KTO KARATAY ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK ANABİLİM DALI
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ TEZLİ
YÜKSEK LİSANS PROGRAMI

TABLO VERİLERİNİN BİLGİ TABANLARINA EŞLEŞMESİ İÇİN YÖNTEM GELİŞTİRİLMESİ

Arife KARAAĞAÇ

Yüksek Lisans Tezi

KONYA
Eylül 2021

TABLO VERİLERİNİN BİLGİ TABANLARINA EŞLEŞMESİ İÇİN YÖNTEM GELİŞTİRİLMESİ

Arife KARAAĞAÇ

KTO Karatay Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik Elektronik Anabilim Dalı
Elektrik ve Bilgisayar Mühendisliği
Tezli Yüksek Lisans Programı

Yüksek Lisans Tezi

Tez Danışmanı: Dr. Öğr. Üyesi Semih YUMUŞAK

Konya
Eylül 2021

BİLDİRİM

Enstitü tarafından onaylanan Yüksek Lisans/Doktora tezimin tamamını veya herhangi bir kısmını basılı veya dijital biçimde arşivleme ve aşağıda belirtilen koşullar dahilinde erişime açma iznini KTO Karatay Üniversitesine verdiğimi bildiririm. Bu izinle, Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak ve gelecekteki çalışmalar (makale, kitap, lisans, patent vb.) için tezimin tamamının veya bir bölümünün kullanım hakları yalnızca bana ait olacaktır.

Tezimin bütünüyle kendi çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Telif hakkı bulunan ve sahiplerinden yazılı izinle kullanılması zorunlu olan kaynakları, yazılı izin alarak kullandığımı ve istenildiğinde izinlerin suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayımlanan "Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge" kapsamında, tezim, aşağıda belirtilen koşullar haricince, YÖK Ulusal Tez Merkezi ve KTO Karatay Üniversitesi Açık Erişim Sisteminde erişime açılır.

☐ Enstitü / Fakülte Yönetim Kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir.[1]

☐ Enstitü / Fakülte Yönetim Kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren … ay ertelenmiştir.[2]

☐ Tezimle ilgili gizlilik kararı verilmiştir.[3][4]

06 Eylül 2021

_____

Arife KARAAĞAÇ

[1] MADDE 6(1) Lisansüstü tezle ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

[2] MADDE 6(2) Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.

[3] MADDE 7(1) Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

[4] MADDE 7(2) Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

ETİK BEYAN

KTO Karatay Üniversitesi Lisansüstü Eğitim Enstitüsü Tez/Proje Hazırlama ve Yazım Kurallarına uygun olarak Dr. Öğr. Üyesi Semih YUMUŞAK danışmanlığında tarafımdan üretilen bu tez/proje çalışmasında; sunduğum tüm veri, enformasyon, bilgi ve belgeleri bilimsel etik kuralları çerçevesinde elde ettiğimi, tüm değerlendirme, analiz, bulgu ve sonuçları bilimsel usullere uygun olarak sunduğumu, tez/proje çalışmasında yararlandığım kaynakların tümüne bilimsel normlara uygun biçimde atıfta bulunarak kaynak gösterdiğimi, tezimin/projemin kaynak gösterilen durumlar dışında özgün olduğunu bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

06 Eylül 2021

_____

Arife KARAAĞAÇ

TEŞEKKÜR

Çalışmalarım boyunca tecrübeleri ile desteğini esirgemeyen tez danışmanım Dr. Öğr. Üyesi Semih YUMUŞAK' a, yapmış olduğum yüksek lisans eğitimim boyunca yardımını eksik etmeyen Dr. Öğr. Üyesi H.Oktay ALTUN'a, her zaman yanımda olan arkadaşlarıma, maddi ve manevi destekleri ile beni bir an olsun yalnız bırakmayan canım aileme; anneme, babama, ablama ve eşime sonsuz teşekkürlerimi sunarım.

06 Eylül 2021

Arife KARAAĞAÇ

ÖZET

Arife KARAAĞAÇ

Tablo Verilerinin Bilgi Tabanlarına Eşleşmesi İçin Yöntem Geliştirilmesi

Yüksek Lisans Tezi

Konya, 2021

Anlamsal ağ, belgelerin meta bilgilerle açıklandığı, yorumlanabilir ve kullanılabilir bir biçimde ifade edildiği bir internet teknolojisidir. Meta bilgi, belgenin içeriğini işlenebilir bir şekilde tanımlar, böylece farklı kaynaklardan gelen veriler ilişkilendirilebilir ve sorgulanabilir olmaktadır. Anlamsal ağlarda, varlıklar benzersiz kaynak tanımlayıcıları (URI) ile belirtilir, bu tanımlayıcılar varlıkların uygulamalar arasında benzersiz bir şekilde başvurulmasına izin verir ve yerel aramanın getirdiği kısıtlamaların üstesinden gelir. Bu tez çalışmasında, tablo biçiminde varolan bir veri kümesi (SemTab 2020) ile anlamsal ağ bilgi tabanlarının benzersiz kaynak tanımlayıcılar üzerinden eşleştirilmesi çalışmaları incelenmiştir. Kullanılan veri kümesinin Wikidata bilgi tabanı ile eşleştirilme çalışmaları incelenmiş ve çalıştırmıştır. Sonrasında, anlamsal olarak birbirlerine bağlantılı şekilde tablolar oluşturulmuştur. İlgili veri kümesi üzerinde bulunan yazım hataları detaylı bir şekilde incelenmiş ve raporlanmıştır. Bu bağlamda, tez çalışmasında incelenen ve açıklanan hata kategorilerine çözüm önerileri sunulmuştur.

Anahtar Kelimeler

Anlamsal ağ, bağlantılı veri, bilgi tabanları, tablo eşleştirme

ABSTRACT

Arife KARAAĞAÇ

Developing A Method To Matching The Table Data With The Knowledge Bases

Master's Thesis

Konya, 2021

The semantic web is an internet technology in which documents are explained with meta information and expressed in an interpretable and usable form. Meta information describes the content of the document in an actionable way so that data from different sources is relatable and questionable. In semantic networks, entities are identified by unique resource identifiers (URIs), which allow entities to be referenced uniquely across applications and overcome the limitations of local search. In this thesis, the studies of matching an existing tabular dataset (SemTab 2020) and semantic network knowledge bases over unique resource identifiers were examined. The matching studies of the dataset used with the Wikidata knowledge base were examined and run. Afterwards, tables were created semantically linked to each other. Spelling errors on the relevant data set were examined in detail and reported. In this context, solution suggestions were presented for the error categories examined and explained in the thesis study.

Keywords

Semantic web, linked data, knowledge bases, table matching

İÇİNDEKİLER

TABLOLAR DİZİNİ

ŞEKİLLER DİZİNİ

KISALTMALAR DİZİNİ

Kısaltma	Açıklama
RDF	Resource Description Framework
RDFS	RDF Schema
WWW	World Wide Web
W3C	World Wide Web Consortium
HTTP	Hyper-Text Transfer Protocol
HTML	Hyper-Text Markup Language
URI	Uniform Resource Identifier
CTA	Column Type Annotation
CEA	Cell Entity Annotation
CPA	Column Property Annotation

1. GİRİŞ

Evrensel bir bilgi platformu olan World Wide Web, İsviçre'nin Cenevre kentinde bulunan uluslararası bir bilimsel organizasyon olan CERN'de (Avrupa Nükleer Araştırma Merkezi) Tim Berners-Lee tarafından gelişimi başlatılmış ve ilerleyen zamanlarda belgeleri yayınlama ve bunlara erişmenin önündeki engelleri azaltarak bilgi paylaşımı için farklı yöntemler geliştirilmiştir (Berners-Lee, 1992). Anlamsal ağ, web sayfalarındaki verileri tanımlama ve birbirleriyle anlamsal ilişkiler kurma fikrini gerçekleştirmeyi amaçlayan W3C tarafından başlatılan uzun vadeli bir projenin adıdır. Bağlantılı veri teknolojisine sahip bu uygulama, WWW mucidi Tim Berners-Lee tarafından tasarlanmıştır (Taye, 2010). Anlamsal ağ, WWW tarayıcısını kullanarak, yalnızca insanların erişebileceği ve okuyabileceği web içeriğinin aynı zamanda makineler tarafından da işlenebileceği fikrine dayanmaktadır.

Kullanıcılar, Web' de veri bulmak için, anahtar kelime kullanan Web arama motorlarını tercih etmektedirler. Teknolojinin gelişmesi ile birlikte bu durum, anlamsal veri aramasını sağlayan bağlantılı veri yöntemi ile değişime uğrayacaktır. Bağlı veriler, yapılandırılmış verileri bir web ortamında yayınlamak ve verilerin birbirleri arasındaki anlamsal ilişki için, farklı web kaynakları ile bağlantı kurmak amacı ile yeni yöntemler ve kurallar belirler. Belirlenecek yeni yöntemler ise RDF ve HTTP protokolünü kullanarak gerçekleştirilir. Böylece, bir veri kaynağındaki verilerin başka bir veri kaynağındaki verilere etkin bir şekilde bağlanmasına olanak tanır.

Ontoloji, anlamsal ağ için önemli destek sağlayan bir teknolojidir. Web kaynaklarının heterojen temsillerini ele almak için bir yol sağlar. Bilgi paylaşımını ve yeniden kullanımını teşvik etmek için yapay zeka alanında ontoloji geliştirilmiştir. World Wide Web Consortium (W3C) çalışma grupları tarafından, meta veri mimarisini tanımlamak için bir yol sağlayan genel bir bilgi temsil aracı içeren Resource Description Framework (RDF) ve RDFS'yi (RDF Schema) geliştirilmiştir (Gómez-Pérez ve ark., 2002). RDF, anlamsal ağ verilerinin veri modelidir ve SPARQL, bu veri modeli için standart sorgu dilidir.

RDF veri tabanlarının kullanımı, tek bir nesneye bağlı kişiler, yerler, olaylar ve kavramlar gibi birçok varlık arasındaki karmaşık ilişkileri ifade etmek için çok uygundur. Bu veri tabanları genellikle grafik veri tabanları olarak adlandırılır.

Çünkü bilgileri, her kaynak arasındaki ilişkiyi açıklayan bilgilerle bir dizi kaynak veya düğümün birbirine bağlandığı grafikler veya ağlar halinde yapılandırırlar. SPARQL, bu veri tabanlarını sorgulamak için kullanılan dildir (Pérez ve ark., 2006). Bu sorgu dili özellikle güçlüdür çünkü kullanıcının verilere getireceği perspektifi önceden varsaymaz. Son zamanlarda, her boyutta ve konuda bilgi tabanları için yüzlerce halka açık SPARQL uç noktası piyasaya sürüldü. Bu uç noktaları kullanarak, istemci, karmaşık sorgulara doğrudan yanıtlar almak için sunucudan tek bir istek ile yanıt alabilir. Sonraki bölümlerde, doğal dil sorgularını tanımlayıp, SPARQL sorgularına dönüştürülerek ve sorgulamanın Wikidata üzerinde gerçekleştirilmesi sonucunda ortaya çıkan eksik veya hatalı verilerin iyileştirilmesi için kullanılan metodoloji anlatılmaktadır.

2. KAYNAK ARAŞTIRMASI

Kaynak araştırılması 3 aşamada analiz edilmiştir. İlk bölümde Web tarihi incelenmiş ve günümüzdeki teknolojiye kadar olan gelişim süreci anlatılmıştır. Daha sonra ikinci bölümde ise Anlamsal ağ (Semantic web) konusu tanımlanmıştır. Anlamsal Ağ'ın uygulama alanları, uygulama dilleri ve piramit yapısı hakkında bilgilendirilme yapılmıştır. Son olarak üçüncü bölümde ise tezin ana konusu olan tablo veri analizi açıklanmıştır. Yapılan analiz Wikidata bilgi tabanı kullanılarak gerçekleştirildiği için, bu arama motoru konu edinilmiş, aynı zamanda çalışılan projenin sorgu dili olarak kullanılan SPARQL yöntemi incelenmiştir.

2.1. Web Tarihi

1990'lardan beri küresel ağlar, modern toplumun haberleşme altyapısının vazgeçilmez bir parçası haline geldi. Web platformu; gazete, film, radyo ve televizyon gibi medya türlerinin dışında bağımsız bir gelişim süreci göstermiştir. Web tarihinin ne olduğunu ve nelere dâhil olmadığını bilmek, iletişim alanına fayda sağlar. Web 'in gelişim süreci, internet geçmişi ile eşit sayılamaz. Kullanıcıların birçoğu İnternet'i web ile bir tutsa da, ikisi birbirinden farklı işletimlere sahiptir. Web, internetin 'WWW' protokollerini kullanan bağlantılı bir yapıdadır (Brügger, 2012).

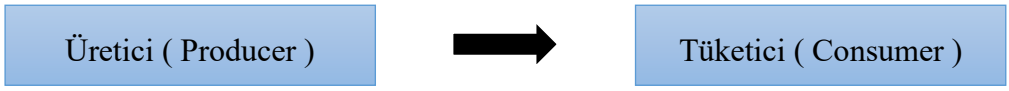
Dünyanın herhangi bir yerinde yaşayan bir kişi veya işletim sistemine sahip bir kuruluş, sayısız düzeyde bağlantılı web siteleri oluşturabilmektedir. Bu oluşan bilgi havuzu düzensiz büyümeye sebebiyet verir. Bir belgeden diğerine URL'ler tarafından oluşturulan bağlantılar, karmaşık bir ağ grafiğinin oluşmasına yol açar. Web içerisindeki anlamlı bilgilerin bağlanabilirliği grafiğin topolojisini oluşturur (Albert et al., 1999).

Web platformu web 1.0' dan itibaren aşama aşama ilerletilmiş ve günümüzde beklentilerimizi üst düzey seviyede karşılamaya devam etmektedir.

- **Web 1.0 (Monolog) :**

Web 1.0, bilgi ağı olarak da adlandırılan webin kökenidir. Kullanıcılar yalnızca web sayfaları üzerinden bilgi okuyabilir ve bilgi alışverişinde bulunabilirler (Nath ve ark.,

2014) . WWW veya Web 1.0, internet üzerinden erişilebilen birbirine bağlı bir köprü metni belge sistemidir. Ağın ilk nesli olan Web 1.0'ın Berners-Lee' ye göre "salt okunur bir ağ" olarak kabul edilebileceği belirtilmiştir (Shivalingaiah & Naik, 2008).

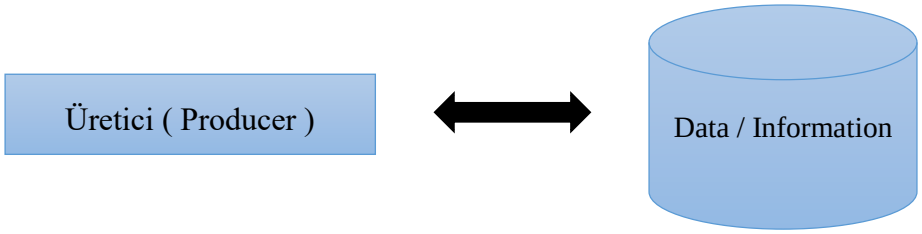


Şekil 1. Web 1.0 (Monolog)

- **Web 2.0 (İnteraktif web) :**

Berners Lee'nin web kimliğine yaklaşımı göz önüne alındığında, Web 2.0 bir "okuma-yazma" ağı olarak bilinmektedir (Shivalingaiah & Naik, 2008). Bu platform, ağ içeriğini zenginleştirmek ve diğer ağ kullanıcılarıyla bağlantı kurarak Web'in görünümüne önemli bir katkı sağlamak için tasarlanmıştır. Web 2.0 terimi, WWW 'nin geliştirilmiş bir versiyonu olarak ortaya çıkmıştır.

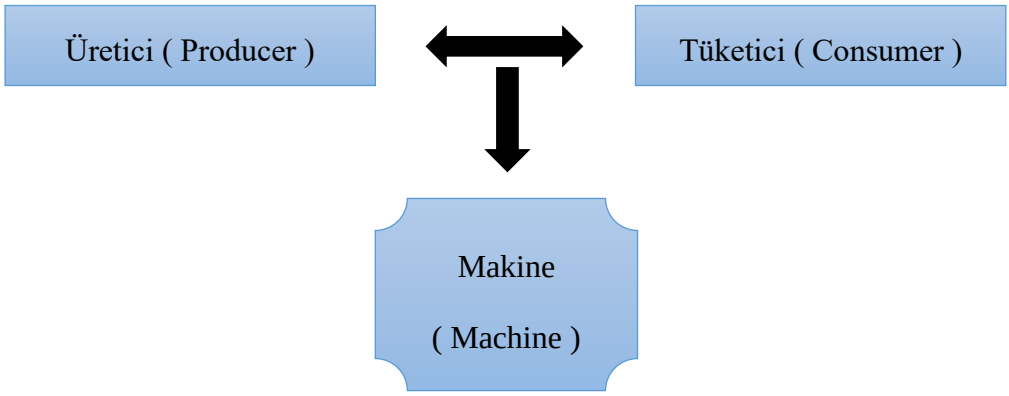
Web 2.0'da, Web kullanıcıları yalnızca içeriği okumakla kalmaz, aynı zamanda Web 1.0' dan ziyade içeriği çevrimiçi olarak yazabilir, değiştirebilir ve güncelleyebilir, işbirliğini destekleyebilir ve kolektif zekanın bir araya gelmesine olanak sağlamaktadır (O'reilly, 2005).



Şekil 2. Web 2.0 (İnteraktif web)

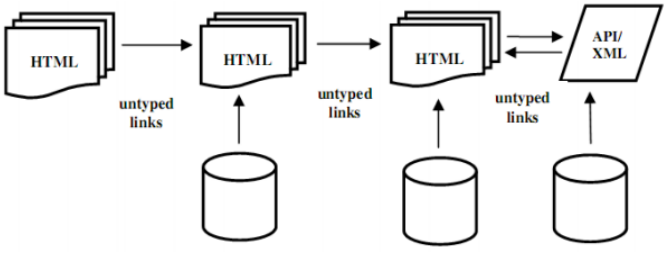
- **Web 3.0 (Semantik web) :**

Web 3.0'ın ana fikri, farklı uygulamalar üzerinde daha verimli arama, otomasyon, entegrasyon ve yeniden kullanım için yapılandırılmış verileri tanımlamak ve ilişkilendirmektir. Anlamsal ağ, veri yönetiminin iyileştirilmesinde, mobil internet erişilebilirliğinin teşvik edilmesinde, yaratıcılığı ve yeniliği ve küreselleşme olgusunun desteklenmesinde, müşteri memnuniyetinde ve sosyal ağlar üzerinde yapılan işbirliği alanlarında önemli rol oynar. (Hendler, 2009).



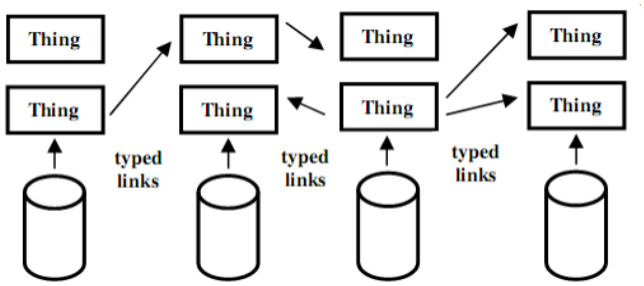
Şekil 3. Web 3.0 (Semantik web)

Web 3.0, daha önce bir belge ağı olarak tanımlanan küresel veri tabanlarını ve web yönelimli mimariyi destekler. Esas olarak statik HTML belgeleriyle ilgilenir, ancak dinamik olarak oluşturulan sayfalar ve alternatif biçimler, mümkün olduğunca aynı kavramsal düzen standartlarını izlemeli ve belgeler arasında bağlantılar olmalıdır (Choudhury, 2014).



Şekil 4. Belge ağı (Kaynak: Patel ve ark., 2013)

Veri ağı, ansiklopedi bilgileri, ilaçlar ve sağlık verileri, müzik, kitaplar ve bilimsel makaleler, sosyal ağ temsilleri, jeo-uzamsal bilgiler ve diğer birçok bilgi, içerik ve bağlantıların anlamlarını içeren küresel bir veri tabanıdır (Patel ve ark., 2013)



Şekil 5. Veri ağı (Kaynak: Patel ve ark., 2013)

- **Web 4.0 :**

Web 4.0, Ultra-Akıllı elektronik ajan, Simbiyotik web ve Ubiquitous web olarak düşünülebilir. İnsanlar ve makineler arasındaki simbiyozdaki etkileşim, simbiyotik ağın arkasındaki güçtür. İnsan beyni kadar güçlü web 4.0 kullanılarak, telekomünikasyonun gelişmesinde ilerleme ve nanoteknolojideki ilerleme için kontrollü ara yüzler oluşturulur (Choudhury, 2014).

Başka bir ifadeyle, makineler web içeriğini okuma konusunda daha hassas olacaktır. Bunun yanı sıra, web sitelerini üstün performans ile hızlı bir şekilde yüklemek ve daha komuta eden ara yüzler oluşturmak için ilk önce neyin yürütüleceğine karar verme yönünde tepki verebilecektir (Patil ve ark., 2018).

2.2. Anlamsal Ağ (Semantic Web)

Bilgi yönetimi, bir organizasyon içinde bilginin elde edilmesi, erişilmesi ve sürdürülmesi ile ilgilenir. Bilgi ile daha fazla üretkenlik elde edilebileceği, yeni değer oluşturabileceği ve rekabet gücünü artırabilecek bir entelektüel varlık olarak görüldüğü için, büyük işletmelerin önemli bir faaliyeti olarak ortaya çıkmıştır. Bilgi yönetimi, coğrafi olarak dağınık departmanları olan uluslararası kuruluşlar için özellikle önemlidir. Çoğu bilgi şu anda metin, ses ve video gibi zayıf yapılandırılmış bir biçimde

mevcuttur. Bilgi yönetimi perspektifinden, mevcut teknoloji aşağıdaki alanlarda sınırlamalardan ibarettir (Davies ve ark., 2003) :

- **Bilgi arama:** Mevcut anahtar kelime tabanlı aramalar, belirli terimleri farklı anlamlarda içeren alakasız bilgilere ulaşabilir. İstenen içerikle ilgili aynı anlama gelen farklı terimler kullanıldığında da bilgiyi kaçırırlar. Bilgi alımı geleneksel olarak belirli bir sorgu ile bilgi deposu arasındaki ilişkiye odaklanır. Öte yandan, seçilen bilgi parçaları arasındaki karşılıklı ilişkilerin kullanılması, izole edilmiş bilgileri anlamlı bir bağlama yerleştirebilir. Bu şekilde ortaya çıkan örtük yapılar, kullanıcıların bilgileri daha verimli kullanmasına ve yönetmesine yardımcı olur.
- **Bilgi çıkarma:** Bilgi kaynaklarından ilgili bilgileri çıkarmak için kullanıcıların taraması ve okuması gereklidir. Bunun nedeni, otomatik aracıların bu tür bilgileri metinsel temsillerden çıkarmak için gereken sağduyu bilgisine sahip olmaması ve farklı kaynaklara dağıtılan bilgileri bütünleştirememeleridir.
- **Bakım:** Zayıf yapılandırılmış metin kaynakları büyüdüğünde bu kaynakların incelenmesi zor ve zaman alıcı bir faaliyettir. Bu tür koleksiyonları tutarlı, doğru ve güncel tutmak, anormallikleri tespit etmeye yardımcı olan mekanikleştirilmiş anlambilim temsillerini gerektirir.
- **Otomatik belge oluşturma:** Kullanıcı profillerine veya ilgili diğer yönlere göre dinamik olarak yeniden yapılandırılan uyarlanabilir web sitelerini etkinleştirir. Yarı yapılandırılmış verilerden yarı yapılandırılmış bilgi sunumlarının üretilmesi, bu bilgi kaynaklarının anlambiliminin makine tarafından erişilebilir bir temsilini gerektirir.

Anlamsal ağ'ın amacı, aşağıdaki belirtilen özelliklere sahip çok daha gelişmiş bilgi yönetim sistemlerine izin vermektir

(Antoniou & Harmelen, 2004):

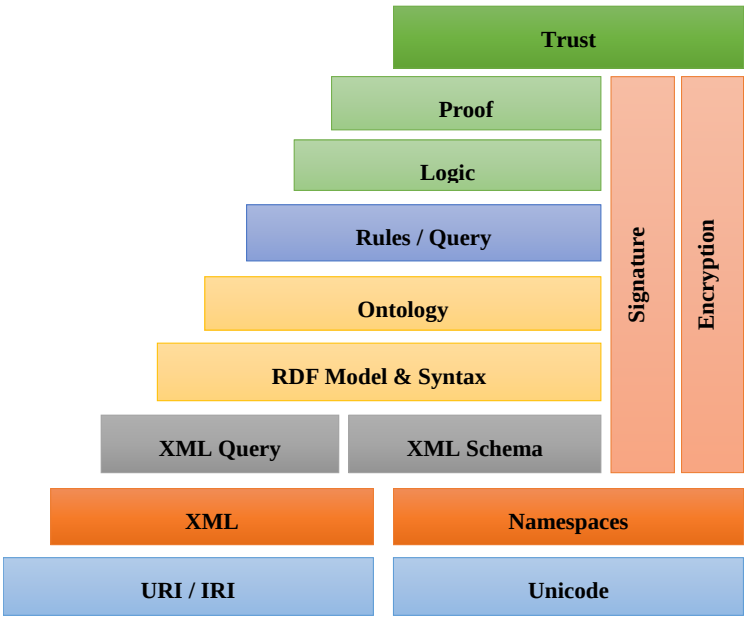
- Bilgi, anlamına göre kavramsal mekânlarda organize edilecektir.
- Otomatik araçlar, tutarsızlıkları kontrol ederek ve yeni bilgiler çıkararak bakımı destekleyecektir.
- Anahtar kelime tabanlı aramanın yerini sorgu yanıtlama alacak: istenen bilgi alınacak, ayıklanacak ve sunulacaktır.

- Birkaç belge üzerinden sorgu yanıtlama desteklenecektir.
- Bilginin belirli kısımlarını kimlerin görebileceğini belirlemek mümkün olacaktır.

Anlamsal ağ, uluslararası standartlar kuruluşu World Wide Web Konsorsiyumu tarafından yönetilen ortak bir harekettir. W3C'ye (Miller & Swick, 2003) göre, "Semantik web, verilerin uygulama, işletme ve topluluk sınırları arasında paylaşılmasına ve yeniden kullanılmasına izin veren ortak bir çerçeve sağlar". Makine tarafından işlenebilir meta-verilerin kullanımına dayalı olarak gelişmiş bilgi erişimi sağlayan bir Semantik web kavramı önerilmiştir (Berners-Lee ve ark., 2001).

Anlamsal ağ'ın temel amacı, kullanıcıların formasyonda daha kolay bulmalarını, paylaşmalarını ve birleştirmelerini sağlayarak mevcut web' in evrimini yönlendirmektir. Anlamsal ağ, başlangıçta tasavvur edildiği gibi, makinelerin anlamlarından yola çıkarak karmaşık insan isteklerini anlamalarını ve bunlara yanıt vermelerini sağlayan bir sistemdir. Böyle bir anlayış, ilgili bilgi kaynaklarının anlamsal olarak yapılandırılmasını gerektirir.

Tim Berners-Lee, Anlamsal ağ'ı başlangıçta şu şekilde ifade etmiştir (Berners-Lee, 1999) : "HTML ve Web tüm çevrimiçi belgeleri büyük bir kitap gibi gösterseydi, RDF, şema ve çıkarım dilleri dünyadaki tüm verileri devasa bir veri tabanı gibi gösterecekti". Tim Berners-Lee, anlamsal ağ için, bir diyagram kullanarak temsil edilen katmanlı bir mimari önermiştir. Semantik web' in gelişimi adım adım ilerler, her adım diğerinin üzerine bir katman oluşturur. Aşağıda gösterilen grafik, anlamsal ağ tasarımının ve vizyonunun ana katmanlarını tanımlayan yapılarını göstermektedir .



Şekil 6. Anlamsal ağ'ın katman yapısı

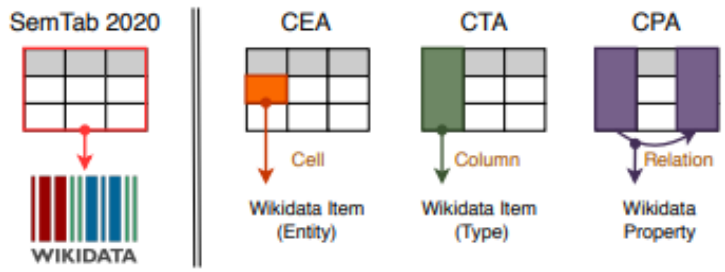
- **Unicode ve URI:** Unicode, hangi dil tarafından yazılmış olursa olsun, herhangi bir karakteri benzersiz bir şekilde temsil etmek için kullanır ve Uniform Resource Identifier (URI), tüm kaynaklar için benzersiz tanımlayıcılardır. Unicode ve URI'nin işlevselliği, anlamsal ağ için dil yığını içinde benzersiz bir tanımlama mekanizmasının sağlanması olarak tanımlanabilir.
- **XML:** Kullanıcı tanımlı bir kelime dağarcığı ile yapılandırılmış web belgeleri yazılmasına izin veren bir dildir. XML, belgeleri web üzerinden göndermek için özellikle uygundur. XML, kullanıcının yeni etiketlerinin anlamını diğer kullanıcılara iletmek için yerleşik bir mekanizmaya sahip değildir.
- **RDF:** Resource Description Framework, web nesneleri (kaynaklar) hakkında basit ifadeler yazmak için varlık-ilişki modeli gibi temel bir veri modelidir. RDF, terimlerin ve kavramların anlamlarını bilgisayarların kolayca işleyebileceği bir biçimde ifade etme teknolojisini sağlar.
- **RDF Şema (RDF Schema):** RDF modelleri için önceden tanımlanmış, temel tip bir sistem sağlar. RDF schema, web nesnelerini hiyerarşiler halinde düzenlemek için modelleme temelleri sağlar. Temel öğeler, sınıflar ve özellikler, alt sınıf ve alt özellik ilişkileri ve etki alanı ve aralık kısıtlamalarıdır.

- **Ontoloji (Ontology):** Çıkarım yeteneği ile belirli bir alanı tanımlamak için kullanılan terimler topluluğudur.
- **Mantık katmanı (Logic layer):** Ontoloji dilini daha da geliştirmek ve uygulamaya özel bildirimsel bilginin yazılmasına izin vermek için kullanılır.
- **Kanıt katmanı (Proof layer):** Gerçek tümdengelim sürecini, kanıtların web dillerinde temsilini ve kanıt doğrulamasını içerir.
- **Güven katmanı (Trust layer):** Dijital imzaların ve güvenilir acentelerin tavsiyelerine veya derecelendirme ve sertifikasyon kuruluşları ile tüketici kuruluşlarına dayalı diğer bilgi türlerinin kullanılmasıyla ortaya çıkmıştır.

2.3. Tablo Verilerinin Analizi

Bilgi diyagramları, ontoloji ve kelime tanımları ile ifade edilen graf biçiminde bir veri tabanıdır (Yumusak, 2020). Belirli alanlar için yerel infografik yapıları oluşturabilir, ancak infografikler genellikle alanların birbirleri arasındaki anlamsal ilişkileri veya özel alan bilgilerini içerebilir. CSV dosyaları gibi tablo verileri, önemli içeriği temsil etmek için verileri yazdırmanın sık tercih edilen bir yoludur. Ancak, çok fazla meta veri bağlantısı kaybolur ve bu da doğrudan erişimi ve kullanımı zorlaştırır. Ayrıca, veriler sıklıkla gürültülü ve belirsizdir. Bu nedenle, mevcut tablolardaki verileri birleştirmek, verileri silmek veya aynı fikrin farklı temsilleri arasında bir eşleşme grafiği çizmek gibi yoğun çaba gerektiren bir hedef haline gelir (Abdelmageed ve ark., 2020).

Tablo açıklama yöntemi, üç alt göreve bölünerek yeni bir terminoloji sunar:



Şekil 7. SemTab - Tablo verileri ile Wikidata eşleştirilmesi (Kaynak: Nguyen ve ark., 2020)

- **CTA (Column Type Annotation):** Sütunları ontoloji türlerine bağlamaya odaklanan şema düzeyinde bir eşleştirmedir.
- **CEA (Cell Entity Annotation):** Varlıklar arası bağlantı kurma olarak da bilinen eşleştirmedir.
- **CPA (Column Property Annotation):** Yalnızca tablonun sütunlarını ontoloji nitelikleri aracılığıyla birbirine bağlamaya odaklayan şema düzeyinde eşleştirmedir.

Tablo verilerinde, verilere açıklama eklemek önemli hedeflerden biridir, çünkü doğru açıklama ile çok fazla bilgiye sahip olmadan başka bilgiler de elde edilebilir (Kim ve ark., 2020). Bu sebeple, yerinde bir açıklama, anlambilimin anlaşılabilirliğini gösterir. Bu açıdan bilgi grafik tasarımın anlamını öğrenmek çok önemlidir. Çünkü yanlış değerlendirmeler, veri işleme hattında başka yanlış değerlendirmelere yol açabilir. Zamanla yanlış sonuçlar her yere yayılabilir. Kullandığımız veriler Wikidata bilgi tabanına dayanmaktadır. Her veri giriş Wikidata bilgi tabanında listelenir ve bu sayede Wikidata bilgi tabanındaki anlamsal bilginin tanımlanmasına olanak tanır.

2.3.1. Wikidata Bilgi Tabanı

Wikidata bilgi tabanı, aktif bilgi veri deposudur. Bilgileri bir araya getirmek ve yaymak için benzersiz bir fırsat sağlar. İşbirliğine dayalı düzenlemeyi destekleyen tek büyük anlamsal ağ (semantic web) kaynağıdır. Bilgiyi dağıtma açısından düşünüldüğünde; Wikipedia ile doğrudan entegrasyonu sayesinde oluşturulan içeriğin, milyonlarca tüketiciyle yüzlerce dilde paylaşılmasına izin verir (Piscopo ve ark., 2019).

Buna ek olarak, artan sayıda kontrollü kelime dağarcığı ve ontolojide benzersiz kavram tanımlayıcılara bağlantılar sağlar, böylece mevcut bilgi tabanları ile ve bunlar arasındaki entegrasyonu kolaylaştırır. Sözcük bilgisi, bilgi grafiğinde temsil edilen ve artan miktarda anlamsal bilgi ile birleştiğinde, doğal dil işleme için güçlü bir kaynak sağlar. Bu uygulama sayesinde, içeriğinin diğer uygulamalarda yeniden kullanılması ve yeniden dağıtılmasıyla ilgili tüm engelleri kaldırır.

2.3.2. Wikidata Sorgu Hizmeti

Wikidata, Wikimedia vakfı tarafından denetlenen ve binlerce kullanıcıdan oluşan bir topluluk tarafından ortaklaşa düzenlenen yeni bir bilgi tabanıdır. Wikidata'nın amacı, Wikimedia ve Wikipedia ile ortak çalışarak bilgi kaynağı oluşturulmasıdır. Bilgileri tek bir yerde depolayarak, birden çok yerde yararlanılmasını sağlayabilir. Wikidata sorgu hizmeti sayesinde ulaşılması istenen bilgileri zamandan tasarruf ederek daha hızlı ve verimli şekilde çalışma olanağı sağlanır. Wikidata veri modeli, varlıkların özellikleri arasından anlamsal ilişki ile birbirine bağlı, etiketli bir grafiğe dayanmaktadır.

Wikidata, verileri, "ifadeler" adı verilen yapılandırılmış bir biçimde düzenler. Bu ifadeler üç ana bölümden oluşur: öğeler ("özne"), tahminler ("özellik") ve nesneler ("değerler"). Örneğin, Wikidata, İstanbul'daki Sultan Ahmet Camii'nin yapımına başlangıç tarihini "Sultan Ahmet Camii" (madde), "başlangıç" (özellik), "1609" (değer) olarak kaydeder. Wikidata öğelere, özelliklere ve değerlere atıfta bulunmak için kendi harf ve sayı sistemini kullanır. Bahsedilen ifadenin dahili temsili "Q80541 P571 1609" şeklindedir.

Wikidata'daki tüm veriler sorgulanabilir. Wikidata'nın sorgu dili SPARQL veya W3C'nin bağlantılı verileri sorgulamak için geliştirdiği "SPARQL Protokolü ve RDF Sorgu Dili" dir. Bu SPARQL sorguları, Wikidata'nın gücünün altını çizerek, kullanıcıların karmaşık sorular sormasına (Boğaziçi Köprüsü'nün 1973 yılındaki resimleri, hangi tarihi kitapların içerisinde yer alıyor?) ve sonuçları görselleştirmesine olanak tanımaktadır. Görselleştirme olanakları, diğer veriler arasındaki grafikleri, resim galerileri, haritalar ve zaman çizelgelerini içerir. Görselleştirmeler web sitelerine yerleştirilebilir ve sorgulardan elde edilen veriler indirilebilir ve ücretsiz olarak yeniden kullanılabilir.

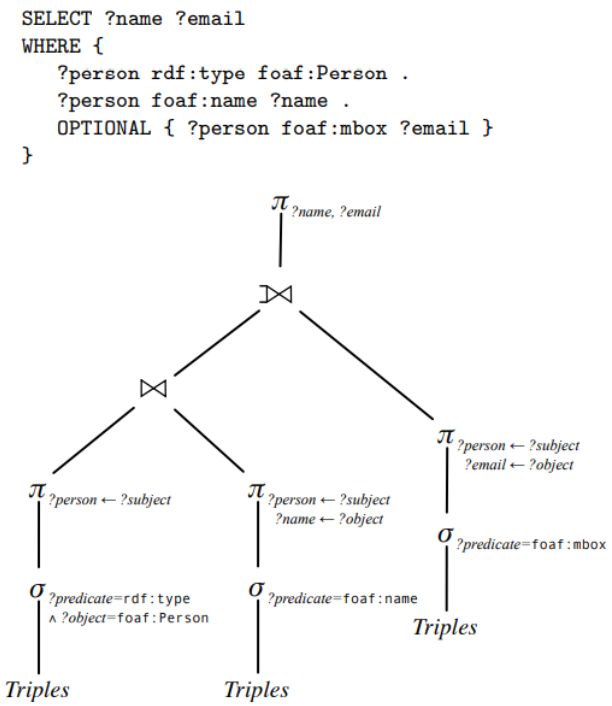
2.3.3. SPARQL

SPARQL protokolü ve sorgu dili, RDF kullanılarak kodlanmış verilerden bilgi çıkarmak için geliştirilmiş yeni bir World Wide Web Consortium (W3C) önerisidir (Schmidt ve ark., 2010). RDF veritabanları (özne, yüklem, nesne) üçlü gruplar halindedir. Her üçlü, özne ve nesne arasındaki ikili ilişki yüklemini kodlar, yani tek bir

bilgi olgusunu temsil eder. Homojen yapıları nedeniyle, RDF veri tabanları etiketlenmiş ve yönlendirilmiş grafikler olarak görülebilir.

SPARQL, anlamsal bilgi grafiği ile eşleşen bir sorgu dilidir. Kısacası, A kaynaklı bir veri tabanı verildiğinde, sorgu A ile eşleşen bir şema ile biçimlendirilir ve eşleşmeden elde edilen değer bir cevap vermek olarak ele alınır. Sorgulaması yürütülecek veri kaynağı A, birden fazla kaynak bulundurabilir. Sorgu dilinin ifade gücünün belirlenmesi, kullanıcıların hangi sorguları oluşturabileceğini anlamak ve sorgunun zorluğunu değerlendirmek için çok önemlidir (Angles ve ark., 2008). İşlev ve karmaşıklığı kavrayabilmek ise bir sorgu dilinin tasarımı için önemli bir faktördür.

Bir SPARQL SELECT sonuç formu, doğrudan değişkenleri ve bunların bağlantılarını döndürür (Cyganiak, 2005). SELECT * sözdizimi, sorgudaki tüm değişkenleri seçen bir kısaltmadır. SPARQL SELECT sorgusunun sonucu RDF düğüm tablosudur. Bu tablolar, RDF ilişkilerini gösterir. Böylece; ilk kısım, sorgu çalıştırıldığında görünen sorgu sonucu değişkenidir ve ikinci kısım, sorgu modu ile birlikte WHERE yan tümcesidir. Son kısım ise isteğe bağlı değiştiricidir. Şekil 2.5, SPARQL sorgusunu ve bunun ilişkisel operatör ağacına dönüşümünü göstermektedir:



Şekil 8. SPARQL ilişkisel sorgu ağacı (Kaynak: Cyganiak, 2005)

SPARQL sorgu dili’ nin en büyük avantajı, grafik desenlerini eşleştirerek RDF grafik verilerindeki ilişkilerde gezinmektir. Bununla birlikte, basit desenler, verilerdeki daha ayrıntılı ilişkileri inceleyen daha karmaşık desenlerle birleştirilebilir. İlişkileri keşfetmek için temel modelleri, desen kombinasyonlarını ve bu birleşimleri kullanarak, bulunan çözümlere ilişkin anlayışınızı genişletebilecek başka modeller ekleyebilirsiniz.

3. MATERYAL VE YÖNTEM

Bu bölümde, ağ içerisindeki bağlantılı verileri keşfetmek ve birbirleri arasındaki anlamsal ilişkiyi sorgulamak için kullanılan SPARQL sorgu dilinin üzerinde geliştirilen sürecin teknik konularından bahsedilmiştir. Tez çalışmasında önerilen düzeltme ve eşleme yöntemleri, Yumusak (2020) tarafından geliştirilen bilgi tabanı eşleme programı tarafından toplanan veriler üzerinde analiz edilmiştir. Bağlantılı verilerin veri eşleşmesi için nasıl bir yol izlediği ve hatalı girdi senaryolar üzerinde yapılan araştırmalar ele alınmıştır.

3.1. Veri Eşleşmesi

Veri eşleşmesi, herhangi bir veri entegrasyon sürecinin can damarıdır. Uygun bir veri eşleştirme stratejisi yoksa filtreleme hataları meydana gelir ve bu da düşük veri kalitesine neden olur. Bu, iş analizini ve iş kararlarını doğrudan etkiler. Bu nedenle, veriler arası bağlantı kurma işlemi boyunca bütünlüğün korunması esastır.

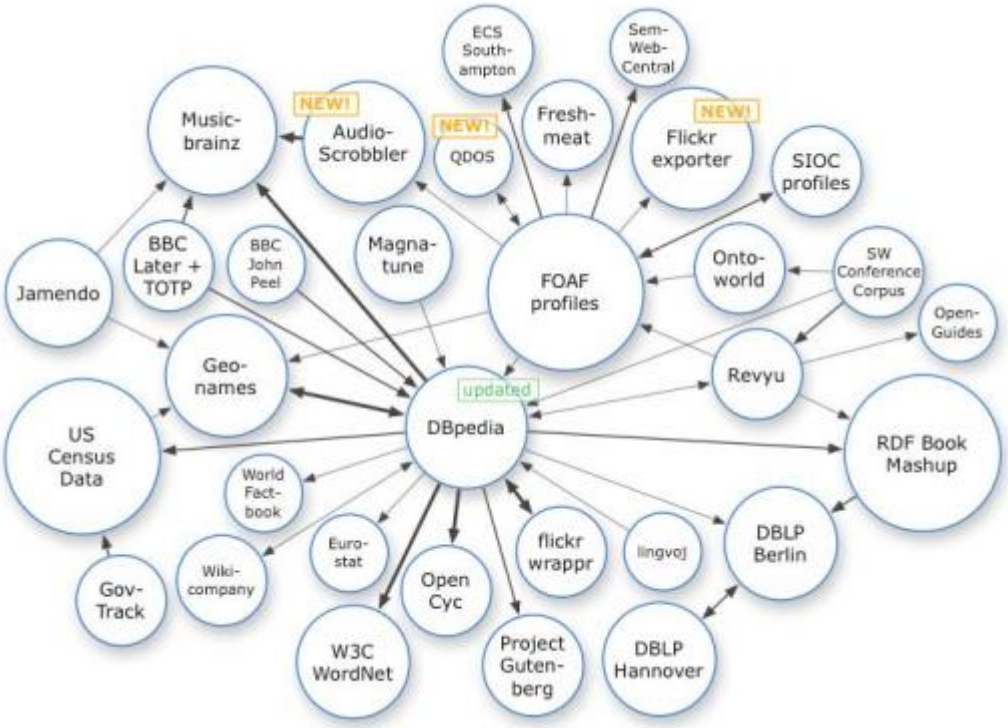
Etkili bir veri eşleştirme grafiği, veri kaynakları arasındaki bağlantıları ve ilişkileri gösterecek, böylece ilişkilendirmeyi görüntülerken olası sorunlar ortaya çıkacaktır. Bu durum, veri entegrasyon sürecindeki olası sorunları önleyebilir. Veri eşleştirme ayrıca ihmal, hata ve tekrarlama riskini azaltarak veri analizini daha doğru hale getirir. Aynı zamanda, gelişmiş raporlama yetenekleri, veri güvenliği, yönetimi ve verilerin birbirleri arasındaki uyumu açısından fayda sağlar.

Verilerin taşınması, işlenmesi ve yönetilmesi için veri eşleşmesi gerçekleştirilir (Christen, 2012). Sonuç olarak, veri eşleştirilmesinin amacı, tek bir veri setinde birden fazla veriyi homojen hale getirmektir. Veri eşleştirmek ve kaynakları, güvenilir bir veri tabanında entegre etmek için birleştirme teknolojisini kullanılarak temiz bir süreç oluşturabilir. Veri miktarı arttıkça ve verileri kullanan sistemlerin karmaşıklığı arttıkça, veri eşleşme süreci daha karmaşık hale gelir ve böylece otomasyon için güçlü araçlara olan ihtiyaç ortaya çıkar (Olteanu ve ark., 2006).

3.1.1. Bağlantılı Veri (Linked Data)

Bağlantılı veri, web'de yapılandırılmış verileri yayınlamak ve verileri farklı veri kaynakları arasında bağlamak için RDF ve HTTP'yi kullanmakla ilgilidir ve bir veri kaynağındaki verilerin başka bir veri kaynağındaki verilere etkin bir şekilde bağlanmasına izin verir. Bağlantılı veri ilkeleri ilk olarak 2006 yılında Berners-Lee tarafından ana hatlarıyla belirtilmiştir (Bizer ve ark., 2008).

Bağlantılı veri tarayıcıları, HTML sayfaları arasındaki bağlantıları takip etmek yerine, kullanıcıların RDF bağlantılarını izleyerek farklı veri kaynakları arasında gezinmesini sağlar. Bağlantılı açık veri "bulutunun" aralığı ve ölçeğinin bir göstergesi Şekil 9'da verilmiştir. Bu diyagramın gösterdiği gibi, temel bağlantılı merkezler DBpedia ve Geonames gibi sitelerdir.



Şekil 9. The Linking Open Data Cloud Diyagramı, Şubat 2008 (Kaynak: Bizer et al., 2008)

Bağlantılı veri teknolojisinin ürettiği veriler, web'de makine tarafından okunabilir bir formatta yayınlanan verilerdir. Anlamları açıkça belirtilmiştir ve birden çok dış veri kaynağıyla ilişkilendirilmiştir. Bağlantılı veri stratejisi, URI' nin nesneyi tek bir HTTP şemasına göre tanımlamasını gerektirir (Hartig & Langegger, 2010). URI yalnızca küresel olarak benzersiz bir tanımlayıcı olarak hareket etmekle kalmaz, aynı zamanda belirli varlık verilerinin yapılandırılmış bir temsiline erişim sağlar.

Berners-Lee, internette veri yayınlamak için bir dizi bağlantılı veri ilkesini ana hatlarıyla belirtmiştir, böylece yayınlanan tüm veriler dört ana alanda tek bir küresel veri alanının parçası olmaktadır (Bizer ve ark., 2011):

- URI, farklı şeylerin ayırt edilmesine veya bir kayıttaki bir öğenin başka bir kayıttaki başka bir öğeyle aynı olduğunun anlaşılmasına olanak tanır.
- HTTP protokolü, kaynakların alınması için basit bir mekanizma sağladığından, nesneleri URI ve protokol kombinasyonu aracılığıyla tanımlamak mümkün olduğundan daha kolay hale gelir.
- URI'yi etkin kullanmak için sorgu alanında RDF ve SPARQL uygulamaları kullanılmaktadır. RDF, Web üzerinde veri yayınlamak, depolamak ve varyasyon için kullanılan bir grafik gösterim formatıdır. Öte yandan, SPARQL, RDF tarafından depolanan verileri almak ve değiştirmek için kullanılan bir sorgu dilidir, böylece internette herhangi bir veri tabanında, veri araması yapılabilir ve ilişkiler keşfedilebilir.
- Yeni bilgiler, mevcut kaynaklarla bağlanılarak mevcut veriler yeniden kullanabilir ve ara bağlantı gerçekleştirebilir.

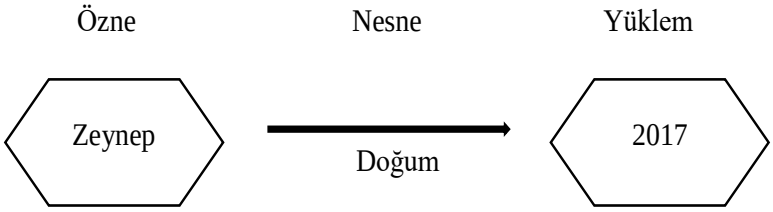
Hyper Text Markup Language (HTML), belgeleri web üzerinde düzenlemek ve entegre etmek için araçlar sağlarken, RDF, küresel olarak bilinen nesneleri tanımlayan verileri düzenlemek ve birbirleri arasında ilişki kurmak için kapsamlı bir veri tabanı tasarımı sağlar. Geleneksel bir belge web sitesini barındıran veri alanı, HTML sayfaları arasındaki bağlantılardan oluşur, veri merkezi hattı ise bir RDF bağlantısıdır (Bizer ve ark., 2008).

RDF bağlantısı, basitçe, bir veri parçasının başka bir veri parçasıyla ilişkili olduğunu belirtir. Bu ilişkilerin farklı türleri olabilir. Örneğin, insanlar hakkındaki verileri birbirine bağlayan bir RDF bağlantısı, iki kişinin birbirini tanıdığını belirtebilir; Bir kişi

hakkındaki bilgileri, bir bibliyografik veri tabanındaki yayınlar hakkındaki bilgilerle birleştiren bir RDF bağlantısı, bir kişinin belirli bir makalenin yazarı olduğunu belirtebilir.

RDF veri modeli, düğümler ve etiketli yönlendirilmiş grafikler biçimindeki bilgileri temsil eder. Veri modeli, farklı kaynaklardan toplanarak farklı şemalar elde edilip, heterojen yapıya sahip bilgilerin bağlantı tasarımı için oluşturulmuştur. RDF, webde kullanılan diğer veri modellerinde doğrulama yapabilen evrensel bir dil olarak kullanılmak üzere tasarlanmıştır.

Aşağıda, veri modelinin kullanım şekli gösterilmiştir:



Şekil 10. RDF veri sorgu modeli

Özne (subject), nesne (object) ve yüklemden (predicate) oluşan RDF üçlüsünün, yüklemin URI'si bağlantının türünü gösterir (Bizer, 2011). Bu veri modeli sayesinde, daha fazla bilgi edinmek için internet üzerinden RDF grafiğindeki herhangi bir URI sorgulanabilir ve aynı zamanda, RDF kullanılarak, farklı şemalar ile temsil edilen bilgiler tek bir grafikte görüntülenebilir (Heath ve ark., 2011).

3.1.2. Meta Veri

Meta veri bilgi hakkında bilgi, veri hakkında veri olarak tanımlanır. Meta verinin genellikle üç temel bileşeni olduğu düşünülür (Gartner, 2016). Bunlar,

- semantics: meta verinin çıktığı element alanlarının anlamları
- syntax : meta verilerin, bir elektronik tabloda veya veritabanı tablosunda veya XML gibi daha genel bir biçimde kodlanma şekli

- content rules : meta verinin içeriğini yöneten kurallar; ne kaydedilmeli, hangi biçimde olmalı ve ne hariç tutulmalı.

Meta verilerde anlambilim, bir standardın alanları veya öğeleri ile bunları dolduran içerik arasındaki ilişkiyi tanımlamak için kullanılan genel terimdir. Bir meta veri standardı genellikle bu alanların bir kümesini tanımlar.

Tablo 1. Beş meta veri standardında 'başlığın' tanımı (Kaynak: Gartner, 2016)

Metadata standard	Başlığın tanımı
Dublin Core	Kaynağa verilen ad. Tipik olarak, bir başlık bir kaynağın resmi olarak bilindiği isim
Anglo-American Cataloging Rules (AACR2)	Bir öğenin baş adı ve herhangi bir alternatif başlığı içerir
Encoded Archival Description	Tanımlanan malzemelerin resmi veya tedarik edilen adı
VRA Core (visual objects)	Bir esere veya resme verilen başlık veya tanımlayıcı ifade
PBCore (public broadcasting)	Varlıkla alakalı bir ad veya etiket

Meta veriler kapsamlı olarak özetlenirse aşağıdaki koşullarda yardımcı olmaktadır (Steinacker ve ark., 2001) :

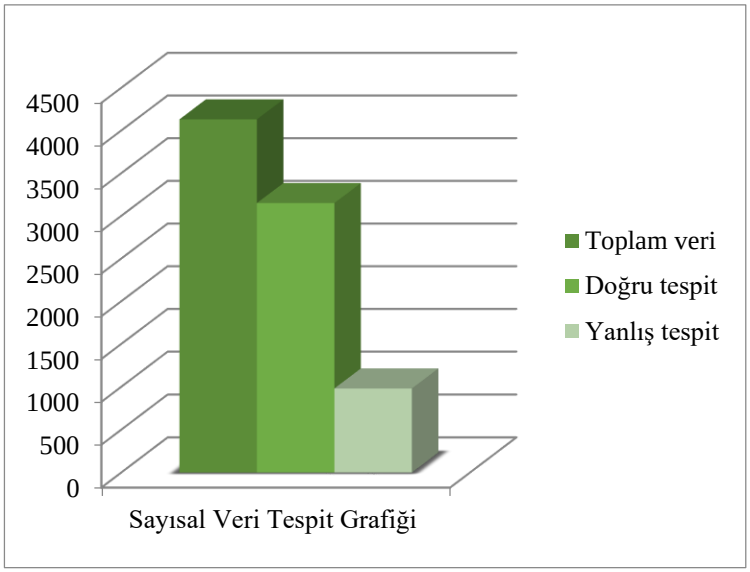
- Verilerin anlamını özetlemek,
- Kullanıcıların verileri aramasına izin vermek,
- Kullanıcıların verinin ne olduğunu belirlemesine izin vermek
- Veri kullanımını etkileyen bilgiler vermek (yasal koşullar, büyüklük, yaş vb.)
- Diğer kaynaklarla ilişkileri belirtmek.

3.2. Veri Eşleşme Olası Hatalı Girdi Senaryoları Ve Önerileri

Yapılan bu çalışma içerinse kullanılan veriler SemTab adı altında düzenlenen yarışmadan alınmıştır (*Semantic Web Challenge on Tabular Data to Knowledge Graph Matching,* 2020.). Tablo verileri içerinde anlamsal ilişkiler söz konusudur. Bu bağlantı veriler Wikidata içerisinden çekilmektedir. Yarışmanın amacı; verilerin içerisinde hatalı yazımların söz konusu olduğu ve bu hataların kodlama sistemi ile giderilmesi sağlatılarak bağlantının kaynağının tespit edilebileceğidir. Analiz sonucunda elde edilen bulgular sonucunda hatalı veriler tespit edilmiştir. Hatalı girdiler doğrultusunda yanlış sonuçlar elde edilmiş veya hiç sonuç alınamamıştır. Hatalı girdi senaryoları üç farklı sınıfta kategoriye ayrılmıştır: (1) Sayısal hatalar, (2) Yazım hataları ve (3) Tarih biçimleri olmak üzere hata tespitleri bulunmuştur. Sayısal hatalar kendi içinde; yuvarlama hatası, noktalama hatası ve negatif sayı hatası şeklinde incelenmiştir. Sayısal hatalar doğrultusunda, virgül ile sayının nasıl biçimlendirilebileceği araştırılmış ve sunulan çalışmalar ile hatanın düzeltilmesi olası görülmüştür. Aşağı ve yukarı yuvarlama sisteminde yakın sayılar göz önüne alınarak, hem pozitif hem de negatif sayılar üzerinden yapılan araştırmalar ile veri analizinin verimliliğinin artırılması hedeflenmektedir. Yazım hataları ise; özel karakter hatası, harf hatası ve boşluk hatası olmak üzere araştırılmıştır. Özel karakter hatası içerisinde istenmeyen kelimelerin çıkarılması, eksik-fazla harf hatası üzerinden, kelimenin arama motorları içerisinde doğruluğu teyit edilerek kelimelerin tamamlanması, kelimeler arasında fazla boşluk ve birleşik hallerin düzeltilmesi ile boşluk hatasının giderilmesi sağlanmıştır. Tarih biçim hatasında ise yeterli bir literatür çalışması bulunmamaktadır, bu yüzden sadece Wikidata sorgu sayfası içerisinde yapılan bir örnek ile öneri sunulmuştur. Ayrıca kullanılan veriler İngilizce altyapılı hazırlanılmıştır. Fakat, farklı bir dil üzerinden veri analizi gerçekleştirilmesinin, Google Scholar yardımı ile mümkün olabileceği ön görülmektedir.

3.2.1. Sayısal Hatalar

Analizi yapılan bu tez için 9460 tane farklı veri incelemesi yapılmıştır. İçerisinde toplam 4130 tane farklı sayısal veri bulunmaktadır. Kullanılan yazılım sonucunda 3150 tane doğru tespit, 980 tane ise hata tespit edilmiştir.



Şekil 11. Sayısal veri tespit grafiği

Sayısal veri analizi üç farklı alan da incelenmiştir.

1. Noktalama Hatası
2. Yuvarlama Hatası
3. Negatif Sayı Hatası

Tespit edilen bu hatalı alanlar için çözüm önerileri getirilmiştir.

1. Noktalama Hatası

Noktalama hatası sayısal hatalar içerisinde çok sık rastlanmıştır. Dosya içeriği ile Wikidata içeriği eşleştirilerek hata tespit edilmiştir. Hataları sebebiyle yazılım sonucunda doğru sonuç alınamamış, sayısal olan veriler programda yazdırılamamıştır. Tespit edilen hatalar için çözüm önerilmiştir. Çözüm analizi, python kütüphaneleri ile Google içeriğindeki çalışmalar üzerinden yapılmıştır.

Python programlama dilinde sayıların virgülle nasıl biçimlendirileceğini araştırılmıştır. Bir sayıyı virgülle biçimlendirmek için format() işleviyle birlikte “{:,}” kullanılmış ve

soldan başlayarak her binlik bölüme bir virgül eklenmektedir (*Python | Add Comma between Numbers - GeeksforGeeks*, 2019).

```
test_num = 1234567

# orijinal numarayı yazdır
print("Orijinal numara: " + str(test_num))

# format() işlevini kullan
# numaralar arasına virgül ekleme
: = Biçim belirteci
d = Bin ayırıcı
res = (format (test_num, ', d'))

# sonuç yazdırma
print("Virgül girdikten sonraki sayı: " + str(res))

# çıktı
Orijinal numara: 1234567
Virgül girdikten sonraki sayı: 1, 234, 567
```

Şekil 12. Format() işlevi ile hata düzeltme (Kaynak: Python | Add Comma between Numbers - GeeksforGeeks, 2019)

2. Yuvarlama Hatası

Programlama sonucunda elde edilen çıktı sonucunda dosya içeriği ile Wikidata içeriğinin örtüşmediği tespit edilmiştir. Tespit edilen hatalar için çözüm önerilmiştir. Çözüm analizi, python kütüphaneleri ile Google içeriğindeki çalışmalar üzerinden yapılmıştır.

Veri seti içerisinde birçok yuvarlama hatası bulunmaktadır. Bu tür veriler ile uğraşmak yanlış sonuçlar elde edilmesine sebep olmaktadır. round(), truncate(), math.ceil(), math.floor(), round_up(), round_down(), round_half_up() ve round_half_down() işlevleri bu hatanın giderilmesini sağlayan tekniklerdir (*How to Round Numbers in Python? - GeeksforGeeks*, 2020). Bir sayıyı verilen basamak sayısına yuvarlayan ve sayıların yuvarlanmasını kolaylaştıran yerleşik işlevler sunarlar.

round() yöntemi iki parametre kullanır: Birinci parametre, yuvarlanacak sayı ve yuvarlarken dikkate alması gereken ondalık basamaklardır. İkinci parametre isteğe bağlıdır ve belirtilmediği sürece varsayılan olarak 0'ı esas alır ve bu durumda en yakın

tam sayıya yuvarlanır ve dönüş türü olarak da bir tam sayıya dönüşür. Ondalık basamaklar, yani ikinci argüman mevcut olduğunda, verilen basamak sayısına yuvarlanacaktır. Dönüş türü bir kayan nokta olacaktır.

round(number, number of digits)

- number: yuvarlanacak sayı
- number of digits: verilen sayının yuvarlanacağı basamak sayısı. İsteğe bağlıdır ve belirtilmemişse varsayılan olarak 0'dır ve yuvarlama en yakın tam sayıya yapılır.

Verilen ondalık basamaktan sonraki sayı ise;

- >=5'ten + 1 son değere eklenecek
- <5, belirtilen ondalık basamaklara kadar olduğu için son değerden geri dönecektir.

```
# tamsayılar için
print(round(11))

# ondalık sayılar için
print(round(22.7))

# Yuvarlandığı son ondalık basamak (ndigit+1)'inci basamak =5
print(round(4.465, 2))

# Yuvarlandığı son ondalık basamak (ndigit+1)'inci basamak >=5
print(round(4.476, 2))

# Yuvarlandığı son ondalık basamak (ndigit+1)'inci basamak <5
print(round(4.473, 2))

#çıktı
11, 23, 4.46, 4.48, 4.47
```

Şekil 13. Round() işlevi ile hata düzeltme (Kaynak: How to Round Numbers in Python? - GeeksforGeeks, 2020)

truncate() işlevi, pozitif ve negatif sayılarla kullanılabilir. Kesme kavramı ile belirli bir konumdan sonraki her rakam 0 ile değiştirilir. Öncelikle; ondalık noktayı, p basamaklarını sağa kaydırmak için sayıyı 10^p ile çarpılır ve int() kullanarak bu yeni sayının tamsayı kısmı alınır. P ondalık basamağının kaydırılması, 10^p'ye bölerek sola geri dönmesini sağlar.

```
 # ikinci bağımsız değişken 0'dır. Sayının tamsayı kısmı döndürülür.
def truncate(n, decimals = 0):
        multiplier = 10 ** decimals
        return int(n * multiplier) / multiplier

print(truncate(16.5))
print(truncate(-3.853, 1))
print(truncate(3.815, 2))

# ondalık noktanın soluna doğru basamakları kesilir
print(truncate(346.8, -1))
print(truncate(-2947.48, -3))

#çıktı
16.0
-3.8
3.81
340.0
-2000.0
```

Şekil 14. Truncate() işlevi ile hata düzeltme (Kaynak: How to Round Numbers in Python? - GeeksforGeeks, 2020)

math.ceil(): Bu işlev, verilen bir sayıdan büyük veya ona eşit olan en yakın tamsayıyı döndürür. **math.floor():** Bu işlev, verilen bir sayıdan küçük veya ona eşit en yakın tamsayıyı döndürür.

```
# pozitif sayı için tavan değeri
# ondalık sayı
print(math.ceil(4.2))

# negatif sayı için tavan değeri
# ondalık sayı
print(math.ceil(-0.5))

# ondalık sayı için taban değeri ve negatif sayı
print(math.floor(2.2))
print(math.floor(-0.5))

#çıktı
5
0
2
-1
```

Şekil 15. Math.ceil() ve Math.floor() işlevleri ile hata düzeltme (Kaynak: How to Round Numbers in Python? - GeeksforGeeks, 2020)

round_up yöntemi ile ilk önce n'deki ondalık nokta, n'yi 10 ** ondalık sayı ile çarparak doğru sayıda sağa kaydırılır. Yeni değer, math.ceil() kullanılarak en yakın tam sayıya yuvarlanır. Son olarak, ondalık nokta 10 ** ondalık basamağa bölünerek sola kaydırılır.

```
# math kütüphanesini tanımlama
import math

# round_up fonksiyonunu tanımlama
def round_up(n, decimals = 0):
        multiplier = 10 ** decimals
        return math.ceil(n * multiplier) / multiplier

# pozitif değelerin girdisi
print(round_up(2.1))
print(round_up(2.23, 1))
print(round_up(2.543, 2))

# negatif değelerin girdisi
print(round_up(22.45, -1))
print(round_up(2352, -2))

#çıktı
3,0 - 2,3- 2,55 - 30,0 - 2400,0
```

Şekil 16. Round_up() işlevi ile hata düzeltme (Kaynak: How to Round Numbers in Python? - GeeksforGeeks, 2020)

Yukarı yuvarlama, sayı doğrusunda her zaman bir sayıyı sağa, aşağı yuvarlama ise sayı satırında her zaman bir sayıyı sola yuvarlar.

round_down() işlevi ile ilk önce n'deki ondalık nokta, n'yi 10 ** ondalık sayı ile çarparak doğru sayıda sağa kaydırılır. Yeni değer, math.floor() kullanılarak en yakın tam sayıya yuvarlanır. Son olarak, ondalık nokta 10 ** ondalık basamağa bölünerek sola kaydırılır.

```
import math

# round_down tanımlama
def round_down(n, decimals=0):
        multiplier = 10 ** decimals
        return math.floor(n * multiplier) / multiplier

# farklı değerlerin girdisi
print(round_down(2.5))
print(round_down(2.48, 1))
print(round_down(-0.5))

#çıktı
2.0, 2.4, -1.0
```

Şekil 17. Round_down() işlevi ile hata düzeltme (Kaynak: How to Round Numbers in Python? - GeeksforGeeks, 2020)

round_half_up() fonksiyonu ile ondalık noktanın istenilen basamak sayısı kadar sağa kaydırılmasıyla uygulanır. Bu durumda kaydırılan ondalık noktadan sonraki basamağın 5'ten küçük mü yoksa büyük mü olduğunun belirlenmesi gerekmektedir. Kaydırılan değere 0,5 eklenip, math.floor() fonksiyonu ile yukarı yuvarlanır.

```
import math

# round_half_up tanımlama
def round_half_up(n, decimals=0):
        multiplier = 10 ** decimals
        return math.floor(n * multiplier + 0.5) / multiplier

# farklı değerlerin girdisi
print(round_half_up(1.28, 1))
print(round_half_up(-1.5))
print(round_half_up(-1.225, 2))

#çıktı
1.3
-1.0
-1.23
```

Şekil 18. Round_half_up() işlevi ile hata düzeltme (Kaynaak: How to Round Numbers in Python? - GeeksforGeeks, 2020)

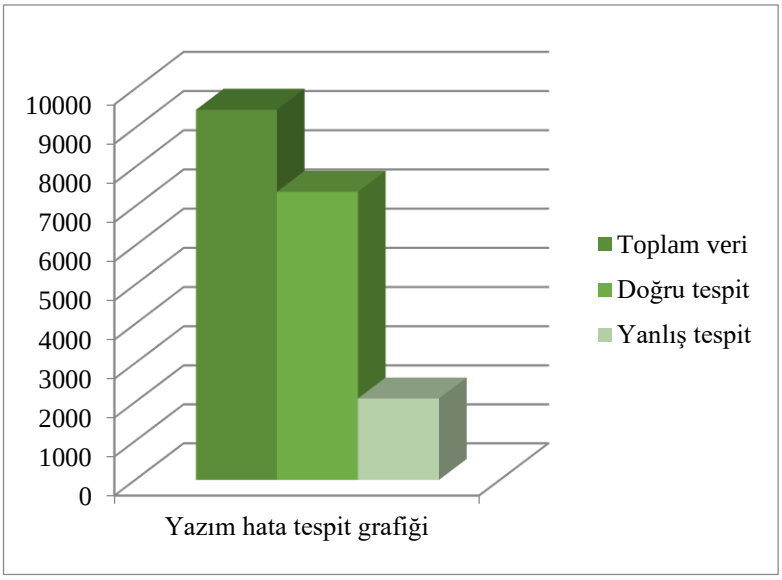
round_half_down() işlevi yarıya yuvarlama yöntemine benzer şekilde en yakın sayıya yuvarlanması, farkı ise iki sayıdan daha küçük olana yuvarlanarak bağları koparmasıdır. Yarım aşağı yuvarlama stratejisi, round_half_up() işlevindeki math.floor() öğesinin math.ceil() ile değiştirilmesi ve ardından eklemek yerine 0,5'in çıkarılmasıyla uygulanır.

```
import math

# round_half_down fonksiyonunu tanımala
def round_half_down(n, decimals=0):
        multiplier = 10 ** decimals
        return math.ceil(n * multiplier - 0.5) / multiplier

# farklı değerlerin girdisi
print(round_half_down(2.5))
print(round_half_down(-2.5))
print(round_half_down(2.25, 1))

#çıktı
2.0
-3.0
2.2
```

Şekil 19. Round_half_down() işlevi ile hata düzeltme (Kaynak: How to Round Numbers in Python? - GeeksforGeeks, 2020)

3.2.2. Yazım Hataları

Analizi yapılan bu tez için 9460 tane farklı veri incelemesi yapılmıştır. Dosya içerisinde sayısal verilerin yanı sıra diğer sütunlarda yazı şeklinde bilgilerde bulunmaktadır. Bu yüzden 9460 tane verinin büyük bir çoğunluğu yazılardan oluşan veri içermektedir. Yapılan işlemler sonucunda 7370 tane doğru tespit ,2090 tane ise hata tespit edilmiştir.



Şekil 20. Yazım hata tespit grafiği

Sayısal veri analizi üç farklı alan da incelenmiştir.

1. Özel Karakter Hatası
2. Boşluk Hatası
3. Harf Hatası

Tespit edilen bu hatalı alanlar için çözüm önerileri getirilmiştir.

1. Özel Karakter Hatası

Dosyalarda ki veriler içerisine ‘*’,’#’,’:’,’-‘ ,’!’, ‘â’, ‘€’, ‘“’, ‘|’, ‘Ã’, ‘«’, ‘‰’, ‘©’, ‘Å’, ‘¶’, ‘¥’, ‘€™’, ‘Ä’, ‘Æ’, ‘€’, ‘º’, ‘¾’, ‘½’, ‘^’ , ‘™’, ‘³’, ‘1/2’, ‘Â’, ‘§’, ‘¼’, ‘¤’,’³’,’‡’, ‘-‘, ‘Ã³’, ‘±’ bunlar ve bunlar gibi özel karakterler karıştırılmıştır. Bazı veriler Google üzerinden dahi aratılıp bulunamayacak kadar belirsiz hale getirilmiştir. Tespit edilen hatalar için çözüm önerilmiştir. Replace(), join(), translate() ve filter() metotları bu sorunun giderilmesinde fayda sağlamaktadır (*Python | Removing Unwanted Characters from String - GeeksforGeeks*, 2020).

Döngü içerisinde özel karakter olup olmadığını kontrol ettikten daha sonra onu boş bir dizeyle değiştirmek için **replace()** yöntemi kullanılır.

```
# bad_chars listesi
bad_chars = [';', ':', '!', "*"]

# test için bozulmuş örnek bir dize
test_string = "Ge;ek * s:fo ! r;Ge * e*k:s !"

# orijinal dize
print ("Orijinal dize : " + test_string)

# replace() yöntemini uygulayarak istenmeyen karakterleri kaldırma
for i in bad_chars :
        test_string = test_string.replace(i, '')

# sonuç dizesini yazdırma
print ("Sonuç listesi : " + str(test_string))

# çıktı
Orijinal dize : Ge;ek*s:fo!r;Ge*e*k:s!
Sonuç listesi : GeeksforGeeks
```

Şekil 21. Replace() işlevi ile hata düzeltme (Kaynak: Python | Removing Unwanted Characters from String - GeeksforGeeks, 2020)

Join() tekniği ile dizenin her öğesi, bir listenin eşdeğer bir öğesine dönüştürülür, ardından kaldırılacak belirli karakterler hariç her biri bir dize oluşturmak üzere birleştirilir.

```
# bad_chars listesi
bad_chars = [';', ':', '!', "*"]

# test için bozulmuş örnek bir dize
test_string = "Ge;ek * s:fo ! r;Ge * e*k:s !"

# orijinal dize
print ("Orijinal dize : " + test_string)

# join() yöntemini uygulayarak istenmeyen karakterleri kaldırma
test_string = ''.join(i for i in test_string if not i in bad_chars)

# sonuç dizesini yazdırma
print ("Sonuç listesi : " + str(test_string))

# çıktı
Orijinal dize : Ge;ek*s:for!r;Ge*e*k:s!
Sonuç listesi: GeeksforGeeks
```

Şekil 22. Join() işlevi ile hata düzeltme (Kaynak: Python | Removing Unwanted Characters from String - GeeksforGeeks, 2020)

translate() yöntemini kullanarak bir dizeden birden çok karakteri kaldırılabilmektedir. Bunun gerçekleşmesi için, bir dizeden çıkarılmak istenen karakterlerin listesi yazdırılır. Bu karakterlerden biri bulunursa, boşluk ile değiştirilir. Bu yöntemi kullanmak için önce maketrans() yöntemini kullanarak bir tablo oluşturulmalıdır. Buradaki tablo, hangi karakterin değiştirilmesi gerektiğini tanımlar.

```
# bad_chars (istenmeyen karakterler) listesi
bad_chars = [';', ':', '!', "*"]

# test için bozulmuş örnek dize
test_string = "Ge;ek * s:fo ! r;Ge * e*k:s !"

# orijinal dize
print ("Orijinal dize : " + test_string)

# translate() yöntemi ile istenmeyen karakterleri kaldırma
delete_dict = {sp_character: '' for sp_character in string.punctuation}
delete_dict[' '] = ''
table = str.maketrans(delete_dict)
test_string = test_string.translate(table)

# sonuç dizesini yazdırma
print ("Sonuç listesi : " + str(test_string))

# çıktı
Orijinal dize : Ge;ek*s:fo!r;Ge*e*k:s!
Sonuç listesi : GeeksforGeeks
```

Şekil 23. Translate() işlevi ile hata düzeltme (Kaynak: Python | Removing Unwanted Characters from String - GeeksforGeeks, 2020)

filter() yöntemi dizedeki tüm alfabeleri içeren bir yineleyici döndürür ve join(), yineleyicideki tüm öğeleri boş bir dizeyle birleştirir.

```
# istenmeyen karakterler listesi
bad_chars = [';', ':', '!', "*"]

# içeriği bozulmuş örnek bir dize
test_string = "Ge;ek*s:fo!r;Ge*e*k:s!"

# orijinal dize yazdırma
print("Orijinal dize : " + test_string)

# filter() işlevi uygulanarak istenmeyen karakterlerin kaldırılması
test_string = ''.join((filter(lambda i: i not in bad_chars, test_string)))

# sonuç listesini yazdırma
print("Sonuç listesi : " + str(test_string))

# çıktı
Orijinal dize : Ge;ek*s:for!r;Ge*e*k:s!
Sonuç listesi: GeeksforGeeks
```

Şekil 24. Filter() işlevi ile hata düzeltme (Python | Removing Unwanted Characters from String - GeeksforGeeks, 2020)

2. Boşluk Hatası

Yapılan analiz sonucunda tespit edilen hatalar diğer hata türlerine göre daha az düzeydedir. Ortalama hata sayısı 10 ile 20 arasındadır. Hata tespiti Google üzerinden teyit edilmiş doğru tespitler gözlemlenmiştir ve çözüm önerileri araştırılmıştır.

Ham veriler genellikle düzgün biçimlendirilmemiştir ve metin içinde çift boş karakterlerin yanı sıra dizelerin başında ve sonunda çok sayıda gereksiz boşluk içerir. Replace(), split() – join(), re.sub() ve translate() işlevleri hata tespitleri giderilebilmektedir (*Python | Remove Spaces from a String - GeeksforGeeks*, 2019).

replace( ) işlevi , dizedeki tüm boşluklar ile birlikte kelimeler arasındaki boşlukları da kaldırır.

```
def remove(string):
        return string.replace(" ", "")

string = ' g e e k '
print(remove(string))

# çıktı
geek
```

Şekil 25. Replace() işlevi ile boşluk hatası düzeltme (Kaynak: Python | Remove Spaces from a String - GeeksforGeeks, 2019)

Join() ve **split()** işlevi tamamen uyumlu olarak çalışır. İlk olarak, **split()** yöntemi, bir sınırlayıcı kullanarak tüm dizedeki sözcüklerin bir listesini döndürür. Sonra onları birleştirmek için **join()** yönteminin kullanılması gerekmektedir.

```
def remove(string):
        return "".join(string.split())

string = ' g e e k '
print(remove(string))

# çıktı
geek
```

Şekil 26. Join() ve split() işlevleri ile boşluk hatası düzeltme (Kaynak: Python | Remove Spaces from a String - GeeksforGeeks, 2019)

translate() işlevi dizedeki tüm boşlukları kaldırır ve çıktı olarak kompakt bir dize elde edilir.

```
import string

def remove(string):
        return string.translate(None, ' \n\t\r')

string = ' g e e k '
print(remove(string))

Output
geek
```

Şekil 27. Translate() işlevi ile boşluk hatası düzeltme (Kaynak: Python | Remove Spaces from a String - GeeksforGeeks, 2019)

re.sub() işlevi, bir veya daha fazla eşleşmeyi bir yedek dizeyle değiştirir. İşlevde "//s+", herhangi bir sayıda boşlukla eşleşmesi için normal bir ifade olarak iletilir. Bu eşleşmelerin yerine, baştaki ve sondaki boşlukları kaldırırken " " ve kelimeler arasındaki boşlukları kaldırırken tek bir boşluk (" ") iletilebilir.

```
import re

def remove(string):
        pattern = re.compile(r'\s+')
        return re.sub(pattern, '', string)

string = ' g e e k '
print(remove(string))

# çıktı
geek
```

Şekil 28. Re.sub() işlevi ile boşluk hatası düzeltme (Kaynak: Python | Remove Spaces from a String - GeeksforGeeks, 2019)

3. Harf Hatası

Çalışılan bu projenin büyük bir kısmını oluşturan yazım hataları çoğunlukla harflerden kaynaklı tespitler belirlenmiştir. Bu hata tespiti iki farklı başlık altında incelenmiştir.

- Eksik – Fazla Harf Hatası
- Yanlış Harf Hatası

Yanlış harf kullanımı sebebi ile birçok veri analizi doğru cevaplar ile sonuçlanamamaktadır. Tüm bu tespitleri kapsayacak şekilde çözüm önerileri araştırılmıştır. Gerek python kütüphaneleri gerekse Google üzerinden çalışmalar

gözlemlenmiş, hata çözüm noktaları belirlenmiştir. TextBlob ve pyspellchecker kütüphaneleri en çok kullanılan başlıca yöntemlerdir (*Spelling Checker in Python - GeeksforGeeks,* 2020) .

Python programlama dilindeki **TextBlob** , metinsel verileri işlemek için bir python kütüphanesidir. İsim öbeği çıkarma, duygu analizi, sınıflandırma, çeviri ve daha fazlası yaygın doğal dil işleme görevlerini yerine getirmek için bir API (Application Programming Interface) sağlar. **Correct()** işlevi yazım düzeltmesinin denemek için kullanılır.

```
from textblob import TextBlob

# yazım yanlışı bulunan metin
a = "eies"

print("orijinal metin: "+str(a))

b = TextBlob(a)

# doğru yazım çıktısı
print("doğru metin: "+str(b.correct()))

# çıktı
orijinal metin: eies
doğru metin: eyes
```

Şekil 29. TextBlob kütüphanesi ile hata düzeltme (Kaynak: Spelling Checker in Python - GeeksforGeeks, 2020)

Yazı denetimi, herhangi bir metin işleme veya analizinde temel bir gerekliliktir. Python paketi olan **pyspellchecker** , yanlış yazılmış olabilecek kelimelerin bulunmasını ve olası düzeltmelerin önerilmesinde kolaylık sağlar. Yanlış harf yazımına olduğu kadar büyük-küçük harf hatasına karşıda duyarlıdır. Çözüm önerileri sonucu, olası doğru tüm sonuçları verir.

```
from spellchecker import SpellChecker
spell = SpellChecker()

# yazım yanlışı olabilecek kelimeler
misspelled = spell.unknown(["cmputr", "watr", "study", "wrte"])

for word in misspelled:
        # doğru metne en yakın cevap
        print(spell.correction(word))

        # olası cevapları listeleye alma
        print(spell.candidates(word))

# çıktı
computer
{'caput', 'caputs', 'compute', 'computor', 'impute', 'computer'}
water
{'water', 'watt', 'warr', 'wart', 'war', 'wath', 'wat'}
write
{'wroe', 'arte', 'wre', 'rte', 'wrote', 'write'}
```

Şekil 30. Pyspellchecker kütüphanesi ile hata düzeltme (Kaynak: Spelling Checker in Python - GeeksforGeeks, 2020)

Etkileşimli yazım denetleyicisi, her bir hata için birden fazla alternatif düzeltme önerebilir ve kullanıcı değiştirme önerisinden birini seçer ve otomatik düzeltmede, yazım denetleyicisi en iyi düzeltmeye karar verir ve seçer ve hata otomatik olarak yanlış yazılmış sözcükle değiştirilir.Sözcük dışı yazım hatası düzeltmesi, bir metinde yanlış yazılmış sözcükleri tespit etme ve önerilerde bulunma işlemidir(Melaku, 2017). İmla denetleyicisi, her hata için bir veya daha fazla düzeltme önerebilir ve kullanıcı listeden en iyi kelimeyi seçer ve yanlış yazılmış kelimeyi değiştirir.

Yazım denetimi, yazım denetleyicisi tarafından bildirilen sözcükleri ifade eder ve karmaşık sistemler de hata düzeltme sonuçlarını da iyileştirebilir. Her yanlış yazılmış kelimeden iki karakterlik bir düzenleme mesafesi içindeki tüm olası permütasyonları bulmak için bir Levenshtein mesafesi algoritması uygulayan bir python modülü olan pyspellchecker, sıklıkla hata denetimi için kullanılır (Gagliardi et al., 2020). Daha sonra tüm permütasyonları, karakter eklemeleri, silmeleri, değiştirmeleri ve yer değiştirmeleri bir kelime sıklık listesindeki bilinen kelimelerle karşılaştırır.

Duygu analizi, sosyal medya alanındaki gözle görülür bir büyüme oranı nedeniyle teknolojik dünyada çok önemli bir rol oynamaktadır. Duygu analizine dayalı çalışma için ortak bir zorluk, yanlış yazılan kelimelerdir. Bir metin mesajında kullanılan kısaltmalar biçimindeki argolar ve kısa kelimeler için duygu puanını test edememesi de zorluklardan biridir. Metinsel verilerin ön işlenmesi, modele girdi olarak verilen metinsel verilerin boyutunu küçülttüğü için duygu analizinde çok önemlidir. Metinsel verilerin ön işlemesi yapılırken çeşitli adımlar izlenir. Bu tür çalışmalar sayesinde textblob kütüphanesinin kullanımının anlaşılması da sağlamaktadır.

Metinsel verileri temizlemek için gerçekleştirilen çeşitli ön işleme görevleri, cümlelerin sınırlarının belirlenmesi, doğal dilden geçen kelimelerin çıkarılması, kökten türetme ve simgeleştirmedir. Tokenizasyon, bir veri setini önceden işlerken en önemli adımlardan biridir. Metinsel verilerin küçük belirteçlere bölüneceği şekilde çalışır. Her simge, metin belgesinden veya dilden bir kelimeyi temsil eder. TextBlob'un tokenize edilmiş kelimeleri tokenize etmek ve okumak için önemli ölçüde iyi performans gösterdiği gözlemlenmiştir (Shekhawat, 2019).

Metin bloğuna dayalı TELKOM ürün ve hizmetleri, Twitter'dan elde edilen verileri kullanarak verileri kelime bulutunda görselleştirir, böylece sınıflandırılabilir ve görselleştirilebilir.Daha sonra sınıflandırma, olumlu görüşler, tarafsız görüşler ve olumsuz görüşler hakkında tweet'ler olmak üzere üç kategoriye ayrılan bu twitter duygu analizinde yapılır. Tweet sınıflandırması, textblob adı verilen python kütüphanelerinden biri kullanılarak yapılır. Textblob, önceden sınıflandırılmış incelemelere sahip bir eğitim setine sahiptir ve özel bir sınıflandırıcı oluşturmak için yerleşik sınıflandırıcılar modülü sağlar (Gagliardi et al., 2020). Daha sonra, bu analizin sonucu, TELKOM hakkındaki twitter görüşlerinin nasıl olduğunu görmek için değerlendirme ve karşılaştırma olarak kullanılabilir.

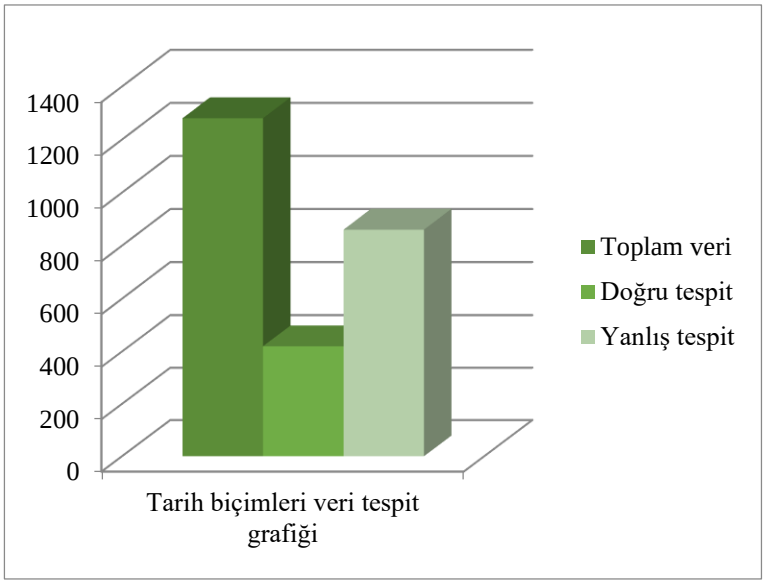
Tweet herkes tarafından görülebilir, ancak gönderen, kimlerin görebileceğini yalnızca arkadaşlarıyla sınırlayabilir. Kullanıcı ayrıca takipçi olarak bilinen başka bir kişinin tweetini de görebilir. TextBlob, metinsel verileri işlemek için bir python kütüphanesidir. Bu özellik, konuşma parçası işaretleme, isim öbeği çıkarma, duygu analizi, sınıflandırma, çeviri ve daha fazlası gibi genel Doğal Dil İşleme (NLP) görevlerine katkı sağlamak için uygulanabilir arayüz sağlar.

Düzenli veri alımı gerçekleştirmek, günde 2-3 kez veri alımı olduğu anlamına gelir ve bu daha sonra tek bir veri olarak toplanır, derlenir ve veriler toplandıktan sonra bir sınıflandırma süreci başlar (Mas Diyasa et al., 2021). Textblob'u kullanarak bu adımın amacı, pozitif, negatif ve nötr verileri ayırt etmektir. Bundan sonra, ön işleme adımı tweet verilerini temizlemeye başlar. Ön işleme metinleri, özel karakterlerin kaldırılmasını, kelimelerin standartlaştırılması, eklerin ve son eklerin kaldırılması, cümlelerdeki kelime zincirlerini ayırma ve büyük/küçük harfe dönüştürmeyi içerir.

Sinir ağları, metin sınıflandırması için sıklıkla kullanılır, ancak rakip örneklerin neden olduğu yanlış sınıflandırmaya karşı savunmasız olabilir: küçük bozulmalar ekleyerek üretilen girdi, sinir ağının yanlış bir sınıflandırma vermesine neden olur. Yazım denetimi kullanılarak bu tür hatalı durumların ortadan kaldırılması sağlanır. Derlenen metin içerikleri için ilk önce sayılar, özel semboller, tarihler, e-posta adresleri ve köprüler için filtrelenir. Ardından, sözlük verilerini kullanmadan önce olası yazım hatalarını düzeltmek için textblob otomatik düzelticisini üzerlerinde çalıştırılır. Ayrıca "would've" gibi ingilizce kısaltmaları olan belirteçlerin yazım hatası olarak algılanmaması için sözlüğe ingilizce kısaltmalar eklenir ve Wikipedia'dan tüm ingilizce kısaltmaların bir listesini alınarak, inceleme sürecin sekteye uğraması engellenir (Alshemali & Kalita, 2019). TextBlob, yazım düzeltmelerini en yüksek olasılıkla bulan olasılıksal bir araçtır.

3.2.3. Tarih Biçim Hataları

Analizi yapılan bu tez için 9460 tane farklı veri incelemesi yapılmıştır. İçerisinde toplam 1280 tane farklı tarihsel veri bulunmaktadır. Yapılan çalışma sonucunda 420 tane doğru tespit, 860 tane ise hata tespit edilmiştir.



Şekil 31. Tarih biçimleri veri tespit grafiği

Hata tespitleri içerisinde en dikkat çeken detay; eğer yıl ve ay 01 ise Wikidata içerisinde sadece yıl olarak yazmaktadır. Fakat bu ayrıntı ile ilgili herhangi bir çalışma denk gelinemedi.

Wikidata Query Service içerisinde bulunan bir örnek çalışma ile tarih verisinin (ör. "1830-01-01T00:00:00Z"^^xsd:dateTime) doğru yazılımı verilmiştir (Wikidata, 2020). Veri seti içerisinde bu doğru yazılım şekli ile bulunan veriler vardır fakat yine de sonuç alınamamıştır.

```
SELECT ?item ?label ?coord ?place
WHERE
{
  VALUES ?type {wd:Q571 wd:Q7725634}
  ?item wdt:P31 ?type .
  ?item wdt:P577 ?date FILTER (?date < "1830-01-
01T00:00:00Z"^^xsd:dateTime) .
  ?item rdfs:label ?label filter (lang(?label) = "en")

  OPTIONAL {
    ?item (wdt:P291|wdt:P840) ?place .
    ?place wdt:P625 ?coord
  }
}
```

Şekil 32. Wikidata sorgu hizmeti - SPARQL (Kaynak: Wikidata, 2020)

4. SONUÇ

Web' de gün geçtikçe artan veri seti sayısı, hem yüksek veri kullanılabilirliği için fırsatlar hem de anlamsal olarak heterojen ve dağıtılmış bir ortamda veri sorgulamanın doğasında bulunan zorlukları beraberinde getirmektedir. Web altyapısı ;URI' ler, HTTP, semantik web standartları RDF - RDFs ve sözlükler üzerine inşa edilen bağlantılı veriler, veri yayınlama, tüketim ve yeniden kullanımın önündeki engelleri etkin bir şekilde azaltabilmektedir.

Bu çalışma, belgelerin anlamsal yönetimi ve web uygulamasının geliştirilmesi için World Wide Web Konsorsiyumu tarafından önerilen Semantik Web ve Bağlantılı Veri ilkelerinin kullanımını ele almaktadır. Bu ilkelerin amacı, taranan belgeleri, makinenin anlayabileceği ve işleyebileceği şekilde kaydetmek, içeriği filtrelemek ve karar verme süreci devam ederken bu tür belgeleri aramamıza yardımcı olmaktır. Bu amaçla, referans bağlantılı veri ontolojileri kullanılarak oluşturulan makine tarafından anlaşılabilir meta veriler, bir bilgi tabanı oluşturarak belgelerle ilişkilendirilir.

Semantik web, HTTP gibi yerleşik web teknolojilerinin bir kombinasyonunu kullanarak hem veri hem de web içeriği üzerinde açıklama, sorgulama ve akıl yürütme aracı sağlamaktadır. Bu sayede, veri entegrasyonu, verilerin yeniden kullanılması, veri toplama-sınıflandırma ve kalite kontrol edilmesi, veri yayınlama ve keşfedilmesi durumların, otomatikleştirmeye yardımcı olan yöntemlerin geliştirilmesini amaçlar. Semantik web' in ayrılmaz ve önemli bir parçası olan bağlantılı veri, yapılandırılmış verileri webde RDF veri tabanını kullanarak yayınlamanın ve bunları birbirine bağlamanın bir yolunu ifade eder.

Bağlantılı veri ağı, farklı veri kümeleri içinde yer alan ve benzersiz veri yayıncıları tarafından sağlanan geniş, dağıtılmış ve birbirine bağlı bir bilgi parçaları ağı sağlar. Anlamsal bağlantılar boyunca bir veri kümesinden diğerine hareket edebilmek için farklı veri kümeler arasında bağlantılar eklemeyi önerir. Bu daha sonra her veri kümesini tüm bağlantılı veri web' inin bağlı bir alt parçası olarak konumlandırır ve anlamsal arama motorları gibi uygulamaların bu bağlantılar boyunca gezinmesini ve farklı veri kümeleri arasında veri keşfetmesini sağlar.

SemTab adı altında düzenlenen yarışma içerisinde bağlantılı tablo verileri mevcuttur. Wikidata üzerinden SPARQL sorgu dili ile sorguladığımız bu veriler yarışmaya özgü değişikliğe uğramıştır. Yarışmanın amacı, yazımları bozulmuş bu tablo verilerinin programa yolu ile düzelmesini sağlamaktır. Bağlantılı olmaları sayesinde arama motorları içerisinde doğruları tespit edilerek sorunlara çözüm önerilmiştir. Birçok yarışmacının katıldığı bu platformda sorgu dilini bilmeden ilerlemek çok güçtür. Hataların giderilmesi için üretilen kodlama ile dahi bulunamayan bozulmuş veriler bulunmaktadır. Bu yüzden, en yüksek oranda doğru veriye ulaşmak çok mühimdir.

5. TABLOLAR

Tablo 2. Sayısal Hatalar (Noktalama)

Dosya Adı	Yanlış Tespit	Doğru Tespit	Kaynak
W9Q07IIX	68670.0	68,125	https://www.wikidata.org/wiki/Q16153781
W9Q07IIX	82205.082	82,041	https://www.wikidata.org/wiki/Q16153737
YMOQDGO8	7200.688	7,172	https://www.wikidata.org/wiki/Q647378
YMOQDGO8	9178.873	9,097	https://www.wikidata.org/wiki/Q131293
YMOQDGO8	3456.633	3,481	https://www.wikidata.org/wiki/Q219956
YMOQDGO8	3785.83	3,767±1	https://www.wikidata.org/wiki/Q281206
YMOQDGO8	9489.21	9,442	https://www.wikidata.org/wiki/Q83077
Y3P0JZSE	3693.003	3,696.7	https://www.wikidata.org/wiki/Q4508056
Y3P0JZSE	126000.2	127,273	https://www.wikidata.org/wiki/Q4508056
Y3P0JZSE	64665.41	64,795	https://www.wikidata.org/wiki/Q4492782
YLAFQZJL	144586.2	144,876	https://www.wikidata.org/wiki/Q697126
YLD9Y1VG	2377.8	2,359	https://www.wikidata.org/wiki/Q501800
YLD9Y1VG	191083.7	191,27	https://www.wikidata.org/wiki/Q501800
YLD9Y1VG	2431.6	2,439	https://www.wikidata.org/wiki/Q501312
YLD9Y1VG	8712.3	8,719	https://www.wikidata.org/wiki/Q501312
XF412HIL	7452	7,452	https://www.wikidata.org/wiki/Q2418286
XF412HIL	9947	9,947	https://www.wikidata.org/wiki/Q2325412
VQH95AU7	4982.9	4,922	https://www.wikidata.org/wiki/Q865682
VQH95AU7	6505.8	6,532	https://www.wikidata.org/wiki/Q371429
YBWDI6SL	92437.49	92,902	https://www.wikidata.org/wiki/Q509653

Tablo 3. Sayısal Hatalar (Yuvarlama)

Dosya Adı	Yanlış Tespit	Doğru Tespit	Kaynak
W9Q07IIX	36889.0	37,000	https://www.wikidata.org/wiki/Q16152313
VTZSYCZH	247.5	250	https://www.wikidata.org/wiki/Q26251437
VTZSYCZH	49.85	50	https://www.wikidata.org/wiki/Q26251385
VTZSYCZH	70.63	70	https://www.wikidata.org/wiki/Q26251385
VTZSYCZH	929.2	920	https://www.wikidata.org/wiki/Q25933055
VTZSYCZH	148.5	150	https://www.wikidata.org/wiki/Q3817141
VTZSYCZH	963	965.8	https://www.wikidata.org/wiki/Q3817141
VTZSYCZH	399.96	404	https://www.wikidata.org/wiki/Q3735894
VTZSYCZH	69.3	70	https://www.wikidata.org/wiki/Q3735894
Y3P0JZSE	3042.05	3,063.5	https://www.wikidata.org/wiki/Q4492782
XJPS769W	629.01	623.4	https://www.wikidata.org/wiki/Q2410839
XJPS769W	0.739	0.74	https://www.wikidata.org/wiki/Q2410839
YLAFQZJL	715.57	711.3	https://www.wikidata.org/wiki/Q42617191
YLAFQZJL	70100.9	69,96	https://www.wikidata.org/wiki/Q42617191
YLAFQZJL	249.72	250.73	https://www.wikidata.org/wiki/Q697126
VQH95AU7	361.9	360.5	https://www.wikidata.org/wiki/Q865682
VQH95AU7	47.3	46.92	https://www.wikidata.org/wiki/Q371429
Z12ZUOQQ	60.31	59.8967± 5.127	https://www.wikidata.org/wiki/Q6848312
Y95CKWKP	194.86	193.7	https://www.wikidata.org/wiki/Q56850391
Y95CKWKP	203.8	202	https://www.wikidata.org/wiki/Q3895871

Tablo 4. Sayısal Hatalar (Negatif sayı)

Dosya Adı	Yanlış Tespit	Doğru Tespit	Kaynak
XCFMZMB9	-3.097	−3.097±2.236	https://www.wikidata.org/wiki/Q83762445
XCFMZMB9	-1.279	−1.279±3.061	https://www.wikidata.org/wiki/Q80668725

Tablo 5. Tarih Biçim Hataları

Dosya Adı	Yanlış Tespit	Doğru Tespit	Kaynak
YI3OWYLW	1976/01/01	1976	https://www.wikidata.org/wiki/Q9055893
VL88K9QO	1638/01/01	1638	https://www.wikidata.org/wiki/Q574773
W0SYNFW7	1327/01/01	1327	https://www.wikidata.org/wiki/Q4453239
WV1SQ7GX	1961-06-21	21 June 1961	https://www.wikidata.org/wiki/Q17992180
W0SYNFW7	1980/04/30	30 April 1980	https://www.wikidata.org/wiki/Q2225408
VTBT947T	2006/01/04	4 January 2006	https://www.wikidata.org/wiki/Q1104509
VNMESCIH	1944/01/01	1944	https://www.wikidata.org/wiki/Q20105955
VNMESCIH	1965/10/10	10 October 1965	https://www.wikidata.org/wiki/Q41754946

Tablo 6. Yazım Hataları (Özel Karakter)

Dosya Adı	Yanlış Tespit	Doğru Tespit	Kaynak
XSOKYG2I	TarvaspÃ¤Ã¤	Tarvaspää	https://www.wikidata.org/wiki/Q55595461
XSOKYG2I	HvittrÃ¤sk	Hvitträsk	https://www.wikidata.org/wiki/Q55595454
WQ6XCI5Q	MaceiÃ³	MaceiÓ Shopping	https://www.wikidata.org/wiki/Q83653343
YU6HYQ9G	Soyu| 16	Soyuz 16	https://www.wikidata.org/wiki/Q847013
XEM42DZZ	BÃ© d'interÃ¨s	Bé d'interès	https://www.wikidata.org/wiki/Q5004973
VY8TXMDY	MSC"Sasha	MSC Sasha	https://www.wikidata.org/wiki/Q83638468
VL88K9QO	Bouaniska trÃ¤dgÃ¥rden	Botaniska trädgården	https://www.wikidata.org/wiki/Q2755982
W0SYNFW7	Ilindenâ€"Preobrazenie Uprising	Ilinden–Preobrazhenie Uprising	https://www.wikidata.org/wiki/Q1145682
VRSWJQXP	TrÃªs Grutas	Três Grutas	https://www.wikidata.org/wiki/Q10292211
Z63VSBY4	GÃ¶ttingen Manifeso	Göttingen Manifesto	https://www.wikidata.org/wiki/Q11767087
YMOQDGO8	Mardin#Province	Mardin Province	https://www.wikidata.org/wiki/Q131293
YMOQDGO8	ElazÄ±ÄŸ Psoince	Elazığ Province	https://www.wikidata.org/wiki/Q483091
YMOQDGO8	AdÄ±yaman Prqvince	Adıyaman Province	https://www.wikidata.org/wiki/Q43924
YMOQDGO8	Å□anlÄ±urfa rovince	Şanlıurfa Province	https://www.wikidata.org/wiki/Q388469

YMOQDGO8	Å□Ä±rak Province	Şırnak Province	https://www.wikidata.org/wiki/Q647378
VTZSYCZH	Sanygast	Sandygast	https://www.wikidata.org/wiki/Q26251385
XANKG4CC	Copa IbÃ©rica de Rugby	Copa Ibérica de Rugby	https://www.wikidata.org/wiki/Q3325068
XANKG4CC	CampeÃ³n de Campeones	Campeón de Campeones	https://www.wikidata.org/wiki/Q1031000
YLD9Y1VG	Lewis Count|	Lewis County	https://www.wikidata.org/wiki/Q495147
XF412HIL	Gmina PrzemyÅ› l	Gmina Przemyśl	https://www.wikidata.org/wiki/Q2325412
Y95CKWKP	2000 Parisâ€•Bourges	2000 Paris–Bourges	https://www.wikidata.org/wiki/Q3895871
VNMESCIH	1965 Paris/Tours	1965 Paris-Tours	https://www.wikidata.org/wiki/Q41754946

Tablo 7. Yazım Hataları (Boşluk)

Dosya Adı	Yanlış Tespit	Doğru Tespit	Kaynak
XRK6MM1W	Gliese809	Gliese 809	https://www.wikidata.org/wiki/Q16069386
W0SYNFW7	Siege ofGenoa	Siege of Genoa	https://www.wikidata.org/wiki/Q7510044
Z63VSBY4	Force11 Manifesto	Force 11 Manifesto	https://www.wikidata.org/wiki/Q33115579
YMOQDGO8	OsmaniyeProvince	Osmaniye Province	https://www.wikidata.org/wiki/Q281206
VNMESCIH	1944Paris-Tours	1944 Paris-Tours	https://www.wikidata.org/wiki/Q20105955
XI7WTP87	TokyoManagement College	Tokyo Management College	https://www.wikidata.org/wiki/Q11525089
W6WQBQ1E	Albumof paintings	Album of paintings	https://www.wikidata.org/wiki/Q50818784
WS8B976S	Institute ofBiophysics	Institute of Biophysics	https://www.wikidata.org/wiki/Q30281882
WKRLFRC3	Oton Municipal Hall	Oton Municipal Hall	https://www.wikidata.org/wiki/Q54395349
XLKJCZM8	1992World	1992 World	https://www.wikidata.org/wiki/Q50284589
VN5K69AK	USSIntrepid	USS Intrepid	https://www.wikidata.org/wiki/Q1351288

Tablo 8. Yazım Hataları (Eksik - Fazla Harf)

Dosya Adı	Yanlış Tespit	Doğru Tespit	Kaynak
XRK6MM1W	Glese 793	Gliese 793	https://www.wikidata.org/wiki/Q5880946
VY8TXMDY	Msc Barelona	Msc Barcelona	https://www.wikidata.org/wiki/Q52379633
YI3OWYLW	Fenpfuhl park	Fennpfuhl park	https://www.wikidata.org/wiki/Q1404794
W9Q07IIX	Afro-incentian	Afro-vincentian	https://www.wikidata.org/wiki/Q16153781
W0SYNFW7	Acclmation	Acclamation	https://www.wikidata.org/wiki/Q3406431
VTBT947T	Dnzig	Danzig	https://www.wikidata.org/wiki/Q1104509
VTBT947T	Brgia	Borgia	https://www.wikidata.org/wiki/Q893548
Primari Guglielmo Marconi	Primari Guglielmo Marconi	Primaria Guglielmo Marconi	https://www.wikidata.org/wiki/Q52787965
YMOQDGO8	Zonguldа Province	Zonguldak Province	https://www.wikidata.org/wiki/Q647378
YMOQDGO8	Kocaeli Povince	Kocaeli Province	https://www.wikidata.org/wiki/Q83965
VTZSYCZH	Palosand	Palossand	https://www.wikidata.org/wiki/Q26251437
VTZSYCZH	Sanygast	Sandygast	https://www.wikidata.org/wiki/Q26251385
XJPS769W	Priktnica	Praktica	https://www.wikidata.org/wiki/Q632365
YLAFQZJL	Bridgend ounty Borough	Bridgend county Borough	https://www.wikidata.org/wiki/Q697126
YLD9Y1VG	Kanawa County	Kanawha County	https://www.wikidata.org/wiki/Q501800

Z12ZUOQQ	Mu Oionis	Mu Orionis	https://www.wikidata.org/wiki/Q6848312
Y95CKWKP	2018 Parsâ€“Bourges	2018 Paris–Bourges	https://www.wikidata.org/wiki/Q56850391
VNMESCIH	1963 Paris-ours	1963 Paris-Tours	https://www.wikidata.org/wiki/Q41754931

Tablo 9. Yazım Hataları (Yanlış Harf)

Dosya Adı	Yanlış Tespit	Doğru Tespit	Kaynak
YU6HYQ9G	Soywz 14	Soyuz 14	https://www.wikidata.org/wiki/Q545595
W9Q07IIX	Afto-Bermudian	Afro-Bermudian	https://www.wikidata.org/wiki/Q16152313
W9Q07IIX	Khanty seople	Khanty people	https://www.wikidata.org/wiki/Q476030
W9Q07IIX	Afro-Antiguan and Barbvdan	Afro-Antiguan and Barbudan	https://www.wikidata.org/wiki/Q16153737
YI3OWYLW	Parque de La Granka	Parque de La Granja	https://www.wikidata.org/wiki/Q9055893
VL88K9QO	Lardins de Joan Maragall	Jardins de Joan Maragall	https://www.wikidata.org/wiki/Q579370
VL88K9QO	Herrenhausen Iardens	Herrenhausen Gardens	https://www.wikidata.org/wiki/Q574773
W0SYNFW7	Tver Uprisiog of 1327	Tver Uprising of 1327	https://www.wikidata.org/wiki/Q4453239
W0SYNFW7	Amsterdbm coronation riots	Amsterdam coronation riots	https://www.wikidata.org/wiki/Q2225408
YBG6SOTJ	Pbrliament of Scotland	Parliament of Scotland	https://www.wikidata.org/wiki/Q1650523
VTBT947T	Lammtasra	Lammtarra	https://www.wikidata.org/wiki/Q1012807

YMOQDGO8	Kars Provjnce	Kars Province	https://www.wikidata.org/wiki/Q83077
VTZSYCZH	Pudsdale	Mudsdale	https://www.wikidata.org/wiki/Q25933055
VTZSYCZH	Krookodi me	Krookodile	https://www.wikidata.org/wiki/Q3817141
VTZSYCZH	Hxcadrill	Excadrill	https://www.wikidata.org/wiki/Q3735894
Y3P0JZSE	Cieboksa ry uyezd	Cheboksar y uyezd	https://www.wikidata.org/wiki/Q4508056
Y3P0JZSE	Friedricjs tadt County	Friedrichst adt County	https://www.wikidata.org/wiki/Q4492782
YLAFQZJL	Isle of Angoese y	Isle of Anglesey	https://www.wikidata.org/wiki/Q42617191
YLAFQZJL	Flinvshir e	Flintshire	https://www.wikidata.org/wiki/Q505610
YLD9Y1VG	Pocahont as Counvy	Pocahontas County	https://www.wikidata.org/wiki/Q501312
VQH95AU7	Koknhse Municipa lity	Koknese Municipalit y	https://www.wikidata.org/wiki/Q865682
VQH95AU7	Saulkrast i Municipa mity	Saulkrasti Municipalit y	https://www.wikidata.org/wiki/Q371429
Y95CKWKP	2003 Parisâ€“ Dourges	2003 Paris–Bourges	https://www.wikidata.org/wiki/Q3895876
VNMESCIH	1968 Patis-Tours	1968 Paris-Tours	https://www.wikidata.org/wiki/Q41754987

KAYNAKLAR

Abdelmageed, N., & Schindler, S. (2020). JenTab: Matching Tabular Data to Knowledge Graphs. In *SemTab@ ISWC* (pp. 40-49).

Albert, R., Jeong, H., & Barabási, A. L. (1999). Diameter of the world-wide web. *nature*, *401*(6749), 130-131.

Alshemali, B., & Kalita, J. (2019). Toward mitigating adversarial texts. *International Journal of Computer Applications*, *178*(50), 1-7.

Kostylev, E. V., Reutter, J. L., & Ugarte, M. (2015). Expressiveness of CONSTRUCT Queries in SPARQL. *Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik*.

Antoniou, G., & Van Harmelen, F. (2004). *A semantic web primer*. MIT press.

Berners-Lee, T. J. (1992). The world-wide web. *Computer networks and ISDN systems*, *25*(4-5), 454-459.

Berners-Lee, T. (1999). *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, *284*(5), 34-43.

Bizer, C., Heath, T., & Berners-Lee, T. (2011). Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts* (pp. 205-227).

Bizer, C., Heath, T., Idehen, K., & Berners-Lee, T. (2008, April). Linked data on the web (LDOW2008). In *Proceedings of the 17th international conference on World Wide Web* (pp. 1265-1266).

Brügger, N. (2012). Web history and the web as a historical source. *Zeithistorische Forschungen–Studies in Contemporary History*, *9*(2), 316-325.

Choudhury, N. (2014). World wide web and its journey from web 1.0 to web 4.0. *International Journal of Computer Science and Information Technologies*, *5*(6), 8096-8100.

Christen, P. (2012). The data matching process. In *Data matching* (pp. 23-35). Springer,

Berlin, Heidelberg.

Cyganiak, R. (2005). A relational algebra for SPARQL. *Digital Media Systems Laboratory HP Laboratories Bristol. HPL-2005-170*, *35*, 9.

Davies, J., Fensel, D., & Van Harmelen, F. (Eds.). (2003). *Towards the semantic web: ontology-driven knowledge management*. John Wiley & Sons.

Gagliardi, G., Gregori, L., & Ravelli, A. A. (2020, May). An NLP pipeline as assisted transcription tool for speech therapists. In *Proceedings of the 12 th International Conference on Language Resources and Evaluation (LREC 2020)* (pp. 124-130).

Gartner, R., & Gartner, R. (2016). *Metadata*. Springer.

Gómez-Pérez, A., & Corcho, O. (2002). Ontology languages for the semantic web. *IEEE Intelligent systems*, *17*(1), 54-60.

Hartig, O., & Langegger, A. (2010). A database perspective on consuming linked data on the web. *Datenbank-Spektrum*, *10*(2), 57-66.

Heath, T., & Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, *1*(1), 1-136.

Hendler, J. (2009). Web 3.0 Emerging. *Computer*, *42*(1), 111-113.

How to Round Numbers in Python? - GeeksforGeeks. (2020). Erişim adresi: https://www.geeksforgeeks.org/how-to-round-numbers-in-python/. Erişim tarihi: 20 Ağustos 2020

Kim, D., Park, H., Lee, J. K., & Kim, W. (2020). Generating Conceptual Subgraph from Tabular Data for Knowledge Graph Matching. In *SemTab@ ISWC* (pp. 96-103).

Diyasa, I. G. S. M., Mandenni, N. M. I. M., Fachrurrozi, M. I., Pradika, S. I., Manab, K. R. N., & Sasmita, N. R. (2021, May). Twitter Sentiment Analysis as an Evaluation and Service Base On Python Textblob. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1125, No. 1, p. 012034). IOP Publishing.

Melaku, T. (2017). Automatic Spelling Checker for Amharic Language. Doktora tezi, Bahir Dar Üniversitesi, Etiyopya

Miller, E., & Swick, R. (2003). An overview of W3C semantic web activity. *Bulletin of the American Society for Information Science and Technology*, *29*(4), 8-8.

Nath, K., Dhar, S., & Basishtha, S. (2014, February). Web 1.0 to Web 3.0-Evolution of the Web and its various challenges. In *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)* (pp. 86-89). IEEE.

Nguyen, P., Yamada, I., Kertkeidkachorn, N., Ichise, R., & Takeda, H. (2020). MTab4Wikidata at SemTab 2020: Tabular Data Annotation with Wikidata. In *SemTab@ ISWC* (pp. 86-95).

O'reilly, T. (2005). What is web 2.0.

Olteanu, A., Mustière, S., & Ruas, A. (2006, July). Matching imperfect spatial data. In *Caetano, M., Painho, M.(Es.), Proceedings of 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences. Lisbon* (pp. 7-9).

Patel, K. (2013). Incremental journey for World Wide Web: introduced with Web 1.0 to recent Web 5.0–a survey paper. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(10).

Patil, H. J., & Surwade, Y. P. (2018). Web technologies from web 2.0 to web 4.0. *International Journal for Science and Advance Research In Technology*, *4*(4), 810-814.

Pérez, J., Arenas, M., & Gutierrez, C. (2006, November). Semantics and Complexity of SPARQL. In *International semantic web conference* (pp. 30-43). Springer, Berlin, Heidelberg.

Piscopo, A., & Simperl, E. (2019, August). What we talk about when we talk about Wikidata quality: a literature survey. In *Proceedings of the 15th International Symposium on Open Collaboration* (pp. 1-11).

Python | Add comma between numbers - GeeksforGeeks. (2019). Erişim adres: https://www.geeksforgeeks.org/python-add-comma-between-numbers/. Erişim tarihi: 01 Temmuz 2019

Python | Remove spaces from a string - GeeksforGeeks. (2019). Erişim adresi: https://www.geeksforgeeks.org/python-remove-spaces-from-a-string/. Erişim tarihi: 17 Ocak 2019

Python | Removing unwanted characters from string - GeeksforGeeks. (2020). Erişim adresi:https://www.geeksforgeeks.org/python-removing-unwanted-characters-from-string/. Erişim tarihi: 25 Eylül 2020

Schmidt, M., Meier, M., & Lausen, G. (2010, March). Foundations of SPARQL query optimization. In *Proceedings of the 13th International Conference on Database Theory* (pp. 4-33).

Semantic Web Challenge on Tabular Data to Knowledge Graph Matching. (2020). Erişim adresi: https://www.cs.ox.ac.uk/isg/challenges/sem-tab/2020/index.html.

Shekhawat, B. S. (2019). *Sentiment Classification of Current Public Opinion on BREXIT: Naïve Bayes Classifier Model vs Python's TextBlob Approach*. Dokta tezi, İrlanda Ulusal Koleji

Shivalingaiah, D., & Naik, U. (2008). Comparative study of web 1.0, web 2.0 and web 3.0.

Spelling checker in Python - GeeksforGeeks. (2020). Erişim adresi: https://www.geeksforgeeks.org/spelling-checker-in-python/. Erişim tarihi:07 Ekim 2020

Steinacker, A., Ghavam, A., & Steinmetz, R. (2001). Metadata standards for web-based resources. *IEEE MultiMedia*, *8*(1), 70-76.

Taye, M. M. (2010). Understanding semantic web and ontologies: Theory and applications. *arXiv preprint arXiv:1006.4567*.

Wikidata. (2020). *Wikidata Query Service*. Erişim adresi: https://query.wikidata.org/

Yumusak, S. (2020). Knowledge graph matching with inter-service information transfer. In *SemTab@ ISWC* (pp. 104-108).

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Arife KARAAĞAÇ

EĞİTİM DURUMU

Lisans Öğrenimi :

2018, Altınbaş Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik ve Elektronik Mühendisliği (%100 Burslu)

Yüksek Lisans Öğrenimi :

2021, KTO Karatay Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik ve Bilgisayar Mühendisliği

Bildiği Yabancı Diller :

İngilizce (Yökdil : 65)

İŞ DENEYİMİ

Stajlar :

2016, EEM Asansör Kumanda Sistemleri

2017, Meram Belediyesi

Tarih: 06 Eylül 2021