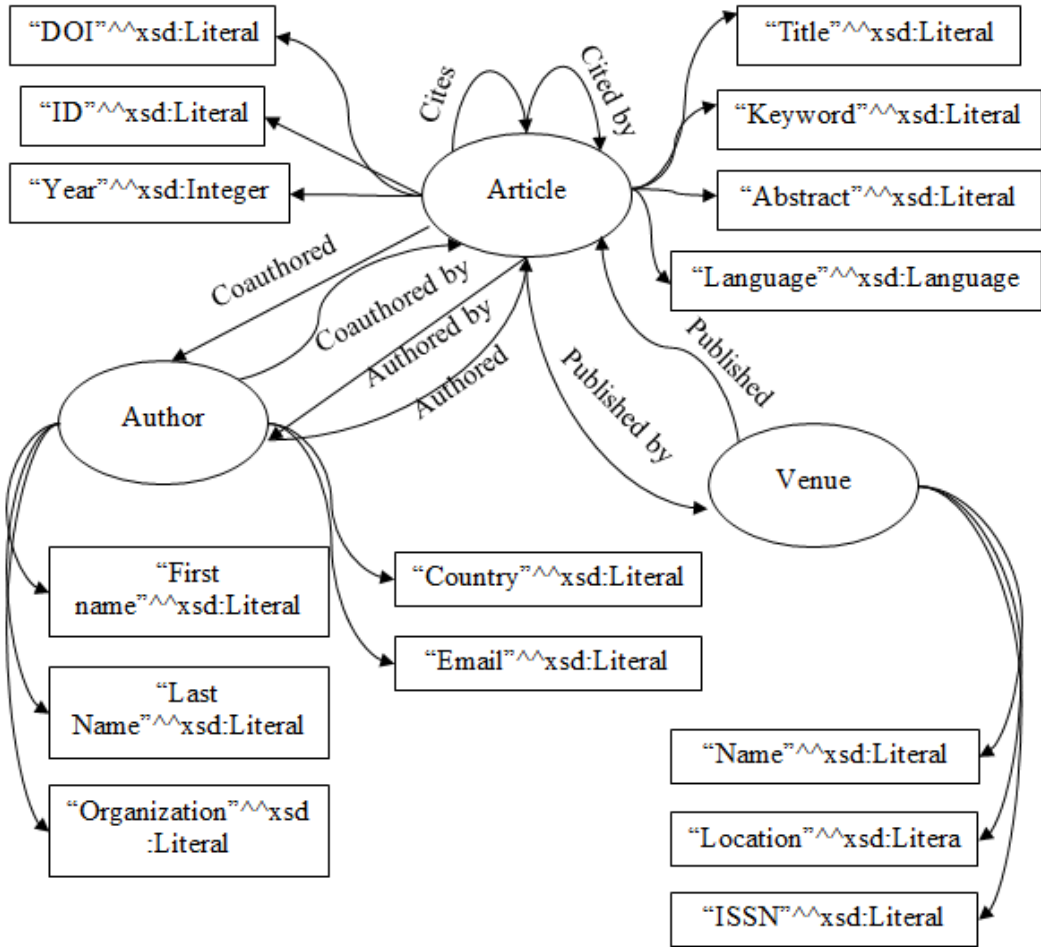
Şekil 4.2. Atıf ontolojisi veri tipi özellikleri

Tablo 4.1’de bir makalenin yazarına ait bilgiler gösterilmektedir. Bu bilgiler şunlardır; adı, soyadı, çalıştığı kurum, yaşadığı ülke, güncel olan email adresi ve yetkileri.

Tablo 4.1. Atıf ontolojisindeki yazar sınıfına ait özellikler

Öznitelikler	Tip
First name	Literal
Last name	Literal
Organization	Literal
Country	Literal
Email	Literal
Authored	Article

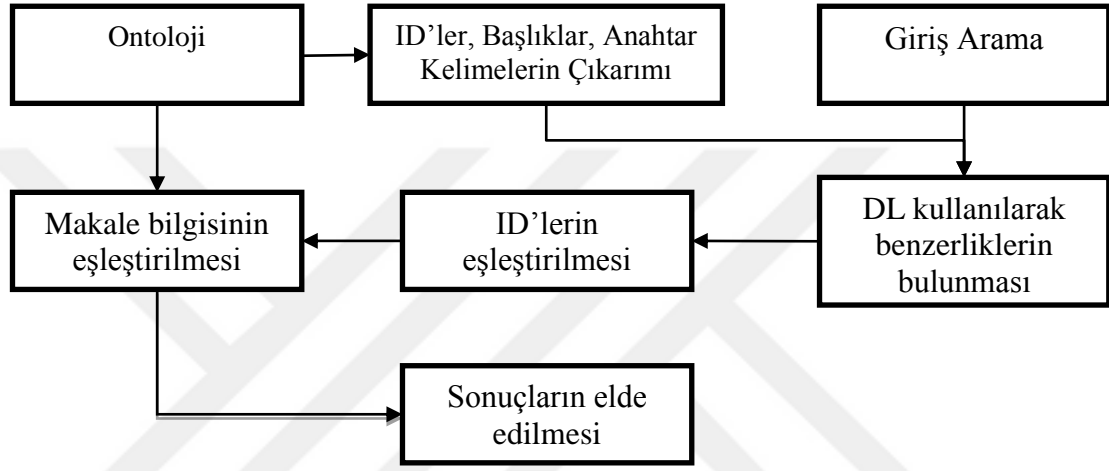
Şekil 4.3’de uygulanan bilgi grafiğindeki nesneleri ve bu nesnelere birbirleri arasındaki ilişkileri ve veri nesnelere tanımlayan özellikleri göstermektedir.



Şekil 4.3. Atıf ontolojisi sınıf ve özellikleri

4.2. Anlamsal Arama Modeli

Bu tez çalışmasında önerilen model, makalelerin bilgilerini depolayan ontoloji veya bilgi grafiği ve arama ifadesine anlamsal olarak benzeyen makaleleri elde etmek için bir derin öğrenme yaklaşımı kullanan bir arama yöntemidir. Şekil 4.4’de tez çalışmasında sunulan ontolojiler üzerinde derin öğrenme tabanlı arama işlemleri için bir akış şeması gösterilmiştir.



Şekil 4.4. Anlamsal arama yaklaşımının işlem adımları

Şekil 4.4’de gösterilen anlamsal arama işlemine ait her bir adım ve aralarındaki ilişkiler (işlemler) şu şekilde ifade edilmektedir:

- **Makale Bilgilerini Almak:** Her makale, belirli özellikleri ile bir nesnedir. Makale nesnesi, *ID*, *title*, *keywords*, *authors*, *date* ve *abstract* gibi özelliklere sahiptir. Bu bilgiler RDF tabanlı olarak ontolojide saklanmaktadır. İstenilen bilgiler SPARQL sorguları kullanılarak alınabilmektedir. Bu tez çalışmasının amacı olan derin semantik tabanlı bir arama işlemi ile birlikte yapıldığında daha etkili ve verimli sonuçlar elde edilir. Veri dosyalarında derin semantik arama için kullanılan ortama göre daha uygun bir formatta saklanır ve tanımlayıcı öznitelikler ile RDF tabanlı ontolojiler ile karşılaştırılarak güncelleme işlemi yapılmaktadır.
- **Açıklama Mantığı (Description Logic, DL) Kullanarak Benzerlikleri Ölçme:** Sistem kullanılarak kullanıcı tarafından arama yapılacak her kelime ile ontoloji

örnekleri üzerinden sorgulanan makalelerin öznitelikleri arasındaki benzerlik oranları hesaplanmaktadır. Sonrasında elde edilen sonuç listesinden aranan kelime grubundaki her bir kelime için arama sonuçlarından en yüksek değere sahip olan arama sonucu en yüksek performansa sahiptir.

- Eşleşen ID'ler: Sorgulanacak cümlesi ile ontolojideki her makale arasındaki benzerlik hesaplamasından sonra, önerilen yöntem eşleşen eşleşmelerin kimliklerini sıralı bir şekilde vermektedir. Böylelikle en çok eşleşme yüzdesine sahip olan nesne ilk sırada yer almaktadır. Elde edilen yüzdeler sıralaması sonucu sistemin herhangi bir uygulama ile kolaylıkla bütünleştirilmesini sağlamaktadır.
- Eşleşen Kimlik Bilgilerini Elde Etmek: Arama ifadesiyle eşleştiği belirlenen makalelerin kimlikleri, önerilen arama yönteminden alınır ve eşleşen makalelerin gerekli bilgilerini almak için kullanılır. Oluşturulan nesnelere gerekli tüm bilgiler varsa, bunları doğrudan önerilen yöntemden almak mümkündür. Aksi takdirde, gerekli bilgiler eşleşen makalelerin kimliklerine dayanarak SPARQL sorguları kullanılarak alınır.
- Sonuçların Gösterimi: Arama sonucunda elde edilen ID'ler eşleşme skorlarına göre sıralanmaktadır.

4.3. Kelime Vektörlerinin Çıkarılması

Makaleler, doğal dil kullanılarak yazılmaktadır. Benzer şekilde arama motorları da doğal dil kullanılarak sorgulamalar yapılmaktadır. Semantik arama terimi arama motorlarının doğal dil kullanılarak yapılan sorgulamaları daha iyi yorumlamaları ve işlemleri için kullanılmaktadır. Ancak semantik arama bundan çok daha fazlası anlamına gelmektedir. Kullanıcının aramayı yaparken sahip olduğu bağlam, bunlardan biridir.

Kelime bağları olarak da adlandırılan dağınmık vektörler, benzer bağlamda görünen kelimeler benzer anlamlara sahiptir. Kelime yerleştirmeler, tipik olarak bir sinir ağı kullanılarak içeriğine dayalı bir kelimeyi tahmine etme görevi üzerinde önceden eğitilir. Kelime vektörleri, sözdizimsel ve semantik bilgileri gömmme eğilimindedir. Kelime vektörleri en basit şekilde kelimelerin birbiri ile olan ilişkilerine

odaklanmaktadır. Bu kelimelerin ilişkilerinden yola çıkarak anlamsal analizler yapılmaktadır.

Bu tez çalışmasında Kelime Vektörleri (Word2Vec) modeline dayanan bir sinir ağı uygulanmaktadır ve arama kelime grubuyla birlikte kelimeler arasındaki makalenin benzerliğini ölçmek için kullanılmaktadır. Giriş olarak girilen kelimeler veya kelime grupları ile makaleye ait “Title”, “Keywords” ve “Abstract” arasında anlamsal olarak yakın olan ifadeler aranmaktadır. Arama kelime grubundaki her bir kelime için, araştırılan makale bileşenlerinden her birinin en yüksek eşleşmesi, yani “Title”, “Keywords” ve “Abstract” değerleri toplamıdır. Sonrasında seçilen değerlerin ortalaması, arama ifadesi ile bileşenlerin her biri arasındaki genel eşleşme ölçüsü olarak hesaplanır. “Title” ve “Keywords”’nin “Abstract” metnine göre önemi; “Title” veya “Keywords”’de eşleşen bir cümleyi içeren bir makale, aynı eşleştirme ölçüsünde olan ancak “Abstract”’te hesaplanan başka bir makaleden daha önemlidir. Böylece, makalenin “Abstract” kullanılarak hesaplanan eşleştirme ölçüsü 0,9 ile çarpılarak “Title” veya “Keywords”’deki eşleşmeler daha fazla vurgulanır.

Son olarak, arama cümlesi ile makale arasındaki genel benzerlik ölçüsü, makalenin üç bileşeni için hesaplanan benzerlik ölçütlerinin en yüksek değeri olarak hesaplanır. Daha sonra, eşik değerine eşit veya ondan daha büyük olan benzerlik önlemlerine sahip makalelerin kimliği alınır ve SPARQL sorgu dili kullanılarak makalenin bilgilerini ontolojiden sorgulamak için kullanılır. Bu makaleler azalan sırayla alınır, yani daha yüksek benzerlik önlemleri alınır ve görüntülenir. Bu tez çalışmasında önerilen arama prosedürünün algoritması şu şekildedir:

Giriş: Arama cümlesi, makaleler.

Çıkış: Makalelerin kimliklerini eşleştirme, önlemleri eşleştirme.

Adım 1: $S \leftarrow$ Arama cümlesini okuyunuz.

$A \leftarrow$ Bilgi grafiğindeki tüm makaleleri okuyunuz.

Adım 2: $M \leftarrow [Len(A), 2]$ // Makalelerin kimlikleri ve eşleşen kelimeler için boş bir dizi oluşturun.

Adım 3: A kümesi için her a makalesi için döngü:

aid ← a'nın ID'si

at ← a'nın başlığı

ak ← a'ya ait anahtar kelimeler

aa ← a'nın özeti

tm ← [len(t),len(S)] // Eşleşmeleri arama ifadesiyle başlık arasında tutmak

x ← 0

A kümesindeki her kelime için döngü:

y ← 0

S kümesindeki her s kelimesi için döngü:

tm[x,y] ← match(w,s) //Kelimeler arasındaki benzerliğin hesaplanması

y+=1

x+=1

ts ← sum(amax(tm, axis=0))/len(S) //Ortalama eşleşmenin hesaplanması.

km ← [len(k),len(S)] // Eşleşmeleri arama ifadesiyle anahtar kelimeler arasında saklamak için boş dizi tanımlama.

x ← 0

For each word w in ak:

y ← 0

For each word s in S:

km[x,y] ← match(w,s)

y+=1

x+=1

ks ← sum(amax(km, axis=0))/len(S)

am ← [len(t),len(S)] // Eşleşmeleri arama cümlesiyle özet arasında saklamak için boş dizi tanımlama.

x ← 0

For each word w in aa:

$y \leftarrow 0$

For each word s in S :

$am[x,y] \leftarrow \text{match}(w,s)$

$y+=1; x+=1$

$as \leftarrow \text{sum}(\text{amax}(tx, \text{axis}=0))/\text{len}(S) * 0.9$ // Başlık ve anahtar kelimelerle eşleşmeleri önceliklendirmek için hata indirgeme.

$M \leftarrow [M, [\text{aid}, \text{max}(ts, ks, as)]]$ // Makalenin kimliğini ve bulunan maksimum eşleşme puanı ekleme.

Return (M) // Makalelerin kimliklerini ve bunlarla eşleşen puanları arama ifadesiyle döndür.

Semantik ve geleneksel arama yöntemleri arasındaki farkın irdelenmesi için örnek olarak “Machine Learning” arama ifadesi, Tablo 4.2’de bilgileri örnek olarak verilen makale ile arama işlemi gerçekleşir.

Tablo 4.2. Makale sınıfına ait bir örnek olarak verilen makalenin özellikleri

Öznitelikler	Açıklama
Title	Teaching Computers the Art of Thinking
Abstract	This article investigates the possible techniques that can be used to allow computers to gain the art of thinking. Such ability can allow computers to learning how to process any type of inputs without the need for any human involvement.
Keywords	Artificial Intelligence; Neural Networks; Automatization.

Buna göre arama cümlesindeki kelimeleri arama, Tablo 4.3’de gösterildiği gibi, makalenin her bir bileşeniyle aşağıdaki benzerlik ölçütü elde edilir. Tablo 4.3’de “Machine” kelimesinin makalenin bileşenlerinde bulunmadığını, “Learning” kelimesinin özetinde bulunduğunu göstermektedir. Sonuç olarak genel benzerlik ölçütü, %50’dir. Anlamsal bilgileri ilişkilendirme bu örnek makale ile makine öğrenimi %50’den daha fazla ilgilidir. Bu nedenle; bu makalelerde yer alan kelimelerin tam anlamıyla uygun aranmaması ve semantik anlamları göz önüne bulundurulmadıkça yanlış benzerlik ölçütleri elde edilebilir.

Tablo 4.3. Makale sınıf örneğine ait bileşenler ile arama ifadesi arasındaki benzerlikler

Aranan Kelime	İçerik	Eşleşen Kelime	Benzerlik (%)
Machine	Title	-	0
Machine	Abstract	-	0
Machine	Keywords	-	0
Learning	Title	-	0
Learning	Abstract	Learning	50
Learning	Keywords	-	0

Tablo 4.4’de önerilen semantik arama yöntemi kullanılarak arama cümlesi ile örnek makale arasındaki benzerlik ölçütleri gösterilmektedir.

Tablo 4.4. Aranan cümle ile örnek makale arasındaki benzerlik sonuçları

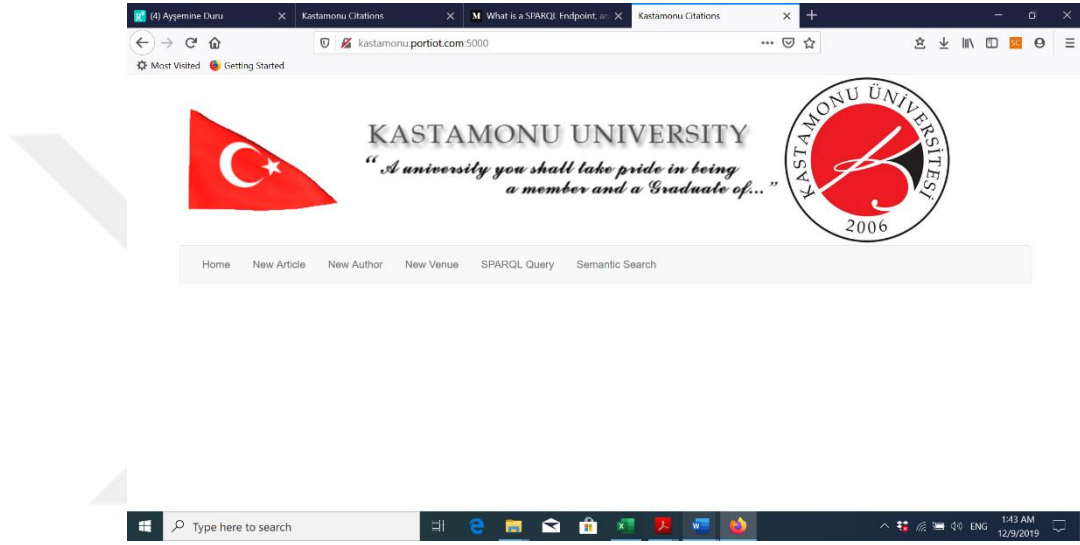
Aranan Kelime	İçerik	En İyi Eşleşen Kelime	Benzerlik (%)
Machine	Title	Computers	37.27
Machine	Abstract	Computers	37.27
Machine	Keywords	Artificial	20.41
Learning	Title	Teaching	66.18
Learning	Abstract	Learning	100
Learning	Keywords	Intelligence	10.79

Semantik arama ile “Machine” kelimesi makalenin başlığında ve soyut bileşenlerinde “Computer” kelimesiyle eşleşmiştir. Sonuç olarak “Machine” kelimesi en yüksek performans ile %37,27 benzerlik ve “Learning” kelimesiyle %100 benzerlik göstermiştir. Örnek makale üzerinde geleneksel arama yöntemini kullanarak hesaplanan %50 benzerlik, semantik arama yöntemleri ile %65,64 benzerlik değeri ile arama cümlesi eşleşmiştir.

4.4. Sistemin Arayüzleri

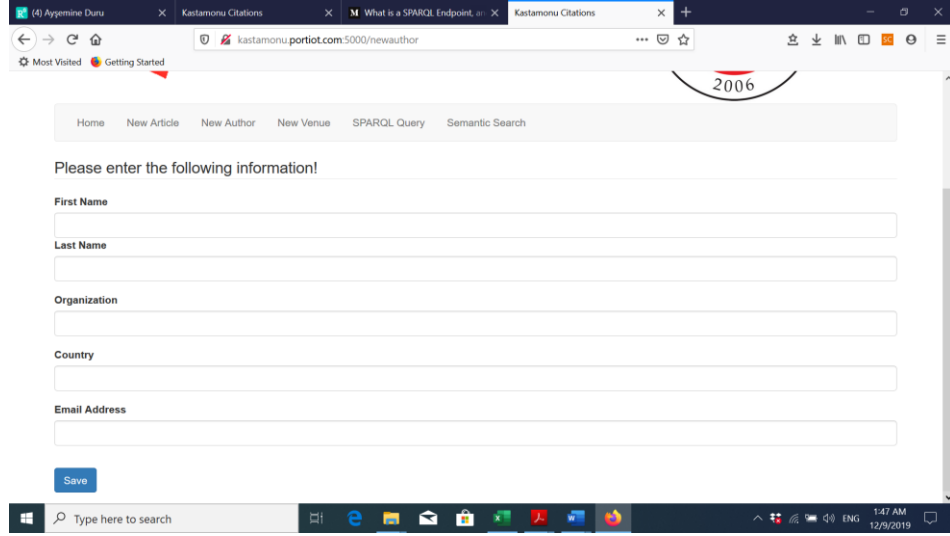
Bölüm 4.1’de atıf ontolojisi tanımlanmıştır. Atıf ontolojisine yeni veri girişi ve sorgulama işlemleri için arayüzler ve görünümeler sunulmuştur. Semantik modeller ve

ilgili verileri üzerinde gezinmek ve düzenlemek için; düğme, ağaç, grafik görselleştirme, görsel sorgu oluşturucu arayüz bileşenleri içerir. Arayüz bileşenlerini içeren sistem, yazarı hakem, okur, alan editörü, editör, editör yardımcısı, kullanıcı, dergi yöneticisi ve son kullanıcı gibi farklı profillerdeki çok sayıda kullanıcıya hizmet vermektedir. Araştırmaya konu olan yazar, makale modüllerinin özellikleri aşağıdaki ekran görüntüleri ile anlatılmaktadır. Şekil 4.5’de web sayfasının giriş ekranı görülmektedir.



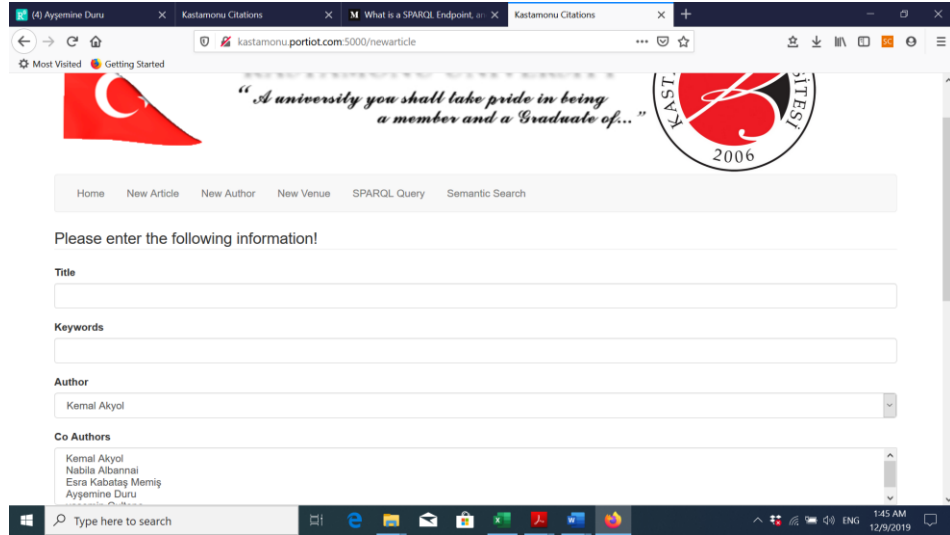
Şekil 4.5. Giriş ekranı

Bilim insanları, hazırladıkları makaleyi ilgili bir araştırma alanında uzmanlaşmış uygun bir dergiye gönderir. Herhangi bir bilimsel dergi yönetim platformu, yazarlar, hakemler ve editörler dâhil olmak üzere yayın sürecindeki farklı katılımcılar arasında çok sayıda bilgi alışverişi gerektirir. Her katılımcı, sistemdeki farklı aktörlerle işbirliği yaparak farklı bir rol oynar. Herhangi bir makalenin çevrimiçi ortamda başarılı bir şekilde yayınlanması için birkaç adımın izlenmesi gerekir. Sistemdeki yazar modülü Şekil 4.6’de gösterilmektedir.



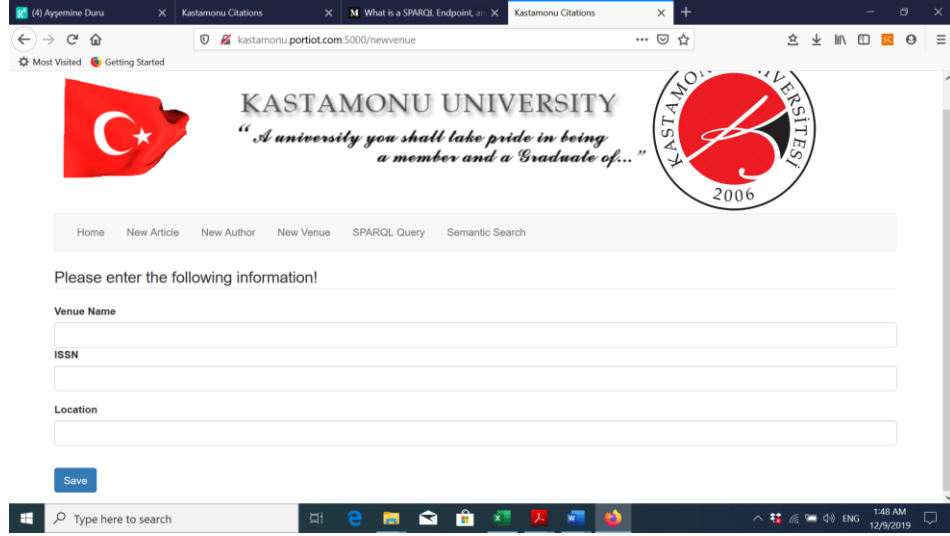
Şekil 4.6. Yazar ekleme ekranı

Sistemde herhangi bir makale göndermek için öncelikle yazar olarak sisteme kayıt olunması gerekmektedir. Yazar sisteme giriş yaptığında “New Article” bağlantısını kullanarak gelen ekrandan başlık, anahtar kelimeler, yazar, diğer yazarlar vb. bilgilerini sistem yükleyebilir. Şekil 4.7’de “New Article” veri girişi ekranı gösterilmiştir.



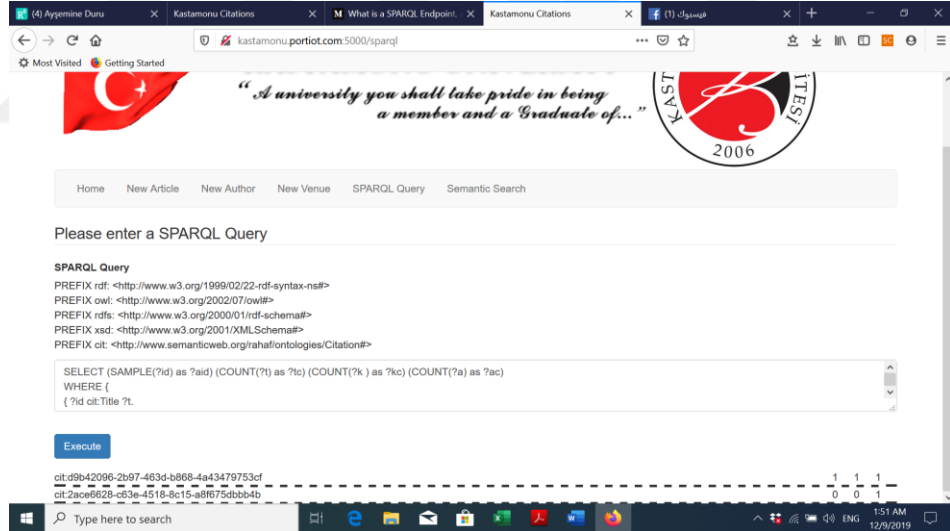
Şekil 4.7. Makale ekleme ekranı

“New Venue” bağlantısı, yayın hakkında Dergi ismi, Uluslararası Standart Süreli Yayın numarası (ISSN), yer hakkında bilgilerin girişi istenmektedir. Şekil 4.8’de yeni dergi hakkında bilgi girişi ekranı gösterilmiştir.



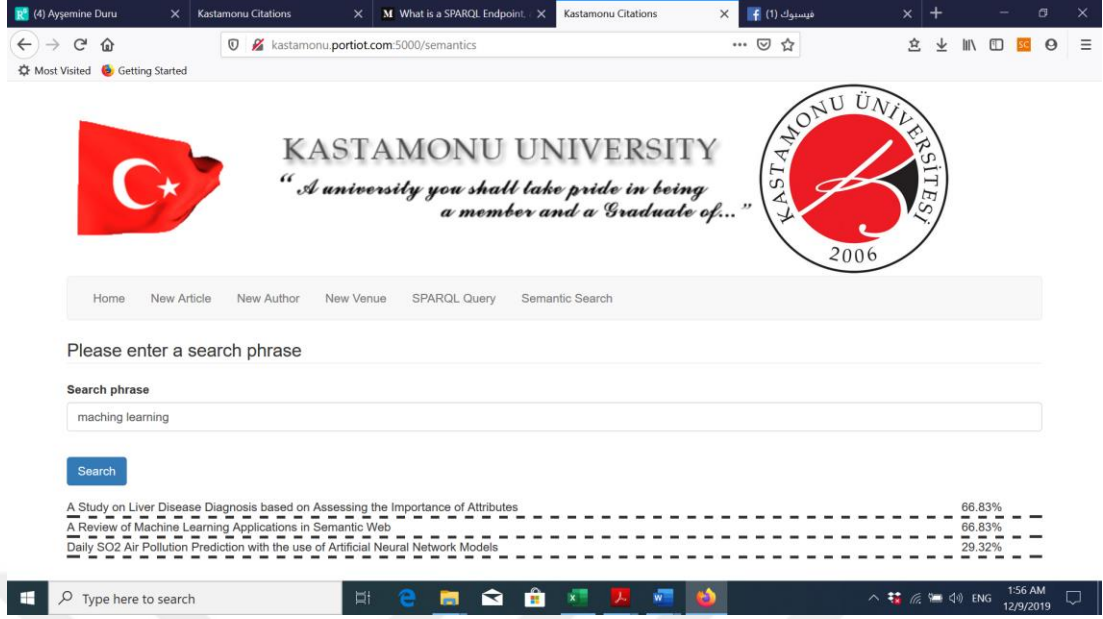
Şekil 4.8. Dergi ekleme ekranı

Sorgulama dili olarak SPARQL kullanılmıştır. Şekil 4.9’da gösterilen ekran ile SPARQL ekleme, silme, güncelleme vb. sorgular çalıştırılabilir.



Şekil 4.9. SPARQL sorgu ekranı

Şekil 4.10’de örnek bir anlamsal sorgu gerçekleştirilmiştir. “Machine Learning” kelime ile benzer olan kelimelerin geçtiği makaleler benzerlik oranları ile birlikte verilmiştir. “A Study on Liver Disease Diagnosis Based on Assessing the Importance of Attributes” makalesi ile %66.83 oranında benzerdir.



Şekil 4.10. Örnek anlamsal arama sonuçları

4.5. Araştırma Bulguları

Bu tez çalışmasında ontolojilerin geliştirilmesi, saklanması ve sorgulanması için Protégé kullanılmıştır. Protégé, bir ontoloji geliştirme editörüdür. Protégé kullanılmasının sebebi, grafiksel arayüz olarak kullanım kolaylığının olması, açık kaynak kod uygulaması olmasıdır.

Ontolojideki kaynaklar (individuals), Açık Akademik Topluluk Atıf Veri Kümesinden (Open Academic Society Citation Dataset) toplanan atıflar kullanılarak oluşturulmuştur (Open Academic Society, 2019). Bu tez çalışmasında gerçekleştirilen uygulamalar için 1.000.000 atıf bilgisini içeren “*aminer_papers_29*” isimli dosya kullanılmıştır. Tablo 4.5’de, Açık Akademik Topluluk Atıf Veri Kümesinin özellikleri verilmiştir.

Semantik arama motoru için Python programlama dili kullanılmıştır (Sanner, 1999). Python, kod okunabilirliğine odaklanan üst düzey, yorumlanmış ve genel amaçlı bir dinamik programlama dilidir. Python, grafik kullanıcı arabirimleri geliştirmek için çeşitli seçenekler sunar.

Sistem için bütün uygulamalar, 2.81 GHz frekansına ve 16 GB'lık rasgele erişim belleğine sahip bir Intel Core™ i7-7700HQ işlemcili Windows işletim sistemine sahip bilgisayar kullanılarak test edilmiştir. Gerçekleştirilen testler, önerilen arama programının sonuçlarını, doğrudan ontoloji üzerinde SPARQL sorgulama dilini kullananlarla karşılaştırmaktadır. Bu karşılaştırmalar, arama cümleleriyle ilgili bulunan benzer makale sayıları, benzerlik önlemlerinin histogramı ve aramayı yapmak ve sonuçları geri almak için gereken zamana dayalı olarak yapılmaktadır.

Tablo 4.5. Açık akademi topluluk atıf veri kümesi özellikleri

Öznitelikler	Tanımı
ID	Her makaleyi yerel olarak tanımlamak için benzersiz bir tanımlayıcı.
Title	Makalenin başlığı.
Authors	Makalenin yazarlarının listesi. Her yazar, kuruluşlarının yanı sıra ad ve soyadlarıyla da tanımlanır.
Venue	Yayınevi.
Year	Makalenin yayınlandığı yıl.
Start page	Makalenin başladığı derginin sayısındaki sayfa sayısı.
End page	Makalenin bittiği derginin sayısındaki sayfa sayısı.
Language	Makale yazım dili.
Volume	Verilen derginin cilt numarası.
ISSN	Düzenleyen yerin uluslararası standart numarası.
DOI	İnternet erişimi için makalenin dijital nesne tanımlayıcısı.
URL	Makalenin dijital kopyasına yönlendiren web adresi.
Keywords	Makalenin anahtar kelimeleri.
Abstract	Makalenin özeti.

4.5.1. SPARQL Kullanımı

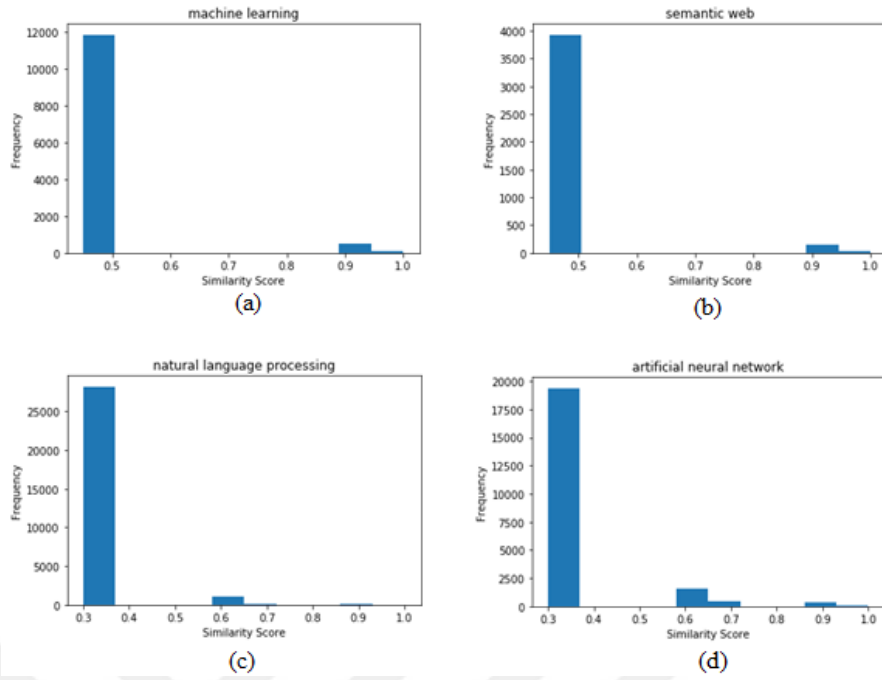
Bu uygulamada, SPARQL sorgu dili, değerlendirme için seçilen cümleleri aramak için kullanılır. Her cümle, her bir makalenin başlığında, “Keywords”de ve “Abstract”da aranır. Arama cümleleri, her bir cümlede bulunan eşleşen makale

sayısı ve ifadeyi makalenin bileşenleriyle eşleştirmek için harcanan süre, Tablo 4.6’de gösterilmektedir.

Tablo 4.6. SPARQL sorgu dilini kullanan arama sonuçları

Arama Aşaması	Eşik	Eşleşen Makale Sayısı	Yürütme Zamanı (Sn)
Machine learning	0.5	570	25.35
	0.7	570	
	0.9	83	
Semantic web	0.5	181	25.83
	0.7	181	
	0.9	41	
Natural language processing	0.5	1236	28.61
	0.7	107	
	0.9	12	
Artificial neural network	0.5	2456	29.87
	0.7	389	
	0.9	72	

Arama cümleleri ile bulunan makaleler arasındaki benzerlik puanlarının histogramı Şekil 4.11’de gösterilmiştir. Şekil 4.11’de gösterildiği şekilde arama ifadesindeki kelimelerin maksimum iki katına sahip olabilen ayrı sayıda olası benzerlik puanı bulunmaktadır. Bu olası benzerlik puanlarının toplam sayısının arama ifadesindeki kelimeleri iki katıdır. Şekil 4.9’de (a) ve (b) karşılık gelen arama ifadelerinde iki kelimenin kullanımına göre, sadece dört olası benzerlik değeri vardır. Örneğin tüm kelimeler “Title”da yer aldığına 1 değeri ve içinde sadece bir kelime bulunduğunda 0,5 değeri üretilir. Ayrıca “Abstract”ta her iki sözcüğü bulmak iskonto faktörünün kullanımına göre 0,9 benzerlik üretir ve arama ifadesindeki kelimelerden sadece biri bulunduğunda 0,45 elde edilir. Benzer şekilde, arama cümlelerinde üç kelimenin kullanımına göre histogram (c) ve (d)’de altı olası benzerlik değeri vardır, böylece, kelimelerin tümü, ikisi veya biri, 1, 0,66 ve 0,33 üretilir. Arama ifadesi “Title”da bulunur ve “Abstract” sırayla aynı sayıda kelime bulunduğunda 0,9, 0,6 ve 0,3 üretilir.



Şekil 4.11. SPARQL ile yapılan aramaların benzerlik puanları

4.5.2. Kelime Vektörlerinin Kullanımı

Bölüm 4.3’de ifade edilen algoritmaya göre arama ifadesindeki kelimeler için tam bir eşleşmenin aranması, benzerlik puanları için ayrı ayrı olası değerlere ve makalelerde sınırlı sayıda eşleşmeye neden olabilir. Bu nedenle, önerilen anlamsal arama şeması kullanılarak aynı makaleler üzerinde yeni bir uygulama gerçekleştirilir. Arama kelime grubundaki kelimeler, her bir makalenin “Title”, “Keywords” ve “Abstract” ile eşleştirilir; burada özetlerdeki kelimelerle gerçekleşen eşleşmeler, “Title” ve “Keywords”deki eşleşmelere daha fazla vurgu yapmak için azaltılır. Tablo 4.7’de arama cümleleri, her ifadede bulunan eşleşen makale sayısı ve ifadeyi bu denemenin makaleleriyle eşleştirmek için harcanan süre gösterilmiştir.

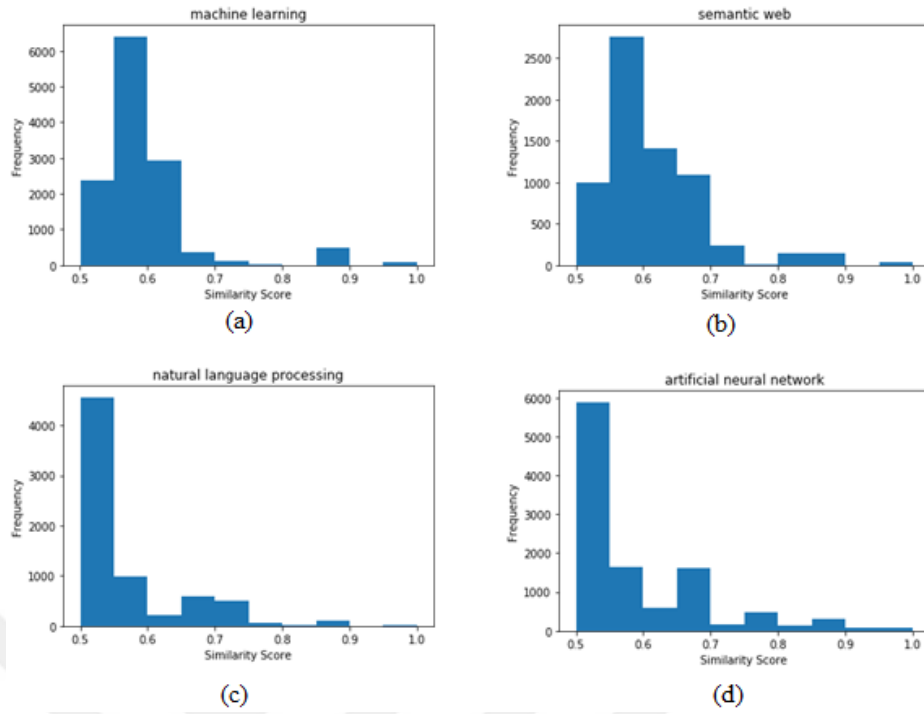
Tablo 4.7. Önerilen arama şemasını kullanarak arama sonuçları

Arama Aşaması	Eşik	Eşleşen Makale Sayısı	Yürütme Zamanı (Sn)
Machine learning	0.5	12788	72.81
	0.7	712	
	0.9	83	

Tablo 4.7. *devamı*

Arama Aşaması	Eşik	Eşleşen Makale Sayısı	Yürütme Zamanı (Sn)
Semantic web	0.5	6832	73.55
	0.7	566	
	0.9	41	
Natural language processing	0.5	7024	93.90
	0.7	667	
	0.9	12	
Artificial neural network	0.5	10942	91.68
	0.7	1200	
	0.9	139	

Şekil 4.12’de arama cümleleri ile 0,5’den büyük benzerliğe sahip bulunan makaleler arasındaki benzerlik puanları gösterilmiştir. Önceki uygulama sonuçlarının aksine, önerilen yöntemdeki benzerlik puanları sürekli değerler olabilir. Bu değerler, arama ifadelerindeki kelimeler ile makalelerin bileşenlerindeki sözcükler arasındaki doğrusal benzerlik puanlarının sonuçlarıdır. Kelime gömme yöntemi ile çıkarılan vektörlerdeki özellik değerlerine dayanarak elde edilmiştir. Bu nedenle, önerilen yöntem benzerlik puanlarının daha iyi bir dağılımına sahip olması ontoloji tabanlı uygulamaya göre daha benzer makaleler arasında eşleştirme yapılmıştır. Bununla birlikte, histogramları (c) ve (d)’i (a) ve (b) ile karşılaştırarak, arama ifadesindeki sözcük sayısını arttırmak, bu makalelerde eşleşen kelimeyi bulma olasılığının yüksek olmasına göre, düşük benzerlik önlemlerine sahip eşleşen makalelerin sayısını artırır.



Şekil 4.12. Önerilen anlamsal arama şeması ile yapılan aramaların benzerlik puanları

Tablo 4.8’de seçilen arama ifadelerindeki her bir kelimeye en çok benzeyen kelimelerin örneklerini ve bunlara uyan ölçülerini göstermektedir.

Tablo 4.8. Seçilen arama cümleleri içindeki kelimelerin anlamsal olarak benzer kelimeler arasındaki benzerlikler

Aranan Kelime	Benzer Kelime	Benzerlik Ölçümü (%)
Machine	Machines	64.47
	Tool	49.14
	factory	45.89
	Fully-automatic	42.76
	Mulcher	41.90
Learning	Teaching	64.54
	Educational	57.16
	Learners	54.78
	Training	54.05
	Knowledge	51.37
Semantic	Semantics	66.45
	Contextual	59.09

Tablo 4.8. *devamı*

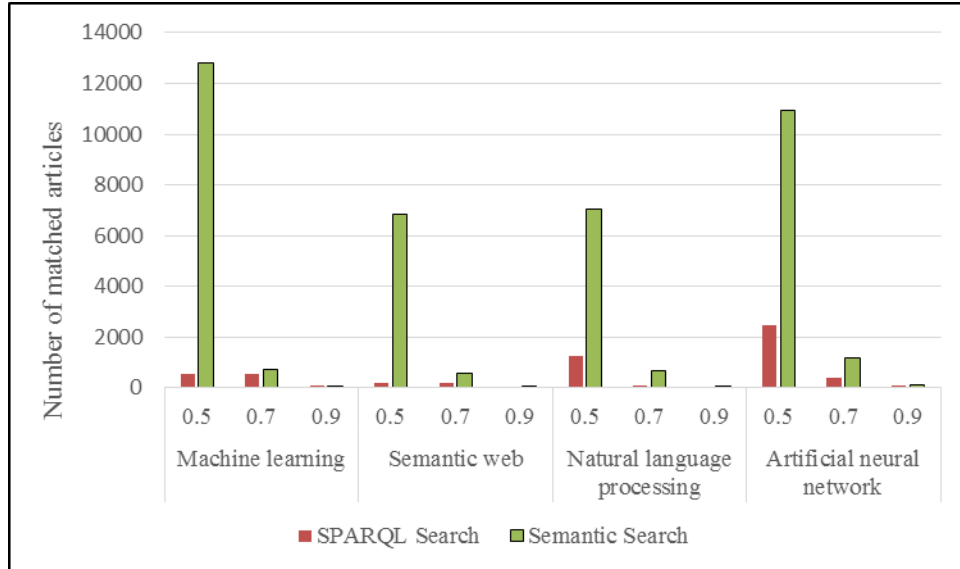
Aranan Kelime	Benzer Kelime	Benzerlik Ölçümü (%)
	Meta	59.06
	Ontology	58.81
	Semantically	56.00
Web	Website	69.71
	Webpage	63.68
	Online	63.19
	Internet	57.87
	Site	53.55
Natural	Manmade	52.76
	Naturally	49.99
	Human	44.83
	Abundant	43.53
	Instinctive	42.59
Language	English	74.77
	Translation	54.41
	Vocabulary	54.36
	Linguistic	54.28
	Idioms	52.96
Artificial	Synthetic	48.18
	Artificially	43.79
	Unnatural	43.12
	Illusion	43.10
	Manmade	41.43
Neural	Neuronal	78.05
	Neurons	73.27
	Synaptic	66.99
	Brain	65.06
	Cortex	64.35
Network	Networks	81.81
	Fiber	53.20
	IPVPN	53.10
	Internet	52.44

Tablo 4.8. *devamı*

Aranan Kelime	Benzer Kelime	Benzerlik Ölçümü (%)
	WiFi	52.02

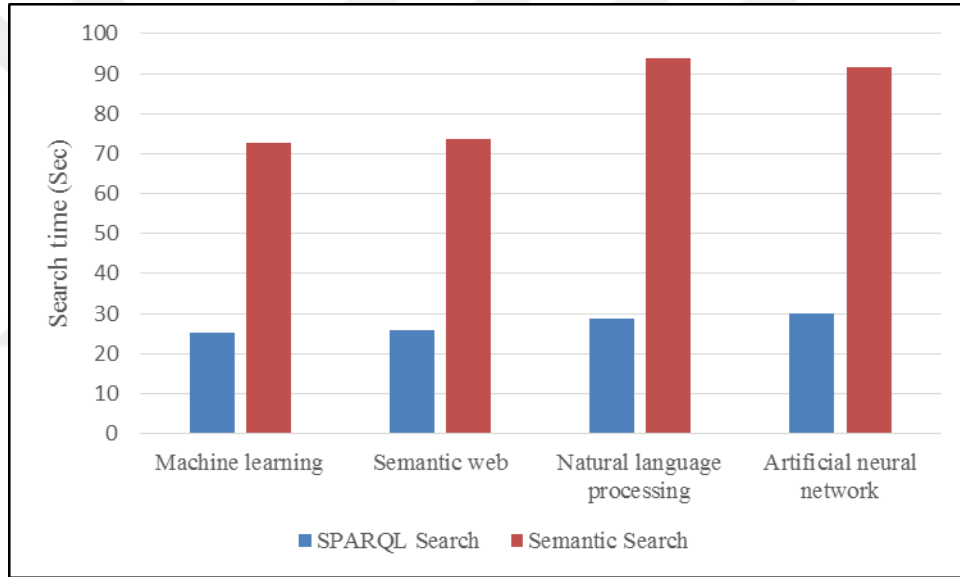
4.5.3. Değerlendirme

Her iki uygulama sonucunda ve Şekil 4.13’de gösterilen eşleşen belgelerin sayısı, bu tez çalışmasının önerilen derin öğrenmeye dayalı semantik arama yönteminin çok daha fazla sayıda eşleşen belge bulabildiğini göstermektedir. SPARQL sorgu dilinin arama sonuçlarına göre hesaplanan daha düşük bir benzerlik puanı, makalelerde daha az sayıda benzer kelimenin bulunduğunu gösterirken, önerilen semantik yöntemdeki bu daha düşük puan mutlaka daha düşük sayıda eşleşen kelimeyi göstermez. Anlamsal arama şemasının, arama kelime cümlesindeki tüm kelimeler için eşleşmeler bulması, ancak eşleşen kelimeler orijinal arama cümlesindeki kelimelere oldukça benzer olduğu için hala düşük bir genel benzerlik puanı elde etmesi mümkündür. Böylece, önerilen yöntem, SPARQL sorgularının basit kullanımıyla karşılaştırıldığında ontolojideki makaleler için arama sonuçlarını önemli ölçüde iyileştirebilmiştir.



Şekil 4.13. Gerçekleştirilen deneylerde kullanılan arama cümleleriyle eşleşen makale sayıları

Ek olarak, arama cümlesi ile makale arasındaki benzerliği ölçmek için her iki yöntem için gereken süre, Şekil 4.14’de özetlenmiştir. Karşılaştırma, önerilen arama düzeninin eşleşen makaleleri bulmak ve arama sorgusunun sonuçlarını döndürmek için daha fazla zaman harcadığını göstermektedir. Arama süresindeki bu artış, arama prosedürünü iki aşamaya bölmenin bir sonucudur, birinci aşama derin öğrenme modelini kullanarak eşleşen makalelerin kimliklerini çıkarırken, ikinci aşama eşyalar ile ilgili bilgilerin kimliklerini ontolojiden almaktadır. Ayrıca, kelimeler arasındaki benzerliği ölçmek, aynı olup olmadıklarını bulmak için bu kelimeleri birbiriyle karşılaştırmaktan daha karmaşıktır. Bu nedenle, benzerlik puanı ölçüm aşamasında daha fazla yürütme süresi harcanmaktadır.



Şekil 4.14. Eşleşen makalelerin arama zamanlarının karşılaştırılması

Programın geliştirilmesinde Python programlama dili ve Flask kütüphanesi (Ronacher, 2018) kullanılmıştır. RDFLib, bilgileri temsil etmek için basit ama güçlü bir dil olan RDF ile çalışmak için bir Python kütüphanesidir (Krech, 2006). Uygulanan web uygulaması beş ana bölümden oluşmaktadır:

- Yeni Yazar: Her yazarın bilgilerini ontolojiye eklemek için kullanılır. Her yazara, ontolojideki diğer nesnelere, özellikle de Yazılarla olan ilişkileri tanımlamak için otomatik olarak benzersiz bir kimlik atanır.

- Yeni Dergi: Otolojiye yeni bir dergi eklenmesine izin verir. Her bir mekâna, ontolojinin tanımlanmış ilişkilerinde kullanılmak üzere benzersiz bir kimlik numarası verilir.
- Yeni Makale: Bir makalenin ilişkilerini tanımlamak için yazarların ve mekânların mevcut bilgilerini kullanan yeni makaleler eklemek için kullanılır. Her yazar, makalenin ana yazarı veya ortak yazar olabilir. Her makalenin yalnızca bir yazarı vardır, ancak çok sayıda ortak yazarı olabilir.
- SPARQL Sorgu: SPARQL sorgularını kullanarak ontolojiyi sorgulamaya izin verir. Önceden tanımlanmış bir dizi ön ek kullanarak, sorgu ontolojiyi okuyan ve gerekli sonuçları alan RDFLib'e iletir.
- Semantik Arama: Önerilen semantik arama yöntemini kullanarak bir arama ifadesi girişi yapılarak makaleleri aramanıza izin verir. Soyutlanan ve daha sonra önerilen yöntem kullanılarak anlamsal arama işlemi gerçekleştirilir. Ardından, makalelerin başlıkları, girilen arama ifadesiyle olan benzerlik önlemleriyle birlikte geri gönderilir.

5. SONUÇLAR

Dijital çağın katlanarak büyümesiyle, daha kolay ve daha güvenilir bir erişim için daha fazla hizmet dijitalleştiriliyor. Dijital kütüphaneler, her gün yayınlanan makalelerin sayısını önemli ölçüde artıran bilimsel makaleleri endekslemek ve yayınlamak için yaygın olarak kullanılmaktadır. Bu büyüme, bu maddelere esnek erişim sağlamak için mevcut dizin oluşturma depolama, dizin oluşturma ve yönetim yöntemlerinin yükseltilmesini gerektirir. Bilgisayarlar arasında nesnelere hakkında bilgi depolamak ve değiştirmek için yaygın olarak kullanılan yöntemlerden biri de bilgi graflarının kullanılmasıdır. Bu grafları, yapılandırılmış verilerde olduğu gibi sabit sayıda özellik kullanmak yerine, her bir nesneyi tanımlamak için farklı nesnelere arasındaki bağlantıları kullanır.

Makaleler, insanlar tarafından, doğal dilde yazılmış cümlelerin anlamsal anlamlarının karmaşıklığına göre, bilgisayarların anlayabileceği bir formatta sunulması zor olan doğal dilleri kullanılarak yazılmaktadır. Derin öğrenme, farklı uygulamalardaki üstün performanslarına göre, doğal dil işleme için yaygın olarak kullanılan makine öğrenme tekniklerinden biridir. Bu uygulamalardan biri, iki kelime arasındaki anlamsal benzerliği ölçmektir. Böylece ikili bir karar yerine doğrusal bir benzerlik ölçütü sağlanır. Bu nedenle, bu yaklaşımı kullanarak eşleşmeleri aramak, özellikle doğal dil kullanılarak yazılmış makaleleri işlerken, arama sonuçlarını önemli ölçüde iyileştirebilir.

Bu çalışmada, makalelerin, yazarların ve yayıncıların bilgilerini saklamak için bir atıf ontolojisi uygulanmıştır. Bununla birlikte, SPARQL dilini kullanarak bir arama ifadesiyle eşleşen belgeleri aramak, kelimelerin anlamsal anlamını göz önünde bulunduramaz; böylece, ifadedeki her kelime için yalnızca ikili kararlar verilebilir. Bu nedenle, sözcükleri farklı kelimeler arasındaki benzerliği ölçmek için kullanılacak vektörlere dönüştürmek için derin öğrenmeye dayanan yerleştirme tekniğini kullanmak için bir arama şeması önerilmiştir. Ayrıca, arama cümlesi ile makalenin özeti arasındaki benzerlik puanı, başlık ve anahtar kelimelerin puanlarını vurgulamak için %90'a düşürülür. Ontolojide eşleşen makaleleri aramak için gereken

zamandaki artışa rağmen, önerilen arama şeması önemli ölçüde daha iyi sonuçlar göstermiştir. Arama sonuçlarındaki iyileşme, özellikle daha düşük benzerlik önlemlerinde daha fazla sayıda eşleştirme makalesi ve puanların daha iyi dağılımı ile gösterilmektedir. SPARQL'in benzerlik puanları, arama cümlesindeki kelime sayısına bağlı olarak ayrı değerlere sahipken, önerilen yöntemden elde edilenlerin sürekli değerleri vardır, böylece daha uygun bir düzen üretilebilir.

Gelecekteki çalışmalarda, makalelerin bileşenlerinde eşleşen kelimelerin arama cümlesiyle karşılaştırılması, genel benzerlik puanının hesaplanmasında dikkate alınacaktır. Kelimelerin cümle içindeki konumlandırılması bazen anlamsal anlam üzerinde önemli bir etkiye sahip olabileceğinden, bu düşüncenin arama sonuçlarını iyileştirmesi mümkündür. Ek olarak, benzerlik ölçüm prosedürü paralelleştirilecektir, böylece birden fazla kelime tek tek yerine arama ifadesindeki tek bir kelimeyle karşılaştırılabilir. Bu tür bir paralelleştirme, arama yapmak için gereken süreyi önemli ölçüde azaltabilir, çünkü tek bir kelime ile benzerliği ölçmek için gereken aynı zaman, bilgisayarın özelliklerine bağlı olarak binlerce kelimeyi karşılaştırmak için kullanılabilir.

KAYNAKLAR

- Agarap, A. F. (2018). Deep learning using rectified linear units (ReLU), CC BY-NC-SA 4.0.
- Albawi, S., Mohammed, T. A., & Alzawi, (2017). Understanding of a convolutional neural network, *The International Conference on Engineering and Technology*.
- Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application, *Journal of Microbiological Methods*, 43(1), 3-31.
- Bengio, Y. (2009). Learning deep architectures for AI, *Foundations and Trends® in Machine Learning*, 2(1).
- Berners-Lee, T. Hendler, J., & Lassila, O. (2001). *The semantic web*, Scientific American 284(5), 34-43.
- Bernstein, A., Hendler, J., & Noy, N. (2016). A new look at the semantic web, *Communications of the ACM*, 59(9), 35-37.
- Bloem P., & Vries, G. K. (2014). Machine learning on linked data, a position paper, *Linked Data for Knowledge Discovery*, 15-19.
- Cambria, E. (2016). Affective computing and sentiment analysis, *IEEE Intelligent Systems*, 31(2), 102-107.
- Cho, K., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv:1406.1078.
- Chong, E. I., Das, S., Eadon, G., & Srinivasan, J. (2005). An efficient SQL-based RDF querying scheme, *Proceedings of the 31st international conference on Very large data bases*, VLDB Endowment, 1216-1227.
- Cortes, C., Gonzalvo, X., Kuznetsov, V., Mohri, M., & Yang, S. (2016). Adanet: Adaptive structural learning of artificial neural networks, *arXiv preprint arXiv:1607.01097*.
- Deng, L., & Yu, D. (2014). Deep learning: methods and applications, *Foundations and Trends® in Signal Processing*, 7(3-4), 197-387.
- Dettmers, T. (2015). Deep learning in a Nutshell: histroy and training paralel forall, <https://devblogs.nvidia.com/deep-learning-nutshell-history-training/>, Erişim tarihi: 01/11/2019.

- de Vries, G. K. (2013). A fast approximation of the Weisfeiler-Lehman graph kernel for RDF data, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 606-621.
- DuCharme, B. (2013). *Learning SPARQL: querying and updating with SPARQL 1.1*. O'Reilly Media, Inc.
- Esfe, M. H., Ahangar, M. R. H., Rejvani, M., Toghraie, D., & Hajmohammad, M. H. (2016). Designing an artificial neural network to predict dynamic viscosity of aqueous nanofluid of TiO₂ using experimental data, *International Communications in Heat and Mass Transfer*, 75, 192-196.
- Faruqui, M., Tsvetkov, Y., Rastogi, P., & Dyer, C. (2016). Problems with evaluation of word embeddings using word similarity tasks, arXiv preprint arXiv: 1605.02276.
- Garcia, A., Gomez, J. M., Colomo, R., & Garcia, F. (2011). Digital libraries and Web 3.0. the CallimachusDL approach, *Computer in Human Behavior*, 27(4), 1424-1430.
- Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing, *Science*, 349(6245), 261-266.
- Ivakhnenko, A. G., & Lapa, V. G. (1965). *Cybernetic Predicting Devices*, New York, CCM Information Corp.,
- Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial, *Computer*, 29(3), 31-44.
- Khoo, C. S. G., & Na, J. C. (2006). Semantic relations in information science, *Annual Review of Information Science and Technology*, 40, 157-229.
- Koivunen, M. R., & Miller, E. (2001). W3C semantic web activity, *Semantic Web Kick-Off in Finland*, 2, 27-44.
- Koutsomitropoulos, D. A., Solomo G. D., & Papatheodorou, T. S. (2013). Semantic query answering in digital repositories: Semantic search v2 for Dspace, *International Journal of Metadata Semantics & Ontologies*, 8(1), 46-55.
- Krech, D. (2006). RDFLib: A Python library for working with RDF, <https://pypi.org/project/rdflib/>, Erişim Tarihi: 12.12.2019.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning, *Nature*, 521(7553), 436-444.
- Lewandowski, D., & Mayr, P. (2006). Exploring the academic invisible web, *Library Hi Tech*, 24(4), 529-539.

- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation, *IEEE conference on computer vision and pattern recognition*, 2015, 3431-3440.
- Lossau, N. (2004). Search engine technology and digital libraries: Libraries need to discover the academic internet, *D-Lib Magazine*, 10(6).
- Lösch, U., Bloehdorn, S., & Rettinger, A. (2012). Graph kernels for RDF data, *Extended Semantic Web Conference*, 134-148.
- Majumder, N., Poria, S., Gelbukh, A., & Cambria, E. (2017). Deep learning-based document modeling for personality detection from text, *IEEE Intelligent Systems*, 32(2), 74-79.
- McCrae J. et al. (2012). Interchanging lexical resources on the semantic web, *Language Resources and Evaluation*, 46(4), 701-719.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013a). Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems*, 3111-3119.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). Efficient estimation of word representations in vector space. arXiv preprint arXiv: 1301.3781.
- Mikolov, T., Yih, W.-T., & Zweig, G. (2013c). Linguistic regularities in continuous space word representations, *Proceeding of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746-751.
- Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning, *Proceedings of the 28th international conference on machine learning*, 689-696.
- Nickel, M., Tresp, V., & Kriegel, H.-P. (2011). A Three-Way Model for Collective Learning on Multi-Relational Data, *ICML*, 11, 809-816.
- Noy, N. F., & McGuinness, D.L. (2001). *Ontology Development 101: A Guide To Creating Your First Ontology*, *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05*.
- Open Academic Society (2019). Open Academic Society, <https://www.openacademic.ai/>, Erişim tarihi: 11/11/2019.
- Pascal, H., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., & Rudolph, S. (2012). *OWL 2 Web Ontology Language Primer*, <https://www.w3.org/TR/owl2-primer/>, Erişim tarihi: 29/08/2016.
- Patkar, V. Z. (2011). Passage to ontology tool for information organization in the digital age, *Journal of Library & Information Technology*, 31(2).

- Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network, *Knowledge-Based Systems*, 108, 42-49.
- Rasmus, A., Berglund, M., Honkala, M., Valpola, H., & Raiko, T. (2015). Semi-supervised learning with ladder networks, *Advances in Neural Information Processing Systems*.
- Rende, F. Ş., Bütün, G., & Karahan, Ş. (2016). Derin öğrenme algoritmalarında model testleri: Derin testler, *10. Ulusal Yazılım Mühendisliği Sempozyumu*, 24-26 Ekim, 54-59.
- Ronacher, A. (2018). Flask (A Python Microframework), <http://flask.pocoo.org>, Erişim Tarihi: 12/12/2019.
- Sadeh, T. (2007). Time for a change: New approaches for a new generation of library users, *Library World*, 108(7/8), 307-316.
- Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks, 2015 IEEE International Conference on *Acoustics, Speech and Signal Processing*, 4580-4584.
- Sanner, M. F. (1999). Python: A programming language for software integration and development, *J Mol Graph Model*, 17(1), 57-61.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview, *Neural networks*, 61, 85-117.
- Shotton, D. Portwin, K., Klyne, G., & Miles, A. (2009). Adventures in semantic publishing: Exemplar semantic enhancements of a research article, *PLoS computational biology*, 5(4), p. e1000361.
- Shotton, D. (2009). Semantic publishing: The coming revolution in scientific journal publishing, *Learned Publishing*, 22(2), 85-94.
- Solodovnik, I. (2011). Metadata issues in digital libraries: Key concepts and perspectives, CC BY 4.0.
- Stanford Medical Informatics (2006). Protégé, <https://protege.stanford.edu/>, Erişim tarihi: 01/02/2019.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision, *IEEE conference on computer vision and pattern recognition*.
- Tang, D., Qin, B., & Liu, T. (2015). Deep learning for sentiment analysis: successful approaches and future challenges, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(6), 292-303.

- Xiao, M., Shi, Z., & Wang, S. (2018). *The impact on citation analysis based on ontology and linked data*, Scientometrics books.
- Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification, *Advances in neural information processing systems*.
- Zhao, H., Lu, Z., & Poupart, P. (2015). Self-Adaptive Hierarchical Sentence Model, *IJCAI*.
- Zhou, T., Sedoc, J., & Rodu, J. (2019). Getting in Shape: Word Embedding Subspaces, the Twenty-Eighth Interbational Joint Conference on Artificial Intelligence, 5478-5484.
- Waard, A. (2010). From proteins to fairytales: Directions in semantic publishing, *Intelligent systems, IEEE*, 25(2), 83-88.
- Wang, H. (2015). Semantic Deep Learning, *University of Oregon*, 1-42.
- Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C. C. J. (2019). Evaluating Word Embedding Models: Methods and Experimental Results, arXiv: 1901.09785v2.
- Wang, Y., Mao, H., & Yi, Z. (2017). Protein secondary structure prediction by using deep learning metod, *Knowledge-Based Systems*, 118, 115-123.
- Wei, Y., et al. (2016). Hcp: A flexible cnn framework for multi-label image classification, *IEEE transactions on pattern analysis and machine intelligence*, 38(9), 1901-1907.
- Wohlgenannt, G., & Minic, F. (2016). Using word2vec to build a simple ontology learning system, *International Semantic Web Conference*.
- Wood, D., Zaidman, M., Ruth, L., & Hausenblas, M. (2014). *Linked Data Structured Data on the Web*, Manning Publications.
- W3C (2014), RDF 1.1 concepts and abstract syntax, <https://www.w3.org/TR/rdf11-concepts/>, Erişim Tarihi: 27.08.2019.

ÖZGEÇMİŞ

Adı Soyadı : Nabila Elmukhtar Mohamad ALBANNAI
Doğum Yeri ve Yılı : Tripoli, LİBYA 1976
Medeni Hali : Evli
Yabancı Dili : Arapça, İngilizce, Türkçe
E-posta : rehamranemrahf2009@gmail.com



Eğitim Durumu

Lise : Libya Lisesi, 1994
Lisans : Gharyan Kapsamlı Meslekler Yüksek Enstitüsü, Libya, 1998

Mesleki Deneyim

İş Yeri : Gharyan Kapsamlı Meslekler Yüksek Enstitüsü, Libya,
Eğitmen, 2002- 2012

Yayımları

Albannai, N. Gültepe, Y., & Najih, A. (2019). A Review of Machine Learning Applications in Semantic Web, *International Journal of Scientific and Technological Research*, 5(8), 1-10.