İNTERNET SERVİS VE KAYNAKLARI ÜZERİNDEN İŞLENMİŞ VERİ SAĞLAYAN BİR PLATFORM GELİŞTİRİLMESİ

DEVELOPING A PLATFORM THAT SUPPLIES PROCESSED INFORMATION FROM INTERNET RESOURCES AND SERVICES

Nicat SÜLEYMANOV

Prof. Dr. İlyas ÇİÇEKLİ
Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

Bilgisayar Mühendisliği Anabilim Dalı için Öngördüğü

YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2015

Nicat Süleymanov'un hazırladığı "İnternet Servis ve Kaynakları Üzerinden İşlenmiş Veri Sağlayan Bir Platform Geliştirilmesi" adlı bu çalışma aşağıdaki juri tarafından BİLGİSAYAR MÜHENDİSLİĞİ DALI' nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Prof. Dr. Hayri Sever .....

Başkan

Prof. Dr. İlyas Çiçekli .....

Danışman

Prof. Dr. Ferda Nur Alpaslan .....

Üye

Prof. Dr. Ahmet Coşar .....

Üye

Yrd. Doç. Dr. Gönenç Ercan .....

Üye

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Fatma SEVİN DÜZ
Fen Bilimleri Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

08/09/2015

NİCAT SÜLEYMANOV

Değerli aileme…

ÖZET

İNTERNET SERVİS VE KAYNAKLARI ÜZERİNDEN İŞLENMİŞ VERİ SAĞLAYAN BİR PLATFORM GELİŞTİRİLMESİ

Nicat SÜLEYMANOV

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. İlyas ÇİÇEKLİ

Eylül 2015, 74 sayfa

Her gün artmakta olan internet verileri arasından istenilen ve doğru veriye ulaşılması zorluklar getiriyor ve kullanıcı, arama motorlarında sorusuna cevap olarak 10 milyon cevap değil, en uygun görülen 10 cevabı, varsa eğer tek doğru cevabı istiyor. Bu çalışmada, Türkçe sorular için soruları çözümleyen ve internet servis ve kaynakları üzerinden bilgi çıkarımı gerçekleştirecek soru cevaplama sistemi anlatılmaktadır. Soru analizi kısmı aşamasında kelimelerin gövde bilgisinden ve örüntü tanıma teknikleri kullanılarak belli soru tiplerini algılayan ve aynı zamanda sorudaki kelimelerin anlam ve morfolojik yapı özelliklerinden varsayımlar yaparak iki aşamalı çözüm yolu denenmiştir. Ayrıca, soru analizi başarı oranını yükseltmek amacıyla WordNet çatısı da kullanılmıştır. Bilgi çıkarımı kısmında internet kaynak ve servislerine başvurulmuştur. Her gün artan internet verilerine erişimin kolaylaştırılması kapsamında Tim Berner Lee tarafından gerçekleştirilen anlamsal ağ çözümü bu çalışmada bilgi çıkarımında kullanılmıştır.

Vikipedi verilerini işleyerek anlamsal ağ kaynağına dönüştürülmesini gerçekleştiren Dbpedia kaynağı kullanılmıştır. Doğal dil Türkçe sorularının doğal dil işleme ve bilgi çıkarma teknikleri kullanılarak anlamsal ağdasorulan ve makinaların anlayacağı Sparql sorgularına dönüştürülmesi sağlanmıştır. Dbpedia'dan çekilmiş Özlem-Yüklem-Nesne bilgileriyle Türkçe sorular arasında eşleştirmeler yapılması ve eşleşen üçlüler üzerinden cevaba ulaşılması sağlanmıştır. Çalışma sürecinde Türkçe-İngilizce karşılaştırma kapsamında ve veri sorgularında Vikipedi API'si de fayda sağlamıştır.

Anahtar kelimeler: Soru cevaplama sistemi, Doğal dil işleme, Veri çıkarımı, Anlamsal ağ

ABSTRACT

DEVELOPING A PLATFORM THAT SUPPLIES PROCESSED INFORMATION FROM INTERNET RESOURCES AND SERVICES

Nicat SULEYMANOV

Master of Science, Computer Engineering Department

Supervisor: Prof. Dr. IlyasCICEKLI

September 2015, 74 pages

Every day increasing information resources makes it harder to reach to the needed piece of information and users do not want 10 billion results from search engines, but they prefer 10 best matched answers, even if it exists they prefer the right answer. In this research,we presenta Turkish question answering system that extracts the most suitable answer from internet services and resources. During the question analyzing period, the question class is determined,from the lexical and morphological properties of words in the question certain expressions are predicted and our two-stage solution approach tries to get the answer. Furthermore, to increase the success rate of the system, WordNet platform is used.

In information retrieval process, the system works over the documents using semantic web information instead of classic search engine retrieved documents. In order to reach easily to needed information among increasing resources, Tim Berner Lee's idea of semantic web information is used in this research. Dbpedia which extracts structural information from Wikipedia articles and also this structured information is accessible on the web. In our research, the matched subject-predicate-object triples with asked question is formulated to get answer in Turkish, for searching and getting Turkish equivalent of the information Wikipedia Search API and Bing Translate API is used.

Keywords: Question answering system, Natural language processing, Information retrieval, Semantic web

TEŞEKKÜR

Bu çalışmada yardım ve desteğini gördüğüm pek çok kişiye teşekkür borçluyum . En başta, tez yazma sürecinde bilgi , fikir ve yorumlarını benimle paylaşan değerli danışmanım ve hocam Prof. Dr. İlyas ÇİÇEKLİ'ye sonsuz teşekkür ederim .

Başım her sıkıştığında kapısını çaldığım , dibe vurduğum anlarda beni cesaretlendiren ve her zaman desteğini hissettiğim değe rli arkadaşım Cemil Zalluhoğlu'na büyük teşekkür borçluyum.

Zaman ayırarak çalışmamı değerlendiren , geri bildirim veren ve değerli jürimde yer alarak görüşlerini paylaşan sevgili hocalarım Prof. Dr. Hayri SEVER, Prof. Dr. Ferda Nur ALPASLAN, Prof. Dr. Ahmet COŞAR, Yrd. Doç. Dr. Gönenç ERCAN' a çok teşekkürler.

Teknik konularda yardımlarına ve fikirlerine başvurduğum iş arkadaşım ve yazılım koordinatörüm Vüsal MESİYEV'e çok teşekkür ederim.

Araştırmam süresince ilgi ve desteklerinden dolayı değerli bölüm hocalarıma teşekkürlerimi sunarım.

Sevgili arkadaşım Turkan İSMAYILLI'ya güler yüzüyle bu süreçte hep yanımda olan ve verdiği fikir ve önerilerin yanı sıra her an desteğini hissettirdiği ve anlayışını esirgemediği için sonsuz teşekkür borçluyum.

İÇİNDEKİLER

ÇİZELGELER

Şekiller

SİMGELER VE KISALTMALAR

DDİ	Doğal dil işleme
API	Application Programming Interface - Uygulama programlama arayüzü
CERN	Conseil Européen pour la Recherche Nucléaire(Avrupa Nükleer Araştırma Merkezi)
JSON	Javascript Object Notation(JavaScript Nesne Notasyonu)
MTW	Meaning To Word(Anlamdan Kelimeye)
SCS	Soru Cevaplama Sistemleri
TREC	Text Retrieval Conference
XML	EXtensible Markup Language (Uzatılmış İşaretleme Dili)
SPARQL	SPARQL Protocol and RDF Query Language
RDF	Resource Description Framework - Kaynak Tanım Çerçevesi
RDFS	RDF Schema
OWL	Web Ontology Language
MQL	Metaweb Query Language
SQL	Structured Query Language
LAT	(Lexical Answer Type) Leksik Cevap Tipi
TDK	Türk Dil Kurumu
W3C	World Wide Web Consortium

1. GİRİŞ

1.1. Problemin önemi

Veriler daima birikir ve zaman geçtikçe artanveri içinde bilgiye ulaşmak zor olur ve sonuçta kolay şekilde gereken veriye erişmek için yeni yöntemler geliştirilmesine ihtiyaçduyulur. Veriler değişik yapıda, şekilde biriktirilebilir. CERN 2013 haberine[1] göre CERN veri merkezindeki fiziki veri kayıtları son 20 yıl içinde 100 petabaytı aşmış durumdadır. Her gün daha da genişlemekte olan verileri daha kullanışlı ve hızlı erişilebilecek yapıda tutmak ve gereken veriye kolay erişimi sağlayacak sistemler gerektirmektedir. Veriler yazı, resim, ses biçiminde olabilir ve kapsadığı alan dikkate alınarak geliştirilmekte olan sistemler koşullara uyum sağlamak zorundadır.

Genelde internet ortamında biriken veriler yapısal olmaz, dolayısıyla bir SQL sorguları veya anahtar kelimelerle kesin cevaba ulaşmak zor olur. Arama motorları soru karşılığında kendi değerlendirme kriterine uygun olarak bir internet sayfa bağlantı listesi sunar, başka şekilde söylersek bir doküman listesi sunar. Daha sonra dokümanlardan uygun pasaja bulmayı ve pasajdan sorunun cevabına erişmeyi soru soran kullanıcı kendisi yapar.

Arama motorları ve bilgi erişimi sistemleri dolayısıyla yeni bir alanın gelişmesine yol gösterir. Soru cevaplama sistemleri (SCS) olarak tanımlanan bu sistemler bilgi erişimi, bilgi çıkarı ve değişik doğal dil işleme tekniklerinden oluşan kompleks bir yapı içerirler ve genelde kapsadığı alan ve sorgu dili gibi koşullara göre bu sistemlerde farklı yöntemler izlenmektedir. Buna ek olarak soru şekilleri de farklılık gösterebilir, soru türlerinden bazıları tek cevaplı sorular ve tanımlama sorularıdır. Her soru şekli için özellikleri dikkate alarak değişik yollar izlenilerek daha başarılı sonuçlara ulaşmak mümkündür.

Yazılı metinlerde dil farkının var olması, gereken dildeki veriye erişmek için farklı çözümler ortaya koymayı gerektirmektedir. Bu çalışmada geliştirilen platform Türkçe için tasarlanmış olup ileride diğer Türk dili ailesindeki diller için de kısmi değişikler yapılarak benzer ve mümkün oldukça daha iyi sonuçlar almak için çalışmalar yapılması planlamaktadır. Türkçe ve İngilizce soru-cevaplama sistemlerini kıyaslarsak, İngilizce'de soruyu anlamak daha az işlem gerektirir ve aynı zamanda soruya cevap alanları genelde daha geniştir. Türkçenin sondan eklemeli dil olması sorudaki kelimeleri ve ilişkileri anlamakta daha fazla doğal dil işleme teknikleri kullanmayı gerektirmektedir.

Geliştirilen platform Türkçe yazılı metinler için tasarlanmış, günlük bilgi öğrenmek amaçlı sorulan ve genelde cevabını Vikipedi makalelerini okuyarak bulunabilecek verilere erişimi sağlamayı hedeflemektedir. Bundan başka diğer veri kaynakları da gözden geçirilmiş ve kullanılabilirliğine dair çalışmalar yapılmıştır.

1.2. Çalışmanın amacı

Çalışmanın amacı Türkçe sorulan soruya internet servis ve kaynaklarından faydalanarak gereken veriye erişimi sağlamaktır. Çalışmada daha çok Vikipedi kaynaklarında bulunabilecek verilere erişimi sağlayacak sistem üzerine yoğun çalışılmıştır. İlk aşamada soruyu işlemek, odak kelimeleri bulmak ve onlar üzerinden veri çekimini sağlamak gerekmektedir. Veri çıkarma kısmında anlamsal ağ sorgulama şekli olan Sparql sorgusu oluşturulmaktadır. Sparql sorgusu sayesinde Türkçe veya İngilizce Dbpedia veri alanından veri çekimi sağlanır ve daha sonra gerekirse bulunan sonucu Türkçeye çevirerek soruya cevap bulunur. Sistem Türkçe için geliştirilse de, diğer Türk dili ailesi üyelerine de kolaylıkla dönüştürülebilir esnek yapıda tasarlanmıştır.

1.3. Tezin organizasyonu

İkinci bölümde benzer çalışmalar anlatılmaktadır ve üçüncü bölümde Türkçe ve İngilizce dillerinde geliştirilen soru cevaplama sistemlerinin değerlendirilmeleri yapılacaktır. Araştırmada kullanılan yapılar ek kütüphaneler sistemlerle ilgili bilgi dördünce bölümde verilecek, sonraki bölümde de ilerideki araştırmaların nasıl bir yol izlemesi gerektiği konularına ve sonraki çalışmalarda başarı oranının artırılması için neler yapılabileceğine dair varsayımlara yer verilecektir. En son genel bulgu ve değerlendirmeler ve sonuçlar sunulacaktır.

2. GENEL KAVRAMLAR

2.1. Bilgi Erişimi

Bilgi erişimi (information retrieval) yapısal veri birikimlerinden: ilişkisel veri tabanlarından ve aynı zamanda geniş dünya ağında arama yaparak belli bir bilgi içeren dokümanların aratılması üzerinde çalışma yapan bir bilgisayar bilimleri dalıdır. Aranan veri herhangi bir bilgi içeren doküman olabilir. Örneğin, metin bilgisi, özellik bilgisi ve benzerleri [2]. Hayatın her alanında günlük olarak, ister arama motorları ile internette arama yaptığımızda, ister e-posta kutusunda bir şey arattığımızda, bilgi alma işlemlerinden faydalanmaktayız. Günümüzün bilgi çağında, bilgi alma işlemi veri tabanlarından veri çekimi prosedürünü çok geride bırakmış kompleks işlemler topluğu olmaktadır.

İlk olarak bilgisayarlardan gereken veri parçaları alma fikri 50'lerde Vannevar Bush tarafından "As We May Think" makalesinde geçmiştir [3]. 70'li yıllara gelindiğinde metin bilgilerinden farklı bilgi alma teknikleri kullanılmaktaydı. 70'lerin başlarında artık "Lockheed Dialog" gibi geniş çaplı sistemler geliştirilmekteydi [4]. 1992'de artık TIPSTER'in bir parçası olan TREC metin alma konferansları gerçekleştirildi [5]. Arama motorlarının gelişmesiyle yeni bir evreye başlandı ve çok büyük ölçekli veri alma sistemlerine ihtiyaç duyuldu.

Sorgular bilgi almada kullanılan ifadelerdir. Arama motorlarında arama kutusuna girilen kelimeler sorguları oluşturmaktadır. Sorgu tek bir nesneyi tanımlamamakta, belirli özellik taşıyan nesne-doküman listesi, gerektiğinde derecelendirmeli bir şekilde aratmak için çalıştırılır. Aranan obje ise veri tabanında tanımlanmış bir nesne, bir bilgidir. Obje audio dokümanı, video, harita ve benzeri şekillerde olabilmekte ve genelde veri tabanında yapısal şekilde bulunmamakta, ya da bir metadatanın bir parçası olarak yer alabilmektedir. Bilgi alma sistemleri sorgulara en çok eşleşen bilgileri ve skorlarına göre derecelendirmeler yaparak hesaplamaktadır. Sonuç olarak, skorları en yüksek olan nesneler kullanıcıya sunulmaktadır. Özet olarak, "Bilgi alma kullanışlı bilgiler içeren geniş yapısal olmayan alandan(genelde metin alanlarından) eşleşen nesneleri(dokümanları) bulma işlemidir." [2].

2.2. Soru Cevaplama Sistemi

Soru-cevaplama sistemleri aranan soru ile cevap aranan metinler içinde biçimsel ve anlamsal benzerlikler aranmasını dikkate alarak geliştirilmektedir. İnternetteki bilgilerin hızla artması sonucu ve arama motorlarının geliştirilmesi sonrası Doğal Dil İşleme(DDİ)

yöntemleri kullanılarak geliştirilmeye başlanan soru cevaplama sistemleri de yeni potansiyel bir bilim alanı olarak görünmeye başlamıştır. İlk soru cevaplama sistemleri genelde sınırlı konular için tasarlanmış olup, TREC-8 sonrasında artık konudan bağımsız gerçek dünya verileri üzerinden geniş ölçekli soru cevaplama sistemlerine doğru yol izlemeye başlanmıştır [6]. Bugünün internet dünyasında yaygın olarak kullanılan arama motorları aranan bilgiyi içeren veya ilgili olan dokümanları kullanıcılara belirli özelliklere göre sıralı bir biçimde sunmaktadır. Kullanıcılar sunulan bu doküman listesinden ihtiyaç duyduğu bilgiyi kendi çıkarmak zorundadır.

Kullanıcıyı bilgi dokümanları içinde gezinmekten, yani bilgi çıkarımı işleminden kurtarmak, özellikle kesin bir cevabı olan sorularda ihtiyaca dönüşmüştürve bu ihtiyacı karşılamak demek soru cevaplama sistemlerini daha da ileriye götürmek demektir. Günümüzde bu tür sistemler ve araçlar bulunmaktadır. Zaman geçtikçe internet dünyasında bilgi miktarı arttıkça paralel olarak bilgi çıkarımı işleminin önemi de artmaktadır. SCS yalnızca arama motorlarının sunduğu dokümanlardan veri çıkarımı işlemi olmayıp, farklı alanlarla da ilişkilidir.

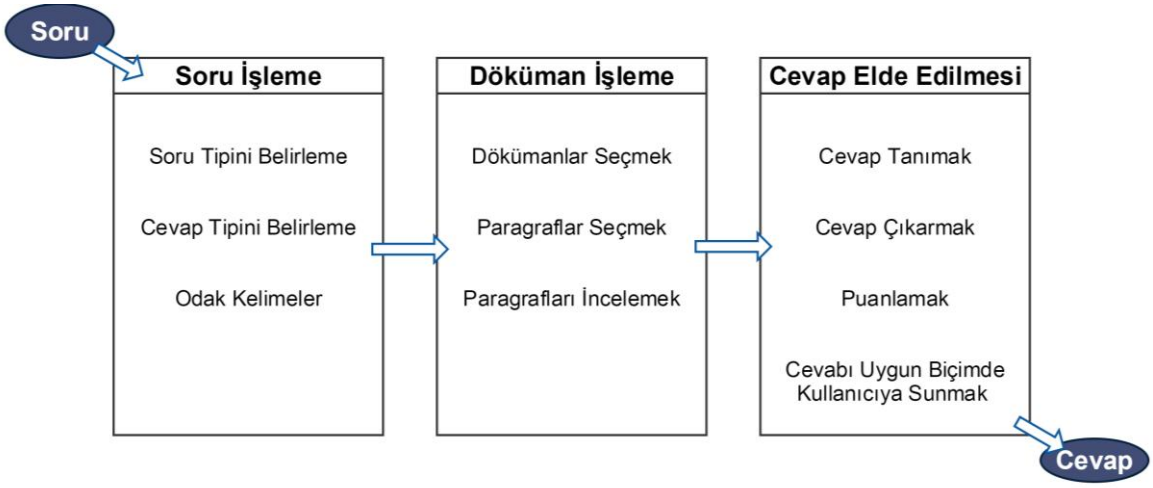
Güncel arama motorları dokümanların listesini sunmasına karşı, SCS'nin görevi, bilgi alma, DDİ ve soru cevaplama teknikleri kullanarak soruyu yanıtlamaktır. Doğal dilde sorulan herhangi bir soruyu yanıtlamak için soru cevaplama sistemi ister dünya internet ağı olsun, ister yerel metin koleksiyonlarından biri veya bazıları, önceden yapılandırılmış veri tabanı veya doğal dil belge koleksiyonlarını kullanabilir. Bu çalışmada soru cevaplama sistemi semantik ağ yapısındaki verilerden bilgilerin çekilmesi üzerinden yola çıkılmıştırve bu soru cevaplama sisteminde gerçekleştirilen prosedürlerdebilginin çekilmesinde bazı kolaylıklar sağlamıştır. Son yıllardaki soru cevaplama sistemlerine genel bakıldığında, artmakta olan internet kaynak ve verilerine karşı yapısal semantik ağ veri kaynaklarına doğru bir geçiş dikkat çekmektedir.

Veri kaynakları türlerine ek olarak, soru sınıflarına göre de farklı çözüm yolları üzerine araştırmalar gerçekleştirilmektedir. Diğer taraftan veri alanlarının kapalı bir çevreyi veya açık alan soru cevaplama sistemleri altında iki farklı çatı görünmektedir. Bu açıdan incelendiğinde verilerin büyüklüğü, alan içeriği ve kaynaklar farklılık göstermektedir. Kapalı alan çalışmalarında alana özgü yöntemlerin kullanılması kolaylık sağlamaktadır ve genellikle tercih edilmektedir.

Soru cevaplama sistemlerinde yaklaşımlar da derlem-tabanlı ve bilgi-tabanlı soru cevaplama sistemleri olarak ikiye ayrılmaktadır. Derlem-tabanlı sistemler alan-bazlı kaynaklardan ve yardımcı araçlardan faydalanmaktadır. Bilgi tabanlı soru cevaplama sistemleri üçdal altında incelenebilir: semantik-tabanlı, sonuç-çıkarma tabanlı ve mantık tabanlı. Semantik tabanlı açık alan sistemleri genelde leksik-semantik bilgiler topluluğu olan WordNet'i kullanmaktadır [7].

2.2.1. Soru cevaplama Sisteminin Genel Yapısı

Soru cevaplama sistemleri genel olarak soru işleme, doküman işleme, cevap çıkarma ve cevap oluşturma süreçlerini kapsamaktadır ve cevap tahmini yapmayı hedeflemektedir.



Şekil 2.1. Soru cevaplama sistemlerinde genellikle takip edilen yöntem

Şekil 2.1 bir soru cevaplama sisteminin temel olarak üç bölümden olduğunu göstermiştir. Soru analizi aşamasında önemli çalışmalardan birisi soru tipinin belirlenmesidir. Örneğin, "Azerbaycan'ın başkenti neresidir?" sorusunun cevabının bir yer adı tipi olduğunun belirlenmesi ve soru tipinin yer adı bulma sorusu olduğunu bulmak soru işleme modülünün görevleri arasında gösterilebilir. Dolayısıyla, cevap tahminleri bulunduğunda soru tipi dikkate alınarak uygun görünmeyen tahminler elenmektedir veya arama kısmında yer adı bilgileri sorgulatılmaktadır. Doküman işleme sırasında arama motorlarından dokümanları alarak cevap tahminlerini içerdiği tahmin edilen pasajlar mercek altına alınır. Cevap çıkarma sürecinin sonunda çeşitli yöntemler kullanılarak aday cevaplar bulunur ve bu pasajlardan çıkartılır. En son kısımda gerekirse, cevap şekline dönüştürülerek kullanıcıya sunulur [8].

2.3. WordNet

İlk olarak Princeton Üniversitesi tarafından 1985 yılında İngilizce için başlatılan WordNet, kelimelerin semantik ağı veya anlamsal veri tabanı olarak nitelendirilebilir [9]. Adı geçen sistemin internet arayüzüne http://wordnetweb.princeton.edu/perl/webwn adresinden ulaşılabilir. Zamanla DDİ alanında farklı projelerde boy gösteren proje soru cevaplama sistemlerinde de yaygın olarak kullanılmaktadır. Türkçe WordNet çalışmaları Oflazer ve arkadaşları tarafından Balkanet projesi kapsamında oluşturulmuştur [10]. Diğer bir Türkçe WordNet oluşturma çalışması Şerbetçi ve arkadaşları tarafından Türkçeden Türkçeye sözlük bilgilerinden özellik çıkarma yöntemleri kullanılarak geliştirilmiştir [11]. Çalıştığımız araştırma kapsamında Oflazer'in oluşturduğu Türkçe WordNet'ten kelime çözümleme ve ilişki çözümleme kısımlarında fayda sağlanmıştır.

Çizelge 2.1. Kemal Oflazer ve takımının WordNet çalışması sonucu tanımlanan ilişki tipi ve sayısı bilgileri

İlişki Tipi	Sayısı
HYPERONYM	11251
HOLO_MEMBER	946
HOLO_PART	1423
HOLO+PORTION	176
CAUSES	100
BE_IN_STATE	577
NEAR_ANTONYM	1400
SUBEVENT	127
ALSO_SEE	270
VERB_GROUP	896
CATEGORY_DOMAIN	384
TOPLAM	17550

Oflazer araştırması temelde Princeton WordNet modelini izler. EuroWordNet projesi kapsamında geliştirilen altı sistemden biri olan bu çalışma, Türkçe için "Anlamdan-Kelimeye" yolubulan sistemiçin defayda sağlamıştır ve araştırmadaörnekler üzerinden %66-%68 başarı gösterildiği belirtilmiştir [12]. MTW (Meaning To Word - Anlamdan Kelimeye) sistemi Türkçe WordNet'inden eş anlamlılık bilgilerini ve kelimeler arası ilişkileri kullanmakta ve anlam bilgilerinden kelimeyi bulmakta benzerlik bilgilerini ve

Türkçe kelimelerin anlam bilgilerinden faydalanmaktadır. WordNet'ten başka bu MTW sistemi de ileride soru cevaplama sistemleri için kullanılabilir.

Oflazer ve arkadaşlarının geliştirdiği çalışma [10] sonucu tanımlanan semantik ilişki listesi Çizelge 2.1.'de yer almaktadır. Fiil grupları, eşanlamlı daha üst grup kelime bilgileri gibi ilişkiler bu anlamda sistem tarafında benzer kelimelerin bulunmasında ve soru sınıflarını belirlemekte ve şablon eşleştirmesi zamanı kolaylık sağlamıştır.

2.4. JSON

JSON (JavaScript Object Notation), çoğu programlama dilleri tarafından ve internet servisleri tarafından desteklenen ve platform bağımsız veri tanımlama şekli olup, günümüz dijital dünyasında yaygın olarak kullanılmaktadır.Sahip olduğu özelliklerle JSON veri haberleşmesi ve gösterimleri için ideal sayılabilecek bir yapıdadır.

Veri paylaşımında kullanılan bazı standart biçimlendirme yöntemleri vardır. Bunların başında Uzatılmış İşaretleme Dili (XML) gelse de gerek kolay okunabilir oluşu, gerekse de nesneye dayalı dillerle daha uyumlu kullanılabilmesi sebepleriyle JavaScript Nesne Notasyonu (JSON) son zamanlarda yaygın olarak kullanılmaktadır. Bu standarda göre küme parantezi içine nesne, köşeli parantez içine ise diziler yerleştirilir. Her bir özellik virgülle ayrılır. Nesnelerin dizi olmasının mümkün olduğu gibi, bir nesne özelliğinin de başka bir nesne olması veya dizi olması mümkündür. JSON iki yapı üzerine kurulmuştur:

- İsim/değer çifti koleksiyonu. Çeşitli programlama dillerinde bu, "object(nesne), record(kayıt), struct(yapı), dictionary(sözlük), hash table(komut çizelgesi), keyed list veya associative array(ilişkisel dizi)" olarak da tanımlanmıştır.
- Sıralı değer listesi. Çoğu programlama dilinde bu, "array, vector, list veya sequence" olarak tanımlanır. Özelliklerin isimleri ile değerleri arasına “:” işareti konularak birbirinden ayrılırlar. Ayrıca gerek özellik ismi gerekse de taşıdığı değer tırnak içerisinde yazılır [13].

Bu yapılar, evrensel veri yapılarıdır.Bütün modern programlama dilleri, bu yapıları, bir şekilde içlerinde barındırmaktadırlar. Programlama dilleri arasında veri değişimi için kullanılan bir formatın, bu yapıları kullanarak oluşturulması da oldukça anlamlıdır. Bu yapılar JSON'da, aşağıdaki şekillerde gösterilirler:

```
{
 "words": {
  "SYNSET": [
{
      "ID": "BILI-60000001",
      "SYNONYM": {
        "LITERAL": {
          "#text": "ayva reçeli",
          "SENSE": "1"
}
},
      "SNOTE": "Quince jam.",
      "POS": "n",
      "ILR": {
        "#text": "ENG20-07172978-n",
        "TYPE": "hypernym"
},
      "STAMP": "orhanb 2004/07/16"
},
...
]
}
}
```


Yukarıdaki örnekte, “words” JsonObject'i içinde “synset” JsonArray’i yer almakta, “synset” JsonArray'de de JsonObject'ler dizisinden oluşmaktadır. Özet olarak, XML yapısından daha sonra geliştirilmiş bir yapı olup, bilgisayar tarafından daha hızla anlaşılabilir olduğundan daha fazla yaygınlık görmüş veri yapısı şeklidir. İster platformun geliştirildiği java, isterse de diğer programlama dillerinde çeşitli kütüphaneler aracılığıyla veri haberleşmesinde kullanılmaktadır.

Çalışma kapsamında hem Vikipedi API'sinden veri çekimi zamanı, hem de Jena çatısı sayesinde SPARQL sorgularından veriler JSON yapısında kabul edilmekte, aynı zamanda MongoDB veri tabanında da veriler JSON teknolojisi kullanılmaktadır.

2.5. Anlamsal Ağ Teknolojileri

2.5.1. Anlamsal Ağ

Dünya internet ağı internet sayfalarından oluşmakta ve insanların anlayabileceği şekilde göstermek niteliği taşımaktadır. Bunun yanı sıra bilgisayarların ve yazılım programlarının da anlayacağı ve veri paylaşımı gerçekleştirebileceği anlamsal internet teknolojisi Tim Berners Lee tarafından ortaya atılmıştır [14]. Farklı sistemlerin anlamsal ağ alanlarının sistemden, sentakstan kaynaklanan ve aynı zamanda semantik heterojenliği de söz konusudur ve farklı çözüm yolları bulunmaktadır. Anlamsal ağ, internet teknolojilerinin gelişimi ve bu teknolojilerin geleceği açısından çok önemli bir kavram olarak değerlendirilebilir. 2001 yılında başlayan anlamsal ağ (web 3.0) çalışmaları W3C tarafından başlatılmıştır. Yakın gelecekte geçilmesi beklenilen web 3.0 versiyonunun tamamen anlamsal ağ altyapısı üzerine kurulması düşünülmektedir.

Berners Lee tanımıyla anlamsal ağ, bilginin iyi tanımlanmış anlamıyla birlikte verilmesiyle bilgisayarlar ve insanların birlikte çalışmasını kolaylaştıran ve bilgisayarların birbirinin dilinden anlamasını sağlayan şuanki internetin uzantısıdır ve internet sayfalarının anlamsal içeriğinin geliştirilmesi sayesinde internet sayfaları arasında gezinen yazılım ajanlarının (software agents) kullanıcılar için kolaylıkla karmaşık görevler i yerine getirebilmesidir. Daha da açıklarsak, anlamsal ağ, internet ortamındaki kaynakların daha etkili bir şekilde bulunması, otomatikleştirilmesi, entegrasyonu ve bir çok uygulama tarafından tekrar kullanılabilirliğini sağlayacak şekilde tanımlanmı ş ve linklenmiş olması fikridir . Anlamsal ağ kaynakları diğer kaynaklarla olan ilişkileriyle var olmaktadır . Anlamsal ağda veri "directed labeled graph"(yönlendirilmiş etiketli grafik) olarak modellenir ve her nokta bir kaynağa karşılık gelir , her kenar bir ilişki tipi ile etik etlenir [15]. Başka bir deyişle , anlamsal ağ bilgisayarların anlayacağı dilde internet kaynakları sağlar ya da mevcut internet kaynaklarını bilgisayarın anlayacağı meta veri ile genişletir (URL-5, 2005). Anlamsal ağ mimarisinin katranlandırılmış gösterimi Şekil2.2'de gösterilmektedir.

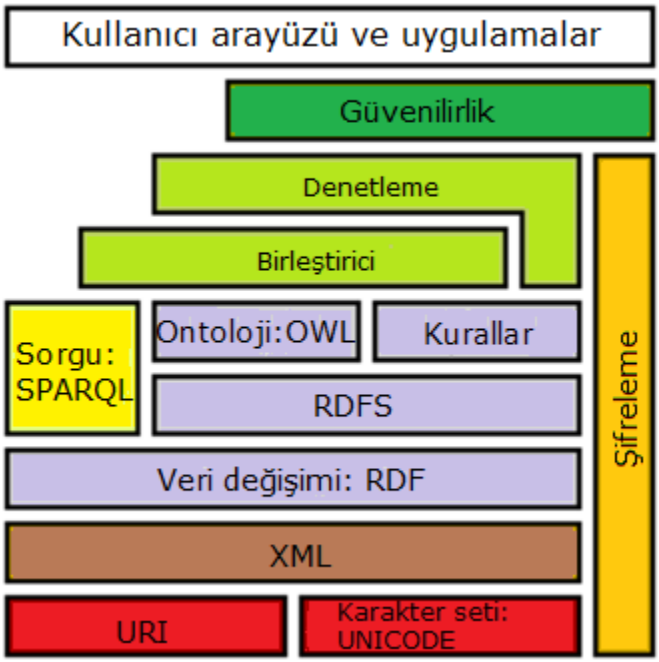
Anlamsal ağ RDF, RDFS ve OWL semantik tanımlama dillerini kullanılmasını gerektirmiştir. XML tabanlı bu diller sayesinde internet kaynakları semantik olarak tanımlanabilmekte ve kaynaklar arasındaki ilişkiler kurulabilmektedir.

Araştırma kapsamında geliştirilen sistem klassik soru cevaplama teknikleri zamanı kullanılan arama motorunun sunduğu dokümanlar üzerine değil, anlamsal ağ veri tabanı

üzerine internet servisleri kullanarak veri çekimi işlemlerinde daha kesin sonuçlara ulaşmayı başarmıştır.

2.5.2. Ontoloji

Anlamsal ağ veri kaynağının ontolojisi denildiğinde, ontoloji dillerinin RDF/XML yapılarında tanımlanan sınıflarının ve özelliklerinin toplamından oluşan veri paylaşma ve genel anlam tanımlama sistemi anlaşılır. Ontolojiler, resmi tanımlamalardır ve insanın bir bilgiyi nasıl düşündüğünü, tanımladığını bir modele referans ederek gösteren mantık tabanlı diller kullanılarak kavram ve ilişkilerin gösterildiği sistemlerdir.



Şekil 2.2. Anlamsal Ağın mimarisi

Ontoloji aynı zamanda belirli bir konuyla ilgili kavramları ve bu kavramlar arasındaki ilişkileri ve kısıtlamaları gösteren , ağaç yapısında modellenen hiyerarşik bir sınıfla ndırma şeklidir. Aslında, bir tür veri sözlüğü ya da kavramsal şema olarak da ifade edilebilir . Ontolojideki kavramlar sınıf (class), kavram özellikleri "property" ve kısıtlamalar "constraint" olarak ifade edilir.

Her ontoloji standart bir şemaya referans vererek ontolojiler arasındaki entegrasyon sağlanır. Bu şekilde belirli standart ontolojilere referans vererek ontoloji oluşturulur ve ontolojilerin tekrar kullanılabilirliği sağlanır.

Ontoloji belli bir etki alanını (domain) temsil eden tüm kavramların bir arada tutulduğu oluşumdur ve her ontoloji kendi alanındaki kavramları barındırabileceği gibi, başka ontolojilerdeki kavramları da referans gösterebilir. İnternetortamındaki her kavramın belli bir etki alanında temsil edilmesi ve ilişkili olduğu diğer kavramlarla bağlantıları doğru bir şekilde tanımlanmalıdır. DBPedia ve Freebase gibi iki büyük oluşumun Vikipedia ve benzeri internet bilgi depolarını kullanarak oluşturdukları veri kümelerine sahiptir. Anlamsal ağ için veritabanı olarak görev yapan bu kaynaklar üzerinde sorgulama Sparql veya MQL gibi sorgulama dilleri kullanılarak kullanılabilmektedir. Freebase kendi üzerinde ek olarak adres satırından kaynak çağırma desteklense (mqlread) de henüz standart bir sorgulama yöntemi geliştirilmemiştir.

Geliştirilen sistemde ontoloji sınıf tanımlamaları ve sınıf özelliklerinden anlamsal ağ veri tabanından yapılan sorgular zamanı kullanılmıştır. Mesela aranan nesnenin kitap nesneleri arasından bulunması zamanı SPARQL sorgusunda "x rdf:type dbpedia-owl:Book" ontoloji sınfı belirteci eklenmektedir veya benzer biçimde sınıf özellikleri ve değerleri belirtilmekle daha kesin sonuçlara ulaşılması sağlanmaktadır.

2.5.3. RDF

RDF (Resource Description Framework - Kaynak Tanım Çerçevesi) XML yapısında bir veri modeli olup, veri nesnelerinin özelliklerini ve ilişkilerini belirtmek için kullanılır. İnternetin gelişme evrelerinin son periyoduna bakılırsa, RDF şema yapısıyla anlamsal ağ gelişmeye başladı. Bilinen ilişkisel veritabanı ve dosya yönetim sistemlerinde olduğu gibi, bu yapıda da belli ağ alanı altında veriler biçimsel bir yapı altındadır. RDF yapısında özne-yüklem-nesne üçlü ifade oluşturarak anlam ifade etmektedir. RDF temel yapı olup, üzerine yeni özellikler eklenerek, yeni ontoloji dilleri oluşturulmaktadır.

RDF makinaların da okuyup anlayabileceği ilk anlamsal ağ dilidir ve yapısı özne-yüklem-nesne üçlüsü olarak tanımlanmıştır ve genelde Yüklem (Özne,Nesne) şeklinde gösterilir. URİ(uniform resource identifier-tek biçimli kaynak belirleyici)'ler kaynaklara ulaşım adreslerini göstermek için kullanılır ve farklı şekillerde gösterilebilir:N3 notasyonu ile gösterimi, RDF/ XML yapısında da gösterimi, RDFS Graf şeması olarak gösterimi [16].

RDFS - RDF Schema, RDF'i ontoloji modelleme biçimi olup RDFS (Resource Description Framework Schema) Şubat 2004'de W3C onayını almış Şema (schema) tanımlama dilidir (W3C, 2004e). RDF dilini anlamsal olarak genişleten herhangi bir alanda kullanılacak olan sözcük kümesini tanımlayan bir tip tanımlama dilidir . RDF'den farklı olarak; sınıf bazında tanım, sınıflar ve ilişkiler arası hiyerarşi tanımlama ve aynı zamanda ilişkiler üzerine kısıtlamalar (domain ve range) yapmayı sağlar . Örneğin, Orhan Pamuk ve Türkiye arasında Dbpedia'da tanımlanmış doğrudan ilişkileri bakarsak Şekil 2.3'deki gibi bir sonuç elde ederiz. Eğer 2. dereceden de ilişkileri hesaba katarsakikinci kısımdaki gibi sonuç elde etmiş oluruz.

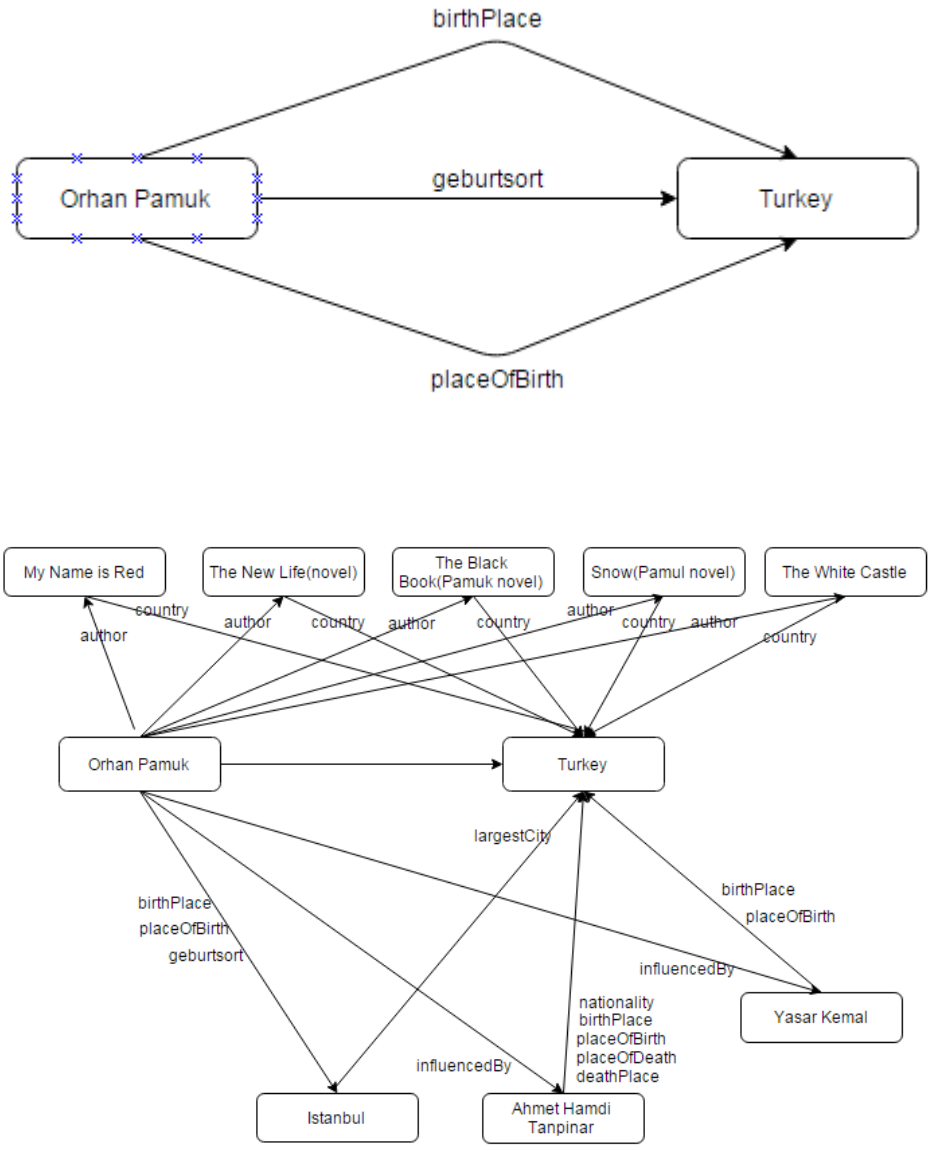
RDFS RDF verilerinde nitelik tanımı yapılırken bu tür bir şema kullanılır. XML şemalarında olduğu gibi bir RDF dosyasında tanımlanabilecek etiketler ve içeriklerini belirler. İlk olarak 1998 yılında yayınlanan bu yapı, 2004 yılında son halini almıştır. RDF için kullanılabilen şu etiketler bu şema içerisinde yer almaktadır. Sınıf bilgileri tanımlanırken;"rdfs:Resource" literal olmadığı sürece değer olarak kullanılan tüm kaynaklar bu etiket kullanılarak tanımlanır. "rdfs:Class"bir sınıf tanımı yapılırken kullanılır. "rdfs:literal" bir değer bilgisinin metin yapısında veri içerdiğini ifade eder. "rdfs:Datatype" bir instance (sınıftan türemiş nesne) için kullanılabilecek bir property'nin (özellik) "DatatypeProperty" mi, "ObjectProperty" mi olduğunu belirtmek için kullanılır. Bunun dışında özellikler için kullanılan diğer etiketler de vardır: "rdfs:domain", "rdfs:range", "rdfs:subClassOf", "rdfs:subPropertyOf" gibi [17].

Geliştirilen sistemde de SPARQL sorgularının "prefix" kısmında gösterilen RDF tanımlamalarıyla sınıf ve özellikler bilgileri kullanılarak kayıtlar üzerinden sorgulama yapılmaktadır.

2.5.5. SPARQL

SPARQL(SPARQL Protocol and RDF Query Language) RDF verileri için kullanılan bir sorgulama dili olup, ilişkisel veritabanından bilinen SQL Diline çok benzeyen bir yapı kullanır. SQL'den farklı olarak yalnızca SELECT sorgusu değil,verinin çekilmesi işleminde kullanılabilen ASK, DESCRIBE ve CONSTRUCT sorgu çeşitleri de kullanılabilir. SELECT SQL'den de bilindiği üzere veri kümesinin tamamını ya da WHERE kısmında belirtilecek koşula bağlı olarak belirli bir kısmını getirir. CONSTRUCT, aynı şekilde SELECT sorgusunda olduğu gibi koşula bağlı getirilen veri kümesini, doğru şekilde üçlüleri içerdiği takdirde bir RDF grafiği çıktısı üretmeye yarar.

ASK sorgulanan içeriğin veri kümesi içinde var olup olmadığı “true” ya da “false” olarak cevabını döndürür. Son olarak DESCRIBE ise WHERE kısmında filtrelenip elde edilen bilginin RDF veri kümesi içindeki tanımlamasını döndürür. DESCRIBE daha çok RDF veri kümesinin içeriğinin bilinmediği durumlarda kullanılır [18].



Şekil 2.3. Dbpedia alanında Orhan Pamuk ve İstanbul arasında tanımlanmış ilişki bağlarının görünümü

Özetle SPARQL klasik veri tabanlarından bilinen SQL’in semantik versiyonu olup RDF veri tabanlarından veri çekmek için kullanılan bir dildir. Dbpedia verilerinden faydalanmak için platformda veri getirme tarafında kullanılmıştır.

Araştırma kapsamında geliştirilen sistemde de veri çekimi anlamsal ağ veri tabanı üzerinden SPARQL sorguları sayesinde gerçekleştirilmektedir ve bu kapsamda Jena çatısı kolaylık sağlamaktadır.

2.5.6. OWL

OWL (Web Ontology Language) 2004 yılında W3C standardı olmuştur. Anlamsal ağ için ontoloji tanımlama dilidir. RDF sözcük kümesine genişletme getirmek için geliştirilmiştir. Bir OWL ontolojisi RDF grafiği yapısındadır . RDF grafik yapısı RDF üçlü ifadelerinden, yani Y(Ö,N)'den oluşur. Üç çeşit OWL ontolojisi yapısı mevcuttur:

OWL Lite: OWL türlerinin en az etkileyici olduğu yapıdır. OWL Lite'da kullanılan ekler basit miktar kısıtlamaları, yerel aralık kısıtlamaları, varoluşsal kısıtlamaları, eşitlik ve farklı özelliklere tür kısıtlamalarını kapsar.

OWL DL: olumsuzlama,veya (disjunction) için tam destek sağlayan ekleri, kıyaslama numaralandırması, değeri sınırlamaları, önem düzeyi kısıtlamaları gibi yapıları kapsar.

OWL Tam: kelime kullanımında kısıtlamalar yoktur ve OWL Lite ve OWL aksine RDF ifadelerin kullanılmasını da içerir.

OWL, bilgisayarların anlayacağı şekilde tasarlanmış RDF gibi XML tabanlı ontoloji tanımlama dili olmasına karşı DL(Description Logic) tabanlı olduğu için ek olarak , sınıf bazında daha fazla tanım ve kısıtlama yapmayı sağlar , özellikleri ve sınıfları betimlemek için daha fazla sözcük grubu getirir. Bunlar sınıflar arası ilişkiler, en önemlilik, eşitlik, özelliklerin daha iyi sınıflandırılması, özelliklerin karakteristikleri ve numaralandırılmış sınıflar (enumerated classes) gibi yapılardır [19].

2.6. DBPEDIA

Dbpedia Vikipedi verilerinden biçimsel veri oluşturmak için başlatılan projedir. Proje Free University of Berlin tarafından başlatılmış [20] ve 2014 istatistiklerine göre, İngilizce Vikipedi metinleri işlemiş ve buna ek olarak içinde Türkçe olmak üzere 28 dilde geliştirilmesine devam edilmektedir. Normal bir dil için Vikipedia metinlerinin işlenmesinden ilave farklı dillerdeki lokal işlenmiş verilerin de birbirleriyle ilişkilendirilmesi çalışmaları da paralel yapılmaktadır. En son 2014 veri setlerinde 4,584,616 İngilizce, 233,737 Türkçe metin işlenmiş, 143,914 Türkçe metnin İngilizce'yle ilişkilendirilmesi gerçekleştirilmiştir. En son 2014 istatistiksel bilgilerine göre, veritabanının Türkçe kısmı %9.3, İngilizce olan kısmı %7.7 daha genişlemiştir [21].

Dbpedia genelde Vikipedi metinlerinin sağında yer almakta olan “infobox” bilgi kutucuğundaki verilerden bilgi çıkarımı yapıyor ve bunun sayesinde örneklerin insan, yer ismi, olay ve başka sınıflara ve sınıflara ait özel özellikler altında verileri tutmak mümkün oluyor. Mesela, İngilizce kısmında 1,445,104 insan hakkında bilgi yer almaktadır ve bunun 268,773’ı sporcudur. Genel Dbpedia'da yer alan ontoloji sınıflara göre hiyerarşik ağaç yapısına “http://mappings.dbpedia.org/server/ontology/classes/” adresinden erişilebilir.

Dbpedia geniş çaplı ve farklı dilleri içinde kapsayan yapısal veritabanı sistemi olup, tek İngilizce veri alanı 2014 istatistiklerine [21] göre 458 milyon nesne (resource), 583 milyon olgudan oluşmaktadır. Dbpedia yapısal veri kaynağını Vikipedi makalelerinden bilgi çıkarımı sayesinde biriktirse de, Vikipedi'nın veya API'sinin cevabı bulunamayacak sorgularının da cevaplarını bulmakta yardımcı olmaktadır. Mesela bir kıtada bulunan ve yüksekliği 5000 metre üstünde olan dağların listesini Dbpedia Sparql sorgusu ile çok rahatlıkla öğrenilebilir.

Dbpedia metinlerden çıkardığı bilgileri “http://dbpedia.org/resource/” alanı altında biriktirmekte ve Vikipedi ile eşleşen başlıkları kullanmaktadır. Dbpedia ontoloji özelliklerini ise “http://dbpedia.org/ontology/” alanı altında biriktirmektedir. Vikipedi metinlerinden ve genelde "infobox"(metinlerin sağ tarafındaki özelliklerin tanımlandığı kutucuk) bilgilerinden özelliklerden veri çıkarımı gerçekleştiren sistem 320 sınıf altında 1650’den fazla özelliği tanıyabilmektedir. Ayrıca bu çalışmaları 28 farklı dilde gerçekleştirmektedir. En fazla entitileri olan sınıflar Çizelge 2.2’de yer almaktadır. Ayrıca Dbpedia diğer 30'dan fazla dış veri setleri ile ilişkilendirilmiş durumda olmakla “open linked data” projesinde önemli paya sahiptir.

Alternatif çalışmalar olarak “wikidata”, “freebase graf” veritabanı, “yago” platformu örnek gösterilebilir. Dbpedia çeşitli doğal dil işleme çalışmalarında kullanılabilir. Sorgu genişletme (query expansion), metin özetleme, belirsizlik çözümlemede ve aynı zamanda soru cevaplama ve veri çıkarımı projelerinde kullanılabilir.

Sonuç olarak, Dbpedia gibi amlamsal ağ veritabanının oluşumuinternetdünyasında en önemli veri kaynaklarından biri olan Vikipedi'den uzun ve kapsamlı bilgi çıkarımı çalışmaları sayesinde mümkün olmuştur ve ontolojinin oluşturulmasında dışarıdan

herhangi bir etkileşim olmamakla beraber SPARQL kullanılabilen internetarayüzü sayesinde ontolojilerin sorgulanması mümkün kılınmıştır.

Çizelge 2.2. Dbpedia anlamsal ağ alanında öne çıkan sınıflar ve içerdiği nesne sayısı bilgileri [21]

Sınıf	Nesne Sayı
Resource(toplam)	4233000
Place	735000
Person	1450000
Work	411000
Species	251000
Organisation	241000

Geliştirilen sistemde de Dbpedia temel veri tabanı olarak kullanılmaktadır.Kullanıcıdan girdi olarak kabul edilen doğal dil soruları çeşitli yöntemler kullanılarak SPARQL sorgularına dönüştürülmektedir ve Dbpedia üzerinden bu sorgular gerçekleştirilmekle sorulan bilgiye erişim sağlanmaktadır.

2.7. API, Vikipedi API

Uygulama programlama arayüzü (Application Programming Interface, kısaca API), bir yazılımın başka bir yazılımda tanımlanmış fonksiyonlarını kullanabilmesi için oluşturulmuş bir tanım bütünüdür. MediaWiki web API internetservisi olup "wiki"özelliklerine, servislerine ve bilgilerine sorgular gerçekleştirme olanağı sağlamaktadır. Bu API aracılığıyla, makaleler oluşturmak ve değişiklikler yapmak amacıyla tekrarlanan işlemleri otomatik ya da yarı otomatik gerçekleştirebilen "bot"lar oluşturulabilir, yönetilebilir, veri erişimi gerçekleştirilebilir. Aynı zamanda sisteme giriş yapma, makale yazmak veya değişikler imkanı sağlamaktadır. Servisin erişim noktası(endpoint) "http://en.wikipedia.org/w/api.php" adersinden erişilebilir [22].

Servis verileri "xml", "jsonfm", "JSON" gibi formatlarda sunmaktadır ve bu sistem çerçevesinde haberleşmede olduğu gibi, servis bilgilerinden veri çekimlerinde de JSON formatı kullanılmıştır.

Araştırma kapsamında kullanıcı sorusuna cevap bulunabilecek en uygun Vikipedi makalesinin belirlenmesiyle Dbpedia alanındaki eşleşen bilgiye de erişim sağlayacağından MediaWiki Search API'si kullanılmıştır. Aynı zamanda Türkçe makalelerin eşleşen İngilizce versiyonunun bulunmasında ve sonuçta bulunan İngilizce bilginin Türkçe karşılığının bulunmasında Vikipedia API'si sayesinde eşleşen makalelerin başlıkları bilgilerine erişimde kullanılmıştır.

2.8. NoSQL veritabanı sistemleri

NoSQL sürekli genişlemekte olan veri kaynaklarına geniş yelpazede farklı veri tabanı teknolojilerini kapsayan bir çözüm sunmaktadır. İlişkisel veri tabanlarına karşımodern yazılımlar için esnek ve zengin özellikleri ile daha hızlı performansa sahip veri tabanı yapıları sunmaktadır.

NoSQL veritabanı olan MongoDB üzerinde sistemin kullanmak istediği veri kaynakların tutulmasında kullanılmıştır. MongoDB, esnek, ölçeklenebilir ve oldukça hızlı çalışan doküman tabanlı veritabanıdır. Proje kapsamında Dbpedia ilişki özellikleri bilgileri ve aynı zamanda Türkçe etiket çevirileri MongoDB veritabanından sorgulatılmaktadır. Diğer taraftan WordNet veritabanı da JSON formatında aynı vertabanında farklı liste (collection) altında tutulmaktadır [23].

3 BENZER ÇALIŞMALAR.

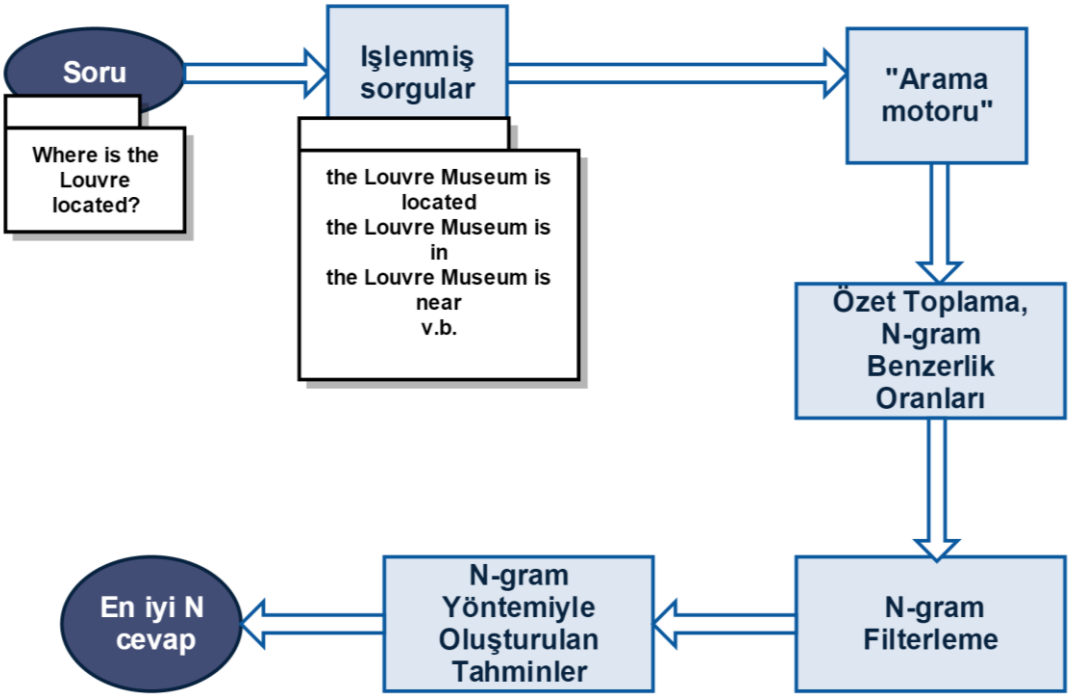
Bu bölümde soru cevaplama sistemlerinde izlenilen genel yaklaşımlara yer verilmiştir. Bunların yanı sıra literatürde daha önce yapılmış çalışmalardan bahsedilmiştir ve Türkçe üzerine yapılan önemli araştırmalara yer verilmiştir.

3.1. Genel Yaklaşımlar

Soru cevaplama sistemleri veri çıkarımı çalışmalarının özel bir türü olup, soruyu doğal dilde anlamak ve geniş alandaki kaynaklardan yola çıkarak uygun veri çıkarımı sonrası seçilen doküman ve sonrasında seçilen pasajlardan cevap üretmeyi hedeflemektedir. Yaygın olarak soru cevaplama sistemleri belli bir alan için değil, genel olarak bütün kaynaklar için tasarlanmaktadır. Bu durumda çeşitli istatistiksel ve dil işleme tekniklerine başvurulur. Soru cevaplamaya çalışan sistemler farklı yöntemlerden yola çıkarak genel olarak internetten veya ontolojilerden ve servislerden yararlanabilir [24]. Sistemin genel yapısı ise bazı ortak özellikleri temel olarak kendisinde bulundurmaktadır. Bunlar soru işleme, doküman işleme, cevap çıkarma ve cevabı uygun biçimde kullanıcıya ulaştırmak olarak listelenebilir.

3.1.1. ASKMSR

Sözcük Türü İşaretleme (POS-tagging), çözümleme, varlık ismi çıkarımı (named entity extraction), WordNet ve benzeri doğal dil işleme tekniklerini kullanmayan, internet kaynaklarından etkili faydalanmak üzerinde yoğunlaşan ASKMSR sistemi [24] ilk aşamada soruyu işlenmiş sorgulara dönüştürerek arama motorlarından doküman çekimi yapmaktadır. Daha sonra sayfa özetlerinden n-gram çıkarımları ve frekanslarına göre skorlama işlemi gerçekleştirir. TREC 2001 veri setinde 6. en başarılı sonuçlar verdiği bildirilmektedir. Platformun en önemli zayıf yönü, doğal dil işleme yöntemlerinin kullanılmaması, sadece arama motorundan dönen metinlerin biçimsel özelliklerinden çıkarımlar yapmasıdır. ASKMSR sisteminin genel iş akışı diyagramı Şekil 3.1'de verilmiştir.



Şekil 3.1. ASKMSR çalışmasında izlenilen yol [24]

3.1.2.AnswerBus

AnswerBus [25] 6 dile uygulanmış soru cevaplama sistemidir. Sorulan soruyu cevaplamak için dil kontrolü yapılmakta, İngilizce olmadığı durumlarda çeviri servisleriyle çeviri yapılmaktadır. Arama motorları sonuçlarından yola çıkarak cümle çıkararak cevap adayları oluşturuyor ve belli kriterlere göre sonuçlar listelemektedir. "Bag of words" tekniğiyle çalışan bu sistem ontoloji kaynaklardan faydalanmamakta ve ayrıca genelde tam cevap değil de, cevap seçimleri veya cevabı içeren metinleri sunmayı yeterli bulmaktadır. TREC-8'in 200 sorusu üzerinden yapılan denemeler sonucu başarılı sonuçlar elde etmiştir ve diğer çalışmaların sonuçları ile kıyaslamaları Çizelge 3.1 de gösterilmektedir .Çizelgede adı geçen sistemlerin ilk 5 cevap içinde doğru cevabı kapsayan ve ilk cevapta doğru sonuca ulaşan soru sayı, 5, 6, ve 7. sütunlarda uygun olarak sorulara harcanan en fazla, en az ve ortalama zamanın saniye cinsinden değerleri, 8. sütunda standart sapma değeri yer almaktadır. En son sütunda ise sorulan soruya dönen bilginin byte cinsinden uzunluğu gösterilmektedir.

Çizelge 3.1. AnswerBUS ve benzer çalışma sonuçlarının değerlendirme tablosu [25]

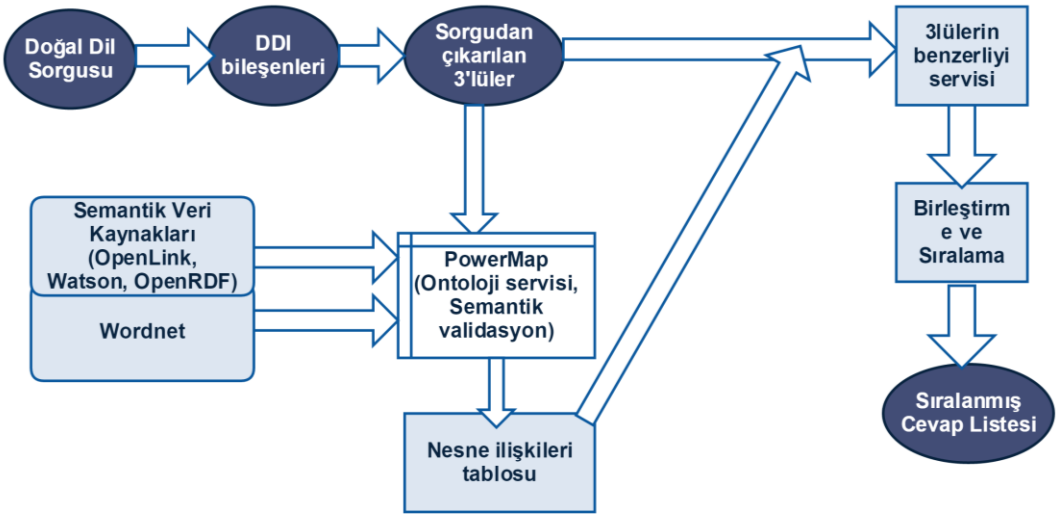
Sistemler	İlk 5 Doğru	En Doğru	NIST Skoru	Tmax	Tmin	Tmean	Tstd dev	Lmean (byte)
AnswerBus	141	120	64.18%	15.06	3.79	7.20	3.07	141
IONAUT				44.88	2.78	12.51	6.81	1312
LCC	97	75	41.73%	342.52	4.30	44.24	32.63	178
QuASM	13	7	4.45%	284.29	2.61	20.72	33.92	1766
START	29	29	14.50%	62.07	2.02	9.84	7.45	

3.1.3.PowerAqua

PowerAqua [26] doğal dildeki soruları anlayıp sınıflandırma ve benzerlik özelliklerinden faydalanarak, ontoloji sorgularında dönüştüren bir sistemdir. Bu sistem geniş yelpazede heterojen farklı kaynaklardan veri çıkarımı ve daha sonra sıralama yapararak gereksiz veriden temizlemektedir. Sistemin DDİ aracı DDİ sorgularını girdi olarak alıp özne-yüklem-nesne anlamsal üçlü çıktısı üretir. İkinci aşamada semantik kaynaklardaki uygun verilerde eşleşmeleri yapar. Son aşamada çıkarılmış veriler üzerinden birleştirme ve sıralama işlemleri gerçekleştirmektedir. Dbpedia semantik ağ veri kaynağından kullanım sistemin başarısını önemli derecede yükseltmiştir. Sistemin mimarisi ve kullandığı araçlar Şekil 3.2'deki gibi özetlenebilir.

3.1.4.Ginseng

Ginseng (A Guided Input Natural Language Search Engine) [27] doğal dil sorularını ontoloji kaynaklara ulaştıran sistem, kullanıcı sorularını otomatik tamamlamaya çalışır. Kural-tabanlı çalışan bu mekanizma sayesinde uygun görülmeyen sorular kısıtlanmaktadır. Sorulan soruyu RDQL sorgusuna çeviren sistem sonrasında Jena çatısı sayesinde ontoloji sorgusu gerçekleştirmektedir. Ginseng, bütün soruları çalıştıramamakla beraber sorular belirli alanları dikkate alarak çalıştırmaktadır. Diğer taraftan, sistem coğrafi alandaki soruların yaklaşık %40'lık kısmını tanıyabilmiştir.



Şekil 3.2. PowerAqua çalışmasında izlenilen yol 26]

3.1.5.Deanna

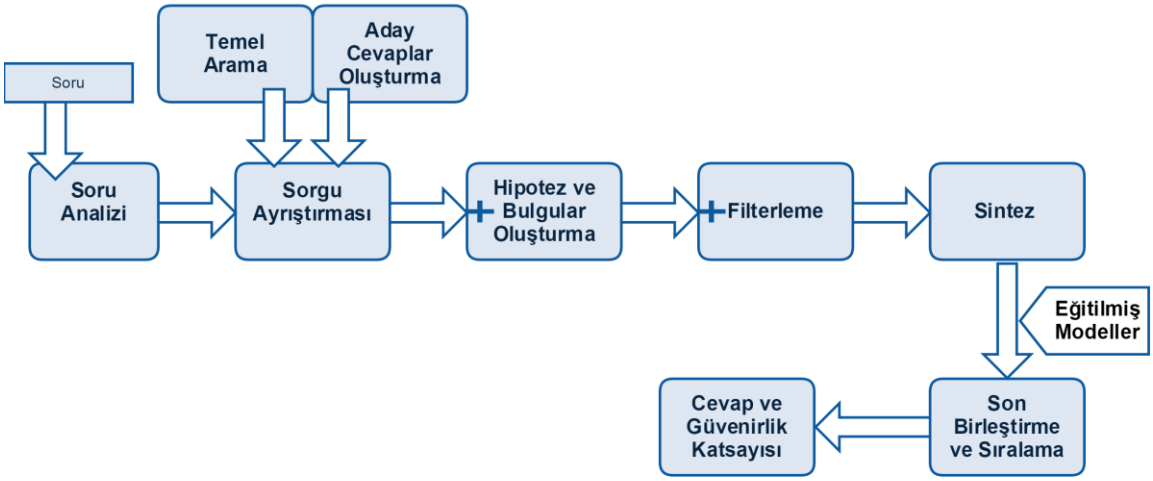
Yahya ve arkadaşları tarafından geliştirilen Deanna sistemi [28] Yago, Dbpedia ve Freebase gibi anlamsal veri kaynaklarından veri çıkarımı yapmaktadır. Daha çok kompleks sorular üzerinden çalışma yapmış sistem, sorularda isimleri nesne ve sınıflama, fiilsel kelime verilerini anlamsal ilişkilerle eşleştirerek, soruları makinaların anlayacağı şekle dönüştürür. "t filminde oynayan y asıllıaktör z'li yazarla ne zaman evlilik hayatına girdi?" tarzındaki karmaşık bir soruyu özne-yüklem-nesne tarzından birden fazla denklem oluşturarak, bilinmeyeni veri çıkarımları aşamasında sorgular sayesinde bulmaktadır.

3.1.6.Watson

“This is Watson” çalışması [29] IBM'in DeepBlue (satranç şampiyonu bilgisayar) çalışmasının devamı niteliğinde olup Jeopardy yarışmasına yönelik 3 yıl boyunca geliştirilen sistemdir. Oyundaki işleyiş üzerine daha çok ipuçlarından yola çıkarak neyin sorulduğunu ve hangi kelimelerden yola çıkarak veri çıkarımı yapacağını belirler. Sonraki aşamada “QClass” kullanılarak soru tipi kategorizasyonu ve leksik cevap tipi (LAT) yardımıyla bilgi çıkarımı işlemi yapmaya çalışır. Soru sınıflandırması ve ipucu, fokus ve LAT çıkarımı gibi işlemler için ek işlem gerekirse soruyu alt sorulara parçalayıp, bu soru parçalarını ayrıca inceleyebilir. Farklı veri çıkarımı algoritmaları ve skorlama

yöntemleriyle yüksek skorlu cevap adayları sentezlenir ve tahmin ve skor sistemi kullanılarak nihai cevap üretilir.

Veri çıkarımı zamanı mevcut Vikipedia data seti ve benzer kaynaklar kullanılmaktadır. Sistem çalıştığı sürece, veriler biriktirilmekte ve bir sonraki sorgularda önceki soru bilgilerinden faydalanarak performansını yükseltmektedir. Kompleks bir sistemden oluşan parçaların doğrudan birbiriyle bağlı olması performans ve başarı oranının yüksek olmasında etkili oluyor. Şekil 3.3 "This is Watson" sisteminin genel yapısını vermektedir.



Şekil 3.3. "This is Watson" çalışmasında izlenilen yol [29]

3.1.7.START Sistemi

START (SynTactic Analysis using Reversible Transformations) sistemi [30] doğal dil sorularını yanıtlamak için tasarlanmış bir yazılım sistemidir. START gelen soruları ayrıştırır, onları bilgi tabanıyla karşılaştırarak ayrıştırma ağaçlarından oluşturulan sorgular ile eşleştirir ve kullanıcıya uygun bilgi kesimlerini sunar. Bu şekilde, START kişiye bilgiye hızlı erişimesini eğitimsiz olarak sağlar.

START MIT Yapay Zeka Laboratuvarı'nda Boris Katz tarafından geliştirilmiştir. Şu anda, sistem Boris Katz liderliğindeki INFOLAB Grup tarafından daha da geliştirilmesi devam etmektedir. 1993 yılından beri çalışmaları sürdürülen START projesi ilk Geniş Dünya Ağı soru cevaplama projesi olarak kabul edilmektedir.

"Doğal dil şerhi" (natural language annotation) denilen bir yöntemle START sistemi bilgi arayanların bilgi kaynaklarına bağlanmasına yardımcı olur. Bu teknikle veri kaynaklarından doğal dil cümleleri, açıklamaları, özellik bilgileri farklı parçalar halinde çıkartılı ve soru girdisine eşleşen parçalardan bilgi çıkarımı gerçekleştirilir. Veri kaynaklarında diyagram, resim, video, audio, internet sayfaları ve benzeri formatlardaki yapılar incelemektedir. Çalışan sisteme "http://start.csail.mit.edu/start-system.html" adresinden ulaşılabilir.

3.1.8.Trueknowledge

Kullanıcı sorularını doğal dil işleme teknikleri kullanarak kendisi için daha sonra veri çıkarımında kullanabileceği sorgu şekline dönüştüren ve bu sorguyla kendi veri kaynaklarında veriyi çeker ve veri çıkarma yöntemleri ile oluşturduğu cevaplardan sonuçları üreterek kullanıcıya sunmaktadır.

Ek veri olarak kullanıcı sorgularından veri girişi sağlayan sistemin esas işleyişi 20000 sınıf kapsayan ontolojisiyle nesnelerin sınıflandırılması ve ilişkilerinin tanımlanmasının sağlanmasıdır. Bir bilgi veri tabanında belirli nesneler arasında tanımlanmış ilişkiler olarak yer almaktadır. Sorguları işleme sırasında doğal dil soruları sistemin anlayacağı dilde sorgulara dönüştürülür. Tanımlanmış yaklaşık 1500 kurala dayalı çalışan sonuç çıkarma yöntemleri ile var olan verilerden yeni veri çıkarımları yapmakta olan sonuç çıkarma modülü aynı zamanda hesaplama işlemlerini de gerçekleştirmektedir. Hangi bilgilerden sonuç çıkardığını bildiği için sistem açıklama çıktısını da cevapla birlikte sunmaktadır. Ayrıca, gerektiği zamanlarda resim gösterimi de sunmaktadır. Soru cevaplama ve kullanıcı girdilerinden ilave, tam farklı bir modül doğal dil işleme yöntemleri ile yapısal olmayan yazısal metinlerden cümle çıkarma, sadeleştirme, dönüştürme ve veri tabanına eklemek görevini gerçekleştirir [31].

3.2.Türkçe Soru Cevaplama Sistemleri

Türkçe soru cevaplama sistemleri çalışmaları daha çok biçimsel ve semantik veri işleme sonucu benzerlik ve örüntü tanıma algoritmaları üzerine yapılmış çalışmalardır.

3.2.1. BayBilmiş

Türkçe için 2002'de M.Fatih Amasyalı ve Banu Diri [32] tarafından tasarlanan ilk soru cevaplama olan BayBilmiş diğer benzer platformlar gibi soruyu sınıflandırmakta ve çözümlemekte, sonrasında arama motoru sorgusu haline getirmektedir. Arama motoru sonuç sayfalarındaki cümleleri filtreden geçirdikten sonra oluşan aday cümleler puanlandırma sistemine tabi tutulmakta ve en yüksek puanlı 5 cevap kullanıcıya iletilmektedir. Soru sınıflandırması zamanında,tanımlanmış 53 soru sınıf şablonu ve anahtar kelimeler sayesinde sorunun tipi belirlenmiştir. Arama motoru sorgusu zamanında 2 farklı yaklaşım izlenmiştir: özet bilgisinden cevap arandığında hızlı çalıştığı, linkten sayfaların kendisine giderek arama yaptığında yavaş olmasına karşı %16 daha başarılı olduğu saptanmıştır. Sistemde bulunan kelimelerin kendilerini, ayrıştırılmış hallerini ve görevlerini tutan farklı şablonların sayısının ve eşlemenin sonucu birebir etkilediği görülmektedir.

3.2.2. “Automatic Question Answering for Turkish with Pattern Parsing” Çalışması

Erbuğ Çelebi ve arkadaşlarının "Automatic Question Answering for Turkish with Pattern Parsing" çalışması [33] dokümanların örüntü tanıma ve çıkarılan özelliklerden faydalanarak soruları cevaplamaya çalışmaktadır. Haber kaynaklarından toplanılan düz metinlerden isim, ölçü, arazi, zaman, organizasyon gibi özellikler çıkarılarak veri tabanına kaydedilir. Sorulan soru işlendikten sonra neyin sorulduğunu çözerek veri tabanı sorguları ile veri çekimi gerçekleştirilir. Vektörel benzerlik algoritmaları yerine kelimeler ve anahtar kelimeler arası uzaklık bilgileri üzerinden yola çıkılmıştır.

3.2.3. Metin Madenciliği ile Soru Cevaplama Sistemi

Diğer benzer çalışma Metin Madenciliği ile Soru Cevaplama Sistemi Sevinç İlhan ve arkadaşları tarafından gerçekleştirilen yapısal olmayan düz veri metin bilgilerinin yapısal vektörel bilgilere dönüştürülmesi ve sonrasında kosinüs benzerlik hesaplanması kullanılan çalışmadır [34]. Sistemin iş prensibi gereksiz kelimelerin temizlenmesi, anahtar sözcüklerin seçilmesi, tf-idf ağırlıklandırılması ve benzerliklerinden yola çıkarak cevabın bulunması hedeflenmiştir.

3.2.4. "A Factoid Question Answering System Using Answer Pattern Matching" Çalışması

"A Factoid Question Answering SystemUsing Answer Pattern Matching" çalışması [35] "A pattern learning approach to question answering within the Ephyra framework" [36] çalışmasının Türkçeye uyarlanmış hali olup, NER kullanılarak başarı oranı daha da yükseltilmiştir. Sistem öncelikle soru ve cevaplardan oluşan veri kaynağıyla eğitiliyor. Daha sonraki soruyu cevaplama aşamasında arama motorunda cevap olabilecek metinlerden soru kısmının yerine geçebilecek cevap kısımları üzerinden cevaba ulaşmak hedeflenmektedir. Çalışma sonuçları Çizelge 3.2'de yer almaktadır. Çizelgede adı geçen yöntemlerin tek başına kullanılarak ve en son satırda bu yöntemlerin birlikte en başarılı kombinasyonunun sonuçları gösterilmektedir. Göründüğü üzere, kelime kökleri bilgisi beklenen performansı göstermemiştir ve en iyi sonuçlar NER kullanılarken elde edilmiştir.

Çizelge 3.2. A Factoid Question Answering System Using Answer Pattern Matching ve benzer çalışma sonuçlarının değerlendirme tablosu [35]

	MRR	Recall	Precision	Fmeas
Raw	0.28	0.24	0.57	0.34
RawAT	0.31	0.30	0.86	0.44
Stemmed	0.29	0.26	0.57	0.36
StemmedAT	0.30	0.29	0.88	0.44
NETagged	0.45	0.45	0.94	0.61
AllWithNE	0.58	0.56	0.86	0.68

3.2.5. Hazircevap

Son yıllardaki en kapsamlı çalışmalardan biri, TÜBİTAK destekli Hazircevap [37] isimli çalışmadır. IBM Watson projesinin Türkçeye uygulanması olarak görünen sistemde soruları cevaplamak için çeşitli aşamalarda veri işleme, DDİ, makina öğrenme algoritmaları uygulanmış ve kural-tabanlı metotlarla soruların analizi gerçekleştirilmiştir. Soru analizi ve sonrasında veri çıkarımından elde edilen dokümanlar üzerinden cevap adayları çıkarımı ve skorlama tekniklerinin uygalanmasından sonra, sonuç oluşturulur. Araştırmacılar, Türkçe kaynakların yeterli olmadığını, özellikle etiketli (tagged) soru/cevap kaynaklarının olmamasını birebir kıyaslama, değerlendirme yapılamaması, aynı zamanda

Türkçenin sondan eklemeli dil yapısından dolayı İngilizceye göre ek veri işleme gerektirdiği değerlendirmelerinde bulunmuşlardır.

3.2.6. Benbilirim

İlk aşamada soru sınıfı olarak zaman, insan, tanım, ölçek veya coğrafi soru tipine ait olması sınıflandırma yöntemleriyle belirlenir. Belirlenemeyen sorular genel soru olarak kabul edilir. Bağlaçlar edatlar ve stopwordlerden arındırılan soru kelimelerin kökleri üzerinden belirlenen sınıflandırma sistemi sonrasında fiilleri benzer fiillerle çaprazlama (multiplex) yaparak arama motorunda sorgulatıyor. Sonuç sayfalarından değil, özet bilgilerinden cevap araması gerçekleştirilmiştir. 108 soru üzerinden gerçekleştirilen sistemin tanım, ölçü ve coğrafi sorularda daha büyük başarı gösterdiği ve soru sınıflarına göre başarı oranları Çizelge 3.3'de görüntülenmektedir:

Çizelge 3.3. Benbilirim sonuçlarının değerlendirme tablosu [38]

Soru Tipi	Doğru	Yaklaşık	Yanlış	Toplam
Zaman	18	5	2	25
İnsan	15	3	5	23
Tanım	7	0	0	7
Ölçü-Değer	4	0	0	4
Coğrafya	36	2	2	40
Genel	9	0	0	9
Toplam	89	10	9	108

4. İNTERNET SERVİS VE KAYNAKLARI ÜZERİNDEN İŞLENMİŞ VERİ SAĞLAYAN SORU YANITLAMA SİSTEMİ

4.1. Giriş

İnternet kullanımındaki hızlı artış ve iletişim alanındaki gelişmeler, veri birikiminin artmasına neden olmuştur. Bu verilerden otomatik olarak bilginin çıkarımı önemli bir çalışma alanıdır. Bu alanda yapılan çalışmaların en önemlilerinden biri soru cevaplama sistemleridir. Soru cevaplama sistemleri, sorunun özelliklerine bakarak, önceden belirlenmiş belirli yöntemlerleinternet ve veri kaynaklarından doğru cevabı oluşturup kullanıcıya sunmayı hedeflemektedir. SCS bilgi alma (information retrieval), bilgi çıkarma (information extraction), doküman indeksleme/filtreleme, yapısal olmayan verileri yapısal verilere dönüştürme, benzerlik algoritmaları ve cevap oluşturma gibi pek çok önemli alanda çalışmaları gerektirmektedir.

Temelde kullanılacak yöntemler iki aşamalıdır.İlk aşamada kullanıcının sorusu DDİ yöntemleri ile incelenir ve sonraki aşamada ise istenen bilgilere bilgi erişim yöntemleri kullanılarak erişilir. Bu tezin konusu olan soru yanıtlama platformu internet servis ve kaynaklarını kullanarak elde ettiği bilgileri cevap çıkarma yöntemleri sonrası uygun biçimde kullanıcıya sunmayı hedeflemektedir. Platform geliştirilirken aşağıdaki adımlar dikkate alınmıştır:

1) Kullanıcı sorguları doğal dil işleme yöntemleri ile algılanacak ve incelenecektir.

2) Sistem sorgu ile ilgili internet kaynakları ve servisleri üzerinden yola çıkalarak cevaplanacaktır.

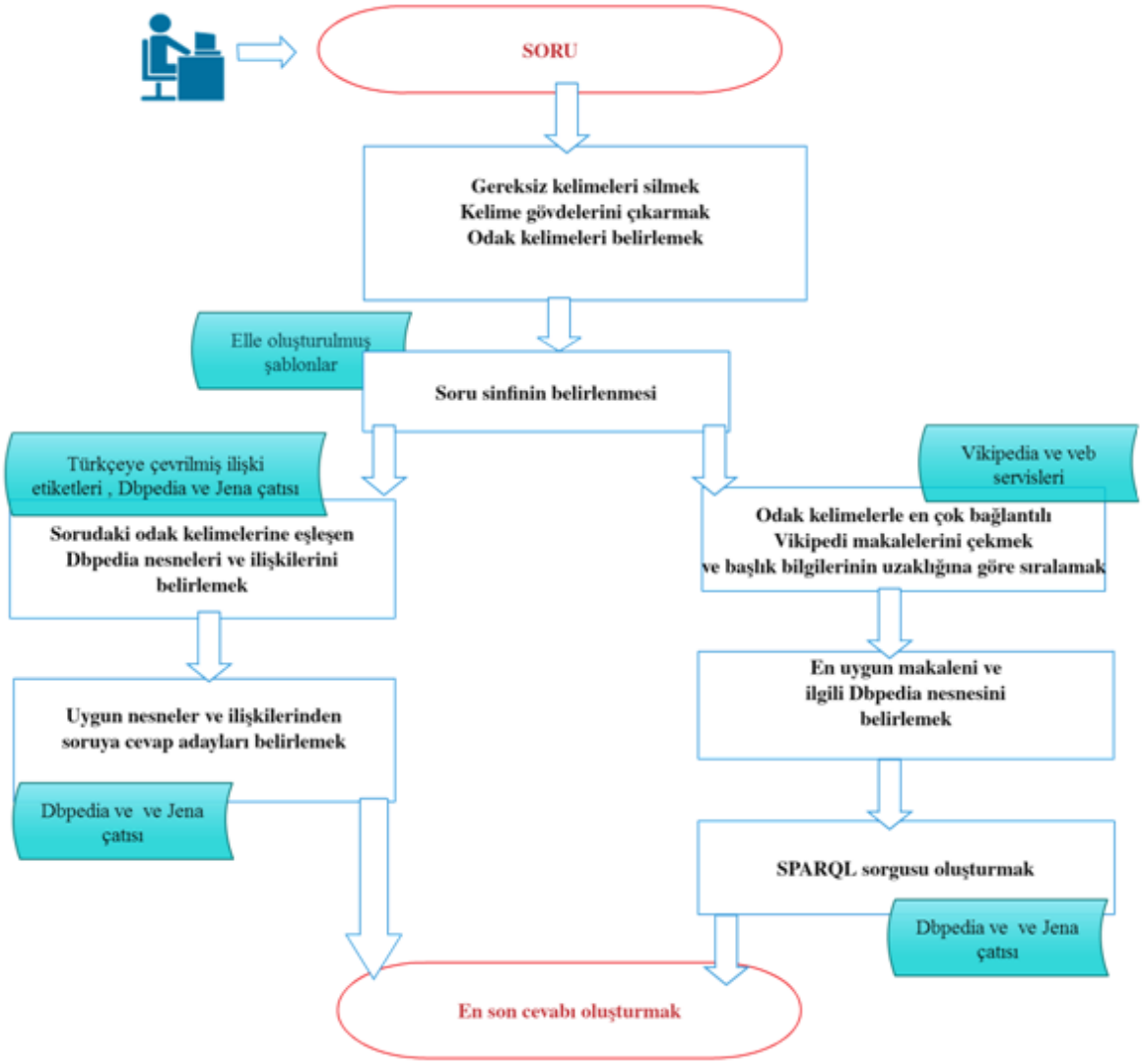
3) Değişik internet servis vekaynaklara erişimi sağlayacak esnek bir platform geliştirilecektir.

4) Veri doğrudan da elde edilebileceği gibi sonuçlar üzerinde işlemde yapmak gerekebileceği dikkate alınacaktır. Örneğin, bazı internet servisleri filmin ismine göre filmin hakkındaki bilgilere doğrudan erişilir. Bazen de arama sayesinde toplanan dokümanlardan yola çıkılarak istatistik yöntemlerle gereken veriye ulaşılmaya çalışılacaktır.

5) Daha sonra verinin tutarlılığı ve yeterliliğine dair çalışmalar yapılacaktır.

6) Başta belirli soru türleri ile çalışma yapılacaktır. Ama daha sonra ontolojiler kullanılarak daha esnek sorguları da cevaplayabilecek şekilde sistem büyütülecektir. İlk olarak günlük hayatta sorulabilecek karşımıza çıkılacak sorular düşünülmüştür.

İlk kısımda soru cümlesi işlenmesi gerekmektedir. İkinci kısımda neyin sorulduğu belirlendikten sonra uygun veri alanları bulunması ve neyin sorulduğu bilgisinden yola çıkarak farklı veri kaynaklarından cevap aratılması gerçekleştirilir. Bu açıdan anlamsal ağ teknolojilerinden, API'lerden veya normal internet sitelerindeki bilgilerden erişim seçenekleri vardır. API'ler ve anlamsal ağ kaynakları daha kesin veriye erişim açısından internet sitesi, makale tarzındaki kaynaklarla kıyasladığımızda daha kolaylıklar sağladığı tespit edilmiştir. Bu aşamada Türkçe anlamsal ağ kaynaklarının yeterli miktarda bulunması problemini ortaya çıkmaktadır.



Şekil 4.1. Soru Cevalama Sisteminde izlenilen yol

4.2. Soru Analizi Problemleri ve İzlenilen Çözüm Yolu

Genek iş akış diyagramı Şekil 4.1'de verilen sistem anlamsal ağ veri alanından veri çıkarımı gerçekleştirmeyi hedeflemekte olup ilk kısımda soruda neyin sorulduğunu anlamaya çalışmaktadır. Elle tanımlanmış şablonlar kullanılarak soru tipini ve neyin sorulduğunu belirledikten sonra Dbpedia veri alanından veri çıkarımı gerçekleştirmektedir. Soru tipini veya neyin sorulduğunu belirleyemediği durumlarda ise farklı bir yöntem izleyerek yine de anlamsal ağ alanı kullanılarak doğru cevaba yönelik tahminler oluşturmaya çalışmaktadır. Bu açıdan klasik Türkçe soru cevaplama sistemlerine göre farklı bir yaklaşım sergilemektedir.

Geliştirilen sisteminin ilk adımı soru cümlesi içerisinde yer alan kelimelerin açık kaynak kodlu Zemberek kütüphanesi [39] yardımıyla gövde ve eklerine ayrılması prosedürüdür. Çalışma sırasında genel olarak kelimeler olduğu gibi değil de, gövde bilgileri kullanılarak varsayımlar yapılmıştır. Zemberek kütüphanesi sayesinde elde edilen gövdeler, hem odak kelimelerin belirlenmesinde, hem de bir sonraki adımda örüntü tanıma yöntemiyle soru sınıflandırma işleminde kullanılmaktadır. Gövde bilgilerinden yola çıkarak sorulan sorunun hangi soru sınıfı içinde yer aldığı belirlenmezse, yani önceden tanımlanan soru şablonlarından hiçbiri ile eşleşmezse, genel soru olarak nitelendirilerek, farklı bir yöntemle cevabı bulmaya çalışılacaktır.

Soru sınıfları elle tanımlanmıştır ve her soru sınıfı için belirli şablonlar oluşturulmuş olup, Dbpedia veri kaynağında da verilerin sınıflandırılmış ontolojisiyle (http://mappings.dbpedia.org/server/ontology/classes/ adresinden Dbpedia'daki ontoloji sınıflarının hiyerarşik şekline erişilebilir) eşleşmektedir. Dbpedia ontolojisinden en çok nesnesi olan sınıfların listesi 2014 istatistiklerine göre Çizelge 2.2'de yer almaktadır. Bu soru sınıflarının en çok kullanılan soru tipi örüntüleri ve kelime gövdeleri bilgilerinden yola çıkarak belirli soru şablonları oluşturulmuştur. Türkçe sondan eklemeli dil olduğundan kelimelerin gövdeleri üzerinden benzerliklerinbulunması daha uygun çözüm yolu olarak kabul edilmiştir. Bazen kelime tek birgövdeden oluşurken, bazen de bu gövdeye bağlı bir veya daha fazla ekten oluşur. Bu kuraldan dolayı çok fazla sayıda yeni biçimdekelimeler oluşturulabilinir. Böylece ekler sayesinde farklı biçimlerde görünen kelime farklılıkları ortadan kaldırılarak ve her biçim için farklı soru tipi şablonu geliştirilmesine gerek kalmamakta, var olan şablonlar esneklik kazanmaktadır. Kelimelerin

gövdelerinin bulunmasında açık kaynak kodlu Türkçe DDİ Kütüphanesi Zemberek kullanılmıştır.

İlk aşamada geliştirilen ve testi yapılan soru sınıfları aşağıdaki listede yer almaktadır: YazarKitabı, ÜlkeÖzellikleri, İnsanYaşı, YayınTarihi, DoğumTarihi, MeslekSorusu, KitapYazarı ve benzerleri.

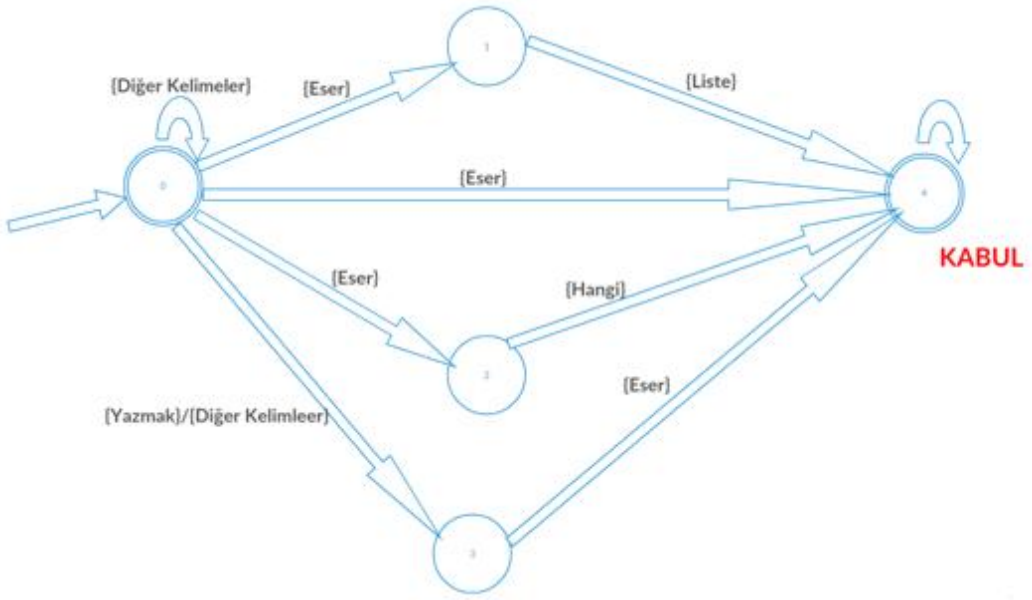
Soru analizi kısmında soru tipleri belirlenmesi aşamasında ikinci yapılan iş, soru tipine göre gereksiz kelime listesi tanımlanması aynı zamanda soru tipi kapsamında örüntü şablonlarının tanımlanmasıdır. Mesela YazarEser sorusuna yönelik olarak “en ünlü” ve “ünlü” gibi çok kullanılan ve analiz işlemiiçin çok da anlam ifade etmeyen sıfatların ve bağlaçların, edatların dikkate alınmamasına karar verilmiştir. Örüntü biçimi tanımlanması işleminde bu kelimelerden arınmış kelime topluğundan analiz kısmı devam ettirilmiştir. Her soru tipine uygun olarak farklı örüntü şablonları oluşturulmuştur. Mesela YazarKitabı soru tipi için, aynı sorunun 7 farklı tipte sorulabilme ihtimali ele alınmıştır (köşeli parantezler içinde değişebilen kelime gösterilmiştir):

"[YAZAR]ın kitaplarını listeleyin."
"[YAZAR] hangi kitapları yazdı?"
"[YAZAR]ın yazdığı kitaplar"
"[YAZAR]'ın yazar olduğu kitaplar"
"[YAZAR]'ın kitaplarının listesi"
"[YAZAR]'ın en ünlü kitapları"
"[YAZAR]'ın en ünlü kitapları hangileridir?"

Bir sonraki işlem olarak, farklı kelime türleri grupları tanımlanmıştır. Mesela YazarKitabı soru tipi için, "kitap", "roman", "eser" gibi kelimeleri {eser} kelime grubu içinde, "yazmak", "kağıda dökmek","kaleme almak" tipindeki fiil kelimeleri{yazmak} ailesi içinde yer almaktadır. Mesela fiil olarak kullanılabilecek “yazmak” tipli kelimelerin topluluğunu aynı zamanda işleyebilmek aşamasında tek bir fiil değil, eşanlamlı, yakın anlamlıkelimeleri de dikkate almak gerekmektedir.

Sorunun tanımlanmış şablonlarla eşleşmesi problemine çözüm olarak sonlu durum makinesi şeklinde çözüm üretilmiştir. Sonlu durum makinesi veya sonlu durum otomata olarak tanımlanan sistem sınırlı sayıda durumdan, durumlar arasında var olabilecek

geçişlerden ve bu zaman ortaya çıkacak sonuçları tanımlamak için kullanılan sistemdir. Sorudaki kelimelerin gövdelerini bulduktan sonra örüntüleri tanımlamak ve sonrasında soruların tanımlanmış otomata örüntüleri ile eşleşip eşleşmediğine karar verilmesi gerekmektedir. Kelimelerin gövdeleri üzerinden değerlendirmeler yapıldığı dikkate alınırsa, YazarKitabı soru tipi için X +{eser}+{liste}/{listelemek}, X+{yazar}+{eser}, X+{eser}+{soru kelimesi} gibi örüntüler tanımlanmıştır."X hangi kitapları yazmıştır?" şeklinde bir soruyu tanıyabilmek için {X soru-kelimesi} {yazı-anlamlı kelime} {yazmak tipinde kelime} şeklinde bir örüntü tanımlanıyor ve daha sonrasında soru sorulduğunda soru bu şekilde biçimlenmişse, YazarKitabı soru sınıfı tipinde soru olduğuna karar verilir. YazarKitabı sınıfı için Şekil 4.2 deki otomata tanımlanmıştır.



Şekil 4.2. Sınıf belirlenmesi sırasında kullanılan örnek örüntü şablonu

YazarKitabı soru şablonlarına uyan sorular için sistem sorulan şeyin yazarın kitapları olduğunu anlayacaktır. Durumlar üzerinden değerlendirmiş olursak, "KABUL" durumagelindiği zaman sorunun şablonun soru sınıfıyla eşleştiği görülür. Dolayısıyla, soru diğer şablonlarla eşleşmeyecektir ve soru analizi kısmının sonucu olarak, sorunun sınıfı bulunmaktadır. Eğer sistem bu sınıftaki belirtilen örüntülere uymadığını görürse, geride kalan diğer sınıf şablonları üzerinden devam ederek sonu sınıfını bulma çalışmasına devam edecektir. Soru sınıfının bulunması aynı zamanda hangi bilginin, yani hangi ilişkibilgisinin sorulduğunu bulmak anlamına gelmektedir.

Bu aşamada tek soru sınıfı belirlenmiyor, ayrıca sorudaki odak kelimeler de belirlenmiş oluyor. Mesela YazarKitabı soruları için genelde ilk kelime veya kelimeler odak kelimeler

olarak belirlendiği otomata diyagramından veya şablonlardan kolayca görülebilir. Diğer taraftan sonraki aşamada kelime özellikleri soru ve kelime özelliği çıkarma işlemlerinin büyük kısmı işlenmiş oluyor. Örnek soru üzerinden konuşursak, "Orhan Pamuk'un kitaplarını listeleyin" tarzında bir sorunun soru analizi kısmında "kitaplarını" sözünün eser grubuna, "listeleyin" sözünün de "liste" grubuna ait olduğunu sistem tanımaktadır. "Orhan Pamuk'un" kısmından odak kelimeleri seçilerek birleştirilecek ve "Orhan Pamuk" çıkarımı odak kelime grubu olarak belirlenmektedir.

Örüntü tanımlamayı biraz daha geliştirip Yazar_Eser sorularını daha başarılı ve kolay tanıyabilmesi için WordNet kelime ailesi platformundan fayda sağlanmıştır. WordNet daha önce de anlatıldığı üzere kelimelerin ve kelime ilişkilerin tutulduğu bir kütüphanedir. Soru örüntülerini tanımada eşanlamlı, yakın anlamlı kelime gruplarını oluşturmamızda "[x kitabı kimin kaleminden çıkmıştır]" tarzındaki yakın anlamlı kelimelerle ifade edilmiş soruları da belirlemekte fayda sağlamaktadır. Örneğin, "Orhan Pamuk'un yazdığı kitaplar" yerine "Orhan Pamuk'un kaleme aldığı kitaplar" diye sisteme sorulduğunda da sistem aynı sonuçları verecektir. Platform için kullandığımız WordNet çalışması Türkçe için geliştirilmiş ve kelime ilişkilerinin tanımlı olduğu yaklaşık 17000'den fazla ilişkinin tanımlı olduğu platformadır [10].

WordNet için tanımlanan sisteme girdi olarak, "yazmak" fiilini tanıttığımızda sonuç olarak, "verb_group", "synonym" gibi ilişkilerden benzer kelime listesi çıkarımı gerçekleştirilir ve sistemde bu gibi kelimelerle karşılaşıldığında "yazmak" sözü gibi kabul edilerek yapılan çıkarımlar sonrası işlemlere devam edilmektedir.

Soru cevaplama prosedürünün ilk aşaması soru analizi kısmında, soru şablonları ve onları sınıflandıran sistem geliştirilirken soruların çok farklı biçimlerinin olduğu dikkate alınmış, bu tip örüntülerle tüm soruları ele alamayacağı için tanımlanmayan soru şablonları için de daha genel ve esnek çözüm yolu sunulmuştur. Veri çıkarımı adımlarında sistemin soru sınıfı belirlenemediğinde izlediği yol detaylı şekilde anlatılacaktır.

4.3. Veri Çıkarımı

Soru analiz kısmından sonrakiişlem uygun Vikipedi makalesinin bulunmasıdır. Bu adımdaanahtar kelimelerden yola çıkılarak Vikipedi arama API'si kullanılmaktadır. "Orhan Pamuk hangi kitapları yazmıştır?" sorusundan "Orhan Pamuk" anahtar kelimeleri

bulunmaktadır. Sorguyu ayrıştırılmış şekilde ele alındığında aşağıdaki gibi görünmektedir:

```
http://tr.wikipedia.org/w/api.php?
action=query &
srlimit=20   &
list=search  &
format=json  &
srsearch=Orhan%20Pamuk
```

Yani "Orhan Pamuk" girdisi için en yakın 20 cevabın JSON formatında listesi istenmektedir. Geri dönen cevap (İlk 5 cevap ayrıntılı şekilde gösterilmiştir) aşağıdaki gibidir.

{

- **warnings**: {
  - **query**: {
    - *****: "Formatting of continuation data has changed. To receive raw query-continue data, use the 'rawcontinue' parameter. To silence this warning, pass an empty string for 'continue' in the initial query."

  }

},

- **batchcomplete**: "",
- **continue**: {
  - **sroffset**: 20,
  - **continue**: "-||"

},

- **query**: {
  - **searchinfo**: {
    - **totalhits**: 303

  },

  - **search**: [
    - {
      - **ns**: 0,
      - **title**: "Orhan Pamuk",
      - **snippet**: "Ferit OrhanPamuk (d. 7 Haziran 1952, İstanbul), Türk yazar. Birçok başka edebiyat ödülünün yanı sıra 2006 yılında Nobel Ödülünü kazanarak bu ödülü alan",

- **size**: 22972,
- **wordcount**: 1880,
- **timestamp**: "2015-07-21T13:57:37Z"

},

- {
  - **ns**: 0,
  - **title**: "Gizli Yüz (film)",
  - **snippet**: "Gizli Yüz, senaryosunu OrhanPamuk'un yazdığı, yönetmenliğini Ömer Kavur'un yaptığı 1991 yapımı Türk filmidir. Pamuk senaryoyu Kara Kitap'taki "Karlı",
  - **size**: 3086,
  - **wordcount**: 76,
  - **timestamp**: "2015-02-15T16:43:02Z"

},

- {
  - **ns**: 0,
  - **title**: "Yeni Hayat (roman)",
  - **snippet**: "Yeni Hayat, OrhanPamuk'un romanı. "Bir gün bir roman okudum ve hayatım değişti" cümlesiyle başlar. Üniversite öğrencisi bir genç, kentten kente otobüs",
  - **size**: 899,
  - **wordcount**: 25,
  - **timestamp**: "2014-08-09T17:13:35Z"

},

- {
  - **ns**: 0,
  - **title**: "Kar (roman)",
  - **snippet**: "Kar, OrhanPamuk'un ilk baskısı 2002 yılında İletişim Yayınları tarafından yayınlanan romanı. On iki yıldır Almanya'da sürgün olan şair Ka Türkiye'ye",
  - **size**: 2231,
  - **wordcount**: 134,
  - **timestamp**: "2014-12-23T18:01:18Z"

},

- {
  - **ns**: 0,
  - **title**: "Saf ve Düşünceli Romancı",

```
            ▪ snippet: "Saf ve Düşünceli Romancı, Nobel ödüllü
              Türk yazar <span
              class="searchmatch">Orhan</span><span
              class="searchmatch">Pamuk'un</span> 2011 yılı Eylül
              ayında ilk baskısı İletişim Yayınları tarafından
              piyasaya sunulan, yazarın",
            ▪ size: 1393,
            ▪ wordcount: 57,
            ▪ timestamp: "2012-08-21T13:32:32Z"

          }

        ]

        //Diğer 15 JSON objesi

    }

}
```

Bu liste üzerinden Levenstein uzaklık algoritması uygulanarak sıralanmaktadır ve sıralanmış liste üzerinden tam eşleşen başlık olup olmadığı kontrol edilmektedir.

```
[
{Orhan Pamuk=0},
{Beyaz Kale=8},
{Kara Kitap=9},
{Babamın Bavulu=9},
{Kar (roman)=10},
{2005=11},
{Öteki Renkler=12},
{Öküz (dergi)=12},
{Manzaradan Parçalar=13},
{Gizli Yüz (film)=15},
{Yeni Hayat (roman)=15},
{Masumiyet Müzesi=15},
{Benim Adım Kırmızı=16},
{Kafamda Bir Tuhaflık=16},
{İletişim Yayınları=16},
{Gizli Yüz (senaryo)=17},
{Columbia Üniversitesi=19},
{Cevdet Bey ve Oğulları=20},
{Saf ve Düşünceli Romancı=21},
{İstanbul: Hatıralar ve Şehir=24}
```

]

Yukarıdaki örnekte ilk başlık için Levenstein uzaklığı 0'a eşit olduğundan ek skor fonksiyonu çalıştırılmadan uygun makaleyi bulma işleminin başarıyla sonuçlandığı kabul edilmektedir. Eğer Levenstein uzaklığı 2'nin üzerinde olursa, yani birebir uyan Vikipedi makale başlığı bulunmadığı durumlarda, odak kelimeler üzerinden varsayımlar yapılmakta, odak kelimeleri içeren başlıklar değerlendirilmektedir. Odak kelimelerin belirlenmesinde ilk başta Zemberek kütüphanesinde tanımlanmış özel isimler listesinde yer alıp almadığı kontrol edilmekte, soruda özel isimler belirlenmediği durumlarda varolan kelimelerden özel isim olabilecek kelimeler üzerinden işlemler gerçekleştirilmektedir.

Yukarıdaki örnekte Vikipedi'de en çok uyan makale başlığın "Orhan Pamuk" olduğuna karar verildiğinden sıradaki işlem Vikipedi makalesinin İngilizce versiyonunun başlığının bulunması gerekmektedir. Bu işlem yaparken Vikipedi API'si kullanılarak ve aşağıdaki sorgu sonucu makalenin diğer dillerdeki başlıklarının listesi elde edilmektedir.

http://tr.wikipedia.org/w/api.php?action=query&format=json&prop=langlinks&lllimit=500&titles=Orhan%20Pamuk

Vikipedi API'si yukaradaki Vikipedia sorgusu için aşağıdaki cevabı döndürmektedir:.

{
- **llcontinue**: "102064|eu",
- **continue**: "||"

- **query**: {
  - **pages**: {
    - **102064**: {
      - **pageid**: 102064,
      - **ns**: 0,
      - **title**: "Orhan Pamuk",
      - **langlinks**: [
        - {
          - **lang**: "an",
          - *****: "Orhan Pamuk"},
        - {
          - **lang**: "ar",

- *****: "قوم اب ن اه روأ"},
- {
  - **lang**: "az",
  - *****: "Orxan Pamuk"},
- {
  - **lang**: "ba",
  - *****: "Орхан Памук"},
- {
  - **lang**: "be",
  - *****: "Архан Памук"},
- {
  - **lang**: "bn",
  - *****: "ওরহানপামুক"},
- {
  - **lang**: "br",
  - *****: "Orhan Pamuk"},
- {
  - **lang**: "bs",
  - *****: "Orhan Pamuk"},
- {
  - **lang**: "ca",
  - *****: "Orhan Pamuk"},
- {
  - **lang**: "ckb",
  - *****: "رهانپاموکۆئ"},
- {
  - **lang**: "cs",
  - *****: "Orhan Pamuk"},
- {
  - **lang**: "cv",
  - *****: "Орхан Памук"},
- {
  - **lang**: "cy",
  - *****: "Orhan Pamuk"},
- {
  - **lang**: "de",
  - *****: "Orhan Pamuk"},

- {
  - **lang**: "el",
  - ***:** "Ορχάν Παμούκ"},
- {
  - **lang**: "en",
  - ***:** "Orhan Pamuk"},
- {
  - **lang**: "eo",
  - ***:** "Orhan Pamuk"},
- {
  - **lang**: "es",
  - ***:** "Orhan Pamuk"},

//ve benzerleri

]}}}}

Vikipedi İngilizce makale başlığı belirlendiğinden ("Orhan Pamuk") sonuç olarak Dbpedia rdf-store alanının da otomatik olarak belirlendiği söylenebilir. Bundan başka soru tipinin YazarEser olduğunu, dolayısıyla sorulanın yazar eserleri olduğu bilindiğinden uygun aşağıdaki Sparql sorgusu şeklinde ifade edilebilir:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbpedia: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://dbpedia.org/property/>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT DISTINCT ?x2 WHERE {
       ?x0 rdf:type foaf:Person.
       ?x1 foaf:name ?x2.
       ?x1 rdf:type dbpedia-owl:Book.
       ?x0 rdfs:label "Orhan Pamuk"@en.
       ?x1 dbpedia-owl:author ?x0.
}
```

Bütün veri paylaşımlarında JSON yapısı kullanıldığından veri çıkarımı sonrası geri dönen cevap Çizelge 4.1'deki gibi olmaktadır.

Yukarıdaki örnekte şablonlarla tanımlanmış soru üzerinden gidildiğinde hangi işlemler yapıldığı, hangi senaryonun izlendiği anlatılmıştır. Sistem soru şablonunu ve

soru sınfını belirleyemediği durumlarda ise sistem daha farklı, daha genel durumları inceleyebilecek bir yol izlemesi gerekmektedir. Soru sınıfı belli olduğunda dolayısıyla sorulan ilişki tipi de belirlenmiş olmaktaydı ve soru tipine uygun Sparql sorgusunda sadece bilinmeyenin değerini bulmakla veri tabanında kayıtlıysa, veriye ulaşılması sağlanmaktaydı. Ama soru sınıfı bilinmediği durumlarda, sorguda odak nesneden ilave ilişki tipinin de çözülmesi gerekmektedir ve bu durumda örüntü tanımadan farklı bir ilişki belirlemek prosedürü de gerçekleştirilmelidir.

Çizelge 4.1. Sparql sorgusu sonucu

Kitaplar
"The Museum of Innocence"@en
"Masumiyet Müzesi"@en
"The White Castle"@en
"Beyaz Kale"@en
"My Name Is Red"@en
"Benim Adım Kırmızı"@en
"Benim Kırmızı"@en
"Kara Kitap"@en
"The Black Book"@en
"Yeni Hayat"@en
"The New Life"@en
"Kar"@en
"Snow"@en

İlişki belirleme prosedürü incelendiği zaman, elde olan kaynaklar, yani sorudaki anahtar kelimeler ve işleme sırasında bulunan Vikipedi makale başlığı bilgisi, ayrıca diğer yardımcı kelimeler üzerinden ilişki belirlenmesi çalışması yapılmalıdır. Diğer dikkat edilecek durum Dbpedia'da ilişki bilgileri İngilizce etiketlerle tanımlanmıştır.

Bu aşamada sistem öncelikle tanıya ve çözebilmesi açısından sisteme ilişkileri tanıtmak gerektiğinden Dbpedia'da tanımlı özelliklerin listesi Sparql sorgusu ile veri çekimi sonrası Nosql veri tabanına kaydedilmiştir. Nosql veritabanı olarak Mongodb kullanılmıştır. İlişkisel veri tabanlarına karşı Nosql-Mongodb tercihinin önemli avantajları olarak performansta başarılı olması ve verileri veri haberleşmesinde kullanılan JSON yapısındatutmasıdır. Sonuç olarak veri tabanından veri çekimi sonrası veri işlemesi zamanı sorgulanan bilginin ayrıca JSON formatına dönüştürülmesine gerek kalmamaktadır. İlk

önce Dbpedia'dan sorgulanan ilişkilerin tüm listesi veri tabanına kaydedilmiştir. Daha sonra Bing Translate API kullanılarak ilişki belirten kelimeler Türkçe'ye çevrilmiş, kelime eşleşmelerinde başarı oranını daha da yükseltmek adına gövde bilgileri de veri tabanına eklenmiştir. Bundan başka Türk Dil Kurumunun (TDK) sağladığı güncel Türkçe sözlüğünden ayrıştırılan Türkçe kelimlerin anlam ifadeleri de veritabanına ilerideki çalışmalarda başarı oranının yükseltilmesi potansiyeli dikkate alınarak eklenmiştir. Bu veriler sorudaki kelimeler ve Dbpedia ilişkileri tutulan veri tabanından uygun ilişki kelimesini bulmak için kullanılmaktadır. Dbpedia'dan çekilmiş 60000'e yakın ilişki etiketi veri tabanına eklenmiştir ve örnek bir semantik ilişki verisi asağıdaki gibi tanımlanmaktadır.


```
{
  "_id" : ObjectId("554bf20a371ffea4eced44cc"),
  "tr" : "yazar",
  "value" : "writer",
  "xml:lang" : "en",
  "type" : "literal",
  "url" : "http://dbpedia.org/property/writer",
  "lemma" : "|||+yazar+yaz+yaz+yazar+yazar",
  "explanation" : [
    "1. isim Bilim, edebiyat, sanat alanlarında kitap yazan veya kitap
        hazırlayan, bir eseri ortaya koyan ve eserin sahibi olan kimse, kalem
        erbabı, müellif \"Her tarih eseri, doğrudan doğruya veya dolaylı olarak
        yazarın hayat tecrübesine bağlıdır.\" - C. Meriç",
    "2. Özellikle gazete ve dergilerde herhangi bir konuda yazı yazan kimse,
        kalem erbabı, muharrir",
    "3. sıfat Yazma özelliği olan",
    "|||"
]
}
```

Soru sınıfının belirlenemediği bir soru analizi zamanı odak kelimeler ve odak olmayan kelimeler grupları farklı gruplar altında incelenmektedir. Odak kelimeler üzerinden Vikipedi makalesi seçilmektedir. Odak kelimesi olmayan kelime grupu ise ilişki tipini belirlemek için incelenmektedir. Örneğin, "Orhan Pamuk hangi kitapların yazarıdır?" sorusunda soru sınıfı ve şablonunun önceden tanımlanmadığı bir durum olarak mercek altına alındığında veya "Türkiye Cumhuriyetini kurucusu kimdir?" sorusundaki özel isimler odak kelimeler olarak belirlenecektir. Bu durumda “Orhan Pamuk” ilk soru için ve “Türkiye” ikinci soru için odak kelimeler olarak belirlenecektir.İkinci sorudaki "kurucu" kelimesi ise Dbpedia'dan çekilmiş ilişki etiketleri içinde yer aldığından ilişkiyi tanımlayan kelime olarak sonraki ilişki üzerinden cevap çıkarma işleminde kullanılacaktır. Dbpedia'da soruyla ilgili nesne bilgisini bulmak için kullandığımız Vikipedi API'sine sorgu parametresi olarak ise odak kelimelerverilmektedir. Soru analizi kısmında "Orhan Pamuk" odak kelimeler grubunda yer almaktadır, "kitap", "yazar" kelimeleri ilişki belirlenmesinde ele alınmaktadır. “Orhan Pamuk” Vikipedi makale başlığı ve Dbpedia semantik ağ bilgilerinden alan adı belirlenmesinin devamında ise ilgili nesnenin tüm ilişkilerinin listesi Sparql sorgusu sonucunda çekilen veriden elde edilmektedir. Diğer soru örneğinde ise "Türkiye" kelimesi odak kelimesidir ve sistem aşağıdaki Sparql sorgusuyla Dbpedia'dan Türkiye nesnesinin 391 ilişki bilgisini içeren listeyi geri döndürmektedir. Bu ilişki bilgilerinin bir kısmı Çizelge 4.2'de verilmiştir.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dbpedia: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://dbpedia.org/property/>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT * WHERE {
        <http://dbpedia.org/resource/Turkey> ?property ?value.
        OPTIONAL {?property rdfs:label ?x.}
        OPTIONAL {?value rdfs:label ?y.}
         FILTER( langMatches(lang(?x), "")  || langMatches(lang(?x), "EN" ) )
        FILTER( langMatches(lang(?y), "")  || langMatches(lang(?y), "EN" ) )
}
```

Çizelge 4.2. Dbpedia'da Türkiye'yle ilgili bilgilerin bir kısmı

Ilişki	Değer	İlişki etiketi	Değer etiketi
http://dbpedia.org/property/largestCity	http://dbpedia.org/resource/Istanbul	"largest city"@en	"Istanbul"@en
http://dbpedia.org/ontology/governmentType	http://dbpedia.org/resource/Unitary_state	"government type"@en	"Unitary state"@en
http://dbpedia.org/ontology/leader	http://dbpedia.org/resource/Ahmet_Davuto%C4%9Flu	"leader"@en	"Ahmet Davutoğlu"@en
http://dbpedia.org/ontology/leader	http://dbpedia.org/resource/Cemil_Çiçek	"leader"@en	"Cemil Çiçek"@en
http://dbpedia.org/ontology/leader	http://dbpedia.org/resource/Recep_Tayyip_Erdo%C4%9Fan	"leader"@en	"Recep Tayyip Erdoğan"@en
http://dbpedia.org/ontology/timeZone	http://dbpedia.org/resource/Eastern_European_Time	"time zone"@en	"Eastern European Time"@en
http://dbpedia.org/property/caption	http://dbpedia.org/resource/Ahmet_Davuto%C4%9Flu	"Caption"@en	"Ahmet Davutoğlu"@en
Ve benzerleri			

Aynı zamanda Dbpedia ilişki etiketlerinin Türkçe karşılıkları sistemin veri tabanında yer almaktadır. Mesela soruda odak kelimesinin "Türkiye" olarak belirlendiği durumda Dbpedia 391 ilişkisinin yer aldığı liste sunmaktadır ve veri tabanından Türkçe karşılıkları sorgulatıldığında aşağıdakı liste elde edilmektedir:

[hizalama, demonym, gsyih ppp , gini, hdi, latd, latm, latns, longd, longew, longm, bilinmeyen_ilişki(n), wikipage harici bir sayfaya bağlantı, alan büyüklüğü, nüfus yoğunluğu km, nüfus yoğunluğu sırası, nüfus yoğunluğu sq mi, nüfus tahmini sırası, hdi kategori, sermaye, hükümet türü, içi km, alan sırası, alanı sq mi, kodu çağırma, sermaye, resim yazısı, seviye alan, ortak adı, geleneksel uzun ad, para birimi, para birimi kodu, tarih biçimi, yön, sürücülerde, kuruluş tarihi, kurulan olay, etnik gruplar, kayan nokta, dipnot bir, nominal gsyih, kişi nominal gsyih , nominal gsyih yıl , kişi başına düşen gsyih ppp , gsyih ppp yıl , gini değiştir, gini sırası, gini yıl, hükümet türü, hdi değiştir, hdi sırası, hdi yıl, görüntü, görüntü ceket, görüntü bayrağı, görüntü eşlem, en büyük şehir, lider adı, lider

başlık, sol, istiklal marşı , resmi diller, yüzde su, nüfus sayımı, nüfus sayımı yıl, sağ, egemenlik tipi, sembol türü, saat dilimi, saat dilimi dst, başlık, başlık çubuğu, utc farkı, utc farkı dst, web sitesi, genişlik, küçük resim]

Bu ilişkilerden hangisinin sorudaki ilişki bildiren kelimeye uygun geldiğini belirlemekle RDF üçlüsünün üçüncü tarafını yani sorulan kısmı Dbpedia alanından sorgulanmaktadır. Dbpedia özellik listesine internetarayüzünden de "dbpedia.org/page/Turkey" adresini erişmek mümkündür. Sıradaki işlem soruda geçen kelimeler ve bu ilişkiler arasında eşleşme yaparak **yüklem-ilişki-nesne** üçlüsünden sorunun cevabı olacak üçüncü kısmı (nesne) bilgisini çıkarma işlemidir. Bu zaman Türkçe ilişki etiketi listesinden birebir erişen ilişki varsa sorulan ilişki olarak kabul edilerek üçlünün üçüncü tarafı da cevap nesnesi olarak kabul edilmektedir. Bulunmadığı durumda en yakın seçimler ve değerleri kullanıcıya sunulmaktadır.

Anlatılan örneklerden de göründüğü gibi, sistemin bu modüle gelindiğinde kelimelerin birbirleriyle ilişkilendirilmeleri ve bu ilişkilerin türleri belirlenmiştir ve daha çok yapısal veri olarak ele alınmaktadır. Veri çıkarımı için yapısal olmayan çeşitli internet verilerinden değil de, semantik alanda çeşitli çalışmalarla semantik ağ verilerine dönüştürülmüş belirli veri kaynaklarından yola çıkmak kolaylık ve hız sağlamaktadır. Buna ek olarak aynı zamanda veri heterojenliğini önlemiş olmaktadır. Bu aşamada başka bir kısıtlama daha vardır. Sistemin semantik ağ veri kaynağı olarak kullandığı Dbpedia'da Türkçe derlenmiş bilgi yeterli miktarda değildir ve dolayısıyla İngilizceyle eşleştirilmiş veriler de yeterli olmamaktadır. Ayrıca diğer semantik ağ veri kaynakları da Türkçe için kullanışlı değildir. İkinci problem ise Dbpedia'da belirtilmiş ilişkiler ve sorularda sorulan ilişkiler arasında eşleştirme gerekmektedir. Dbpedia'da sorgulamalar Türkçe üzerinden aratıldığında yeterli kadar veri çıkarımı yapılamadığı tespit edilmiştir. İngilizce veri çıkarımı gerçekleştirildiğinde ise sonuç olarak elde ettiğimiz veri sayısal ve tarihsel bulguları dikkate almazsak, sonuçlar genel olarak İngilizce olarak geri gelmektedir. Özellikle soru sınıfı belirlenemediği durumlarla bu problem daha çok dikkat çekmektedir.

Bu problemlerin çözümünde farklı bir senaryo işlenmiştir ve sistem sorudaki anahtar kelimeler üzerine odaklanılmıştır. Odak kelimeler olarak da nitelendirebileceğimiz bu kelimeler için aday olarak ilk başta özel isimler üzerinden yola çıkılmıştır. Özel isimlerin belirlenmediği sorularda alternatif yol olarak özel isim olmadığı kesin kelimeler, yani fiiller ve soru kelimeleri elenerek odak noktası olma ihtimali yüksek olan kelimeleri

potansiyel odak kelimeler olarak kabul edilmiştir. Bu kelimelerden uygun Vikipedi başlıkları API sayesinde belirlenmiştir ve ilgili Vikipedi makale başlıkları üzerinden Levenstein uzaklık algoritması kullanılarak sıralama yapılmıştır. Bundan başka sıralamanın yalnız biçimsel özellik yani harf farkıyla belirlenmesinde hata payının olabilme ihtimalini göz önünde bulundurarak ek olarak Vikipedi makaleleri üzerinden de yola çıkarak metin uygunluğu kontrolü çalışmalarıyla başarı oranının yükseltilmesi ilerideki çalışmalarla sağlanabilir. İlgili makalelerin bulunmasında, aynı zamanda Vikipedi makaleleri üzerinde işlemler gerçekleştirilirken Vikipedi arama API'si kolaylık sağlamaktadır. Vikipedi arama API'si girdi olarak kabul ettiği kelimelere uygun başlıklardan yola çıkarak sorguda istenen en yakın 20 başlık geri döndürülmüştür. Eğer başlık ve odak kelimeleri birebir uyuyorsa o makale üzerinden işlem gerçekleştirilmektedir. Aksi durumda en yakın 20 seçenek üzerinden sıralama yapılmakta ve en uygun başlık seçilir. İkinci aşamada ise, amaç Vikipedi makalesine eşleşen nesneyi Dbpedia anlamsal ağ alanında bulmak olarak görünmektedir. Dbpedia özellik çıkarma ve nesne oluşturma sırasında izlediği yoldan faydalanarak uygun metni bulmak zor değildir. Ama Dbpedia'daki Türkçe kaynağın yeterli olmaması sebebiyle İngilizce var olan bilgi kaynağından yola çıkmamız gerekmektedir. Bunun için Türkçe makalenin İngilizce versiyonunun başlığını bulmak gerekir. Genelde makalenin URL'si kısmında "tr" kısmını "en" olarak değiştirdiğimizde Vikipedi sitesi uygun makalenin İngilizce versiyonuna yönlendirmeye çalışmaktadır. Platformda işlemi API aracılığıyla geliştirerek aynı makalenin İngilizce versiyonuna ulaşılmaktadır. Bundan sonraki kısım artık nesneyi Vikipedi'deki İngilizce makale başlığı veya URL'si bilgisinden yola çıkarak bulabildiğimiz için Dbpedia alanından veri çekimi işlemi ilişki bilgisinden yola çıkılarak gerçekleştirilebilir. Dbpedia anlamsal ağından veri çekimi zamanı SQL dilini anımsatan Sparql sorgulama dili kullanılmaktadır. Java programlama dili üzerinden işlemi gerçekleştirmek için Jena çatısı kolaylık sağlamaktadır.

Sparql ilişkisel veri tabanlarından veri çekimi sırasında kullanılan SQL sorgularıyla benzerlik göstermektedir. SQL sorgulamadaki "where" şartına benzer “filter”, "top result" listesine uygun "limit" gibi eşdeğer ifadeler bulunmaktadır. Ayrıca "use schema" ifadesinin benzeri olan "prefix" tanımlamalarla sorguları hangi anlamsal ağ veri alanlarından ve ilişkilerinden yola çıkılarak çalıştırılacağı belirtilmektedir.

Dbpedia'da veriler diğer anlamsal ağ veri bilgileri ile ilişkili olduğundan, ilişkilerden yola çıkarak diğer veri alanlarından da veri çekimi gerçekleştirilebilir. Coğrafi bir alan

sorgulatıldığında, “geospatial” veri alanı ile ilişkisinden ve geospatial'ın bize sunduğu coğrafi bilgi sistemi API'si kullanılarak haritada göstermek imkanı da bulunmaktadır. İlerideki çalışmalarda bu özelliklerle sistem daha geniş alanlara taşınabilir.

Dbpedia'dan veri çekimi prosedürü gerçekleştirilirken soru analizi zamanı öne çıkan bulgular da dikkate alınmaktadır. Örneğin, soruda bir filmin yönetmeninin kim olduğunun sorulduğunu farz edildiğinde, soru cümlesinde kullanılan soru kelimesi "kim" sorulan nesnenin "person" sınıfı nesnesi olduğunu belirtmektedir ve soru sınıfı belirlenemediği durumlarda, {film_adı}+{yönetmek}+{yönetmen} ilişkini, yani kısacası yönetmek ilişkisini çözemese bile sistem, film_adı bilgisinden Dbpedia nesnesini belirledikten sonra ilişkisi olan nesnelerden "person" sınıfı üyesi olduğunu bildiğinden bazı çözümler sunabilir. Bu da adı geçen film bilgileri üzerinden soru cevaplamaya çalıştığımızda cevap olabilecek ilişkili nesnelerin listesini daraltmaktadır. İlişkisi bulunan "person" nesneleri listesinde aktörler, soundtrack bestecisi, kostüm dekoru edenler gibi belirli bilgiler üzerinden daha dar alanda bulmamıza olanak sağlayacaktır. "Filmin yönetmeni kimdir" tarzı soruda sistemin işi daha da kolay olacaktır. Dolayısıyla, cümleden ilişki bilgisi kolaylıkla çıkarılabiliyorsa, ek bilgiler kullanmadan direk sorgulamadan cevaba ulaşmamız gerçekleşmektedir. İlişki bulunmadığında, direk "yazarı kimdir" sorulmamış, "kimin kaleminden çıkmıştır" tarzında sorulmuşsa eğer, sistem önce WordNet'le kim yazmıştır tarzında basit hale indirgeme gerçekleştirmeye çalışacaktır ve gerçekleştiremezse dolayısıyla, şablonlarda bulunamadığı durumlarda, kitap nesnesinin ilişkili olduğu nesnelerin listesini sunulacaktır. Ama insan olabilecek birden fazla ilişkili nesne olacağından ek bir işleme de gerek bulunmaktadır. Veya başka bir örnekte, şablonlarda tanımlanmayan bir ilişki bilgisi sorulduğunda, yine şablonlarda tanımlanmadığı düşünülürse, sorudaki kelimelerden ilişki bilgisini belirlemek gerekmektedir.

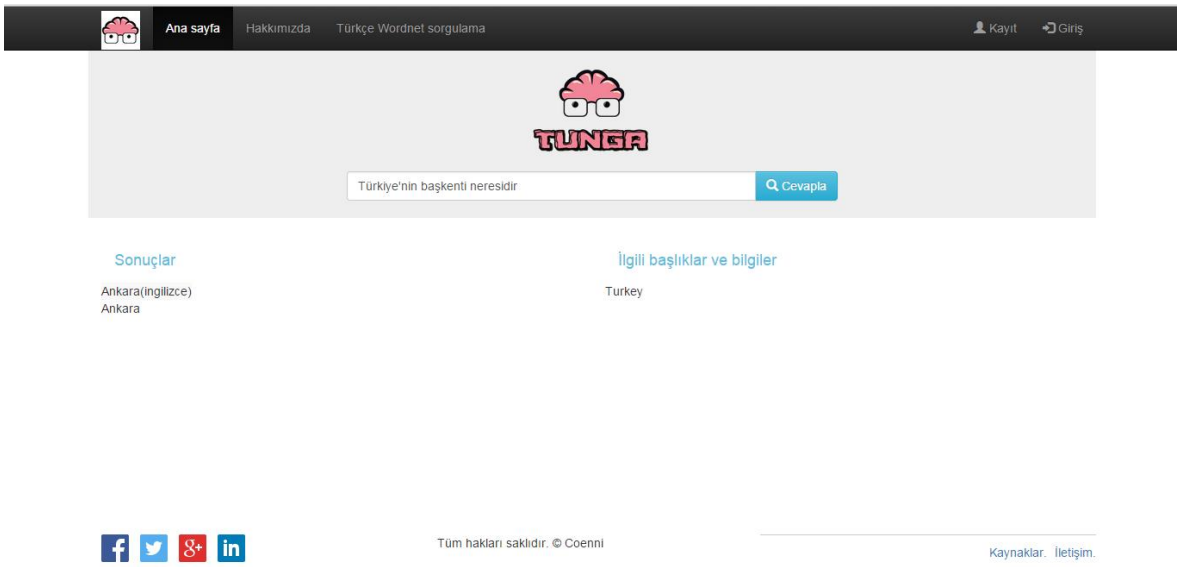
WordNet, Zemberek gibi yardımcı bileşenlerin veya diğer Türkçe DDİ kütüphanelerinde bu özellik olmadığından problem için farklı bir çözüm yolu öne sürülmüştür. Sistem en başta oluşturulduğunda ilişkilerin listesi Dbpedia bilgilerinden sorgulatılmış, ilişkiler listesinin Türkçeye çevrilmesi işlemi gerçekleştirilmiştir. Ve soru cümlesindeki ilişki tanımlanmasında kullanılan kelimelerle nesnenin ilişkilerinin Türkçe'ye çevrilmiş kelimeleri ile eşleştirmeye çalışmaktadır. Eşleşen ilişki etiketi bulunduğu zamanı üçlünün üçüncü nesnesi cevap olarak kabul edilmektedir.

İlerideki çalışmalarda kullanılmak adına TDK'ın kelime anlamı sözlüğü kullanılarak veritabanında ilişkilerle birlikte anlam bilgilerinden de fayda sağlayarak başarı oranının yükselmesi adına farklı çalışmalar da gerçekleştirilebilir.

Sonuç olarak, soru analizi ve veri çıkarımı modüllerini incelendiğinde, analiz ile başlayan süreçte ve sonrasında semantik ağ veri alanlarından bilgilerin çekilmesi ve işlenmesiyle devam eden ve sonuç çıkaran sistem Türkçe dil yapısını da dikkate alarak kompleks işlemler gerektirmektedir. Problemin çözüme ulaşmasında belli kısıtlamalar vardır ki, bunların başında Türkçe yeterli miktarda kullanılabilir verinin özellikle semantik ağ verisinin bulunmamasıdır. Bundan başka Türkçenin sondan eklemeli dil olması İngilizce geliştirilen sistemlere göre daha fazla işlem gerektirmektedir. Diğer bir başka husus DDİ kütüphanelerinin İngilizce için yazılan sistemler kadar dakik ve zengin olmamasıdır. Türkçe WordNet platformı da 25 yıllık İngilizce WordNet'le kıyaslanacak kadar gelişmiş duruma henüz gelmemiştir. Vikipedi'de Türkçe makale sayısının İngilizceyle kıyaslandığında az olduğu bilinmektedir. Bundan başka Dbpedia İngilizce nesne sayısı da çok daha fazla işlendiğinden Dbpedia'daki Türkçe veriler yeterli olmamaktadır. İleride Dbpedia kaynağının Türkçe alanının genişlemesi sonucu, direk Türkçe alanı üzerinden de çalışmalar yapılması olanakları ve başarı oranının yükselmesi öngörülmektedir.

4.4. Değerlendirme ve Bulgular

Sistem, diğer arama sistemlerinde de tercih edilen basit bir arayüzü ile tasarlanmıştır. Platformun teknolojik altyapısına genel olarak değinirsek, sistem internetprojesi olarak geliştirilmiş ve arayüzün tasarlanmasında "Bootstrap CSS"çatısı kullanılmıştır. Arayüz teknolojisi olarak kullanılan bu çatı RESPONSİV özelliğinden dolayı farklı ekran boyut ve çözünürlüklerine uyum sağlamaktadır. Kullanıcıdan alınan sorgunun sunucu tarafa yönlendirilmesi zamanı AJAX sorgularla gerçekleştirilmiştir. Ajax sorgularla sunucuya ulaşan soru metni soru metni analizi ve veri çıkarımı sunucuda Java programlama dili kullanılarak geliştirilmiştir. Java dili, bu dilde yazılan en kapsamlı Türkçe dil işleme kütüphanesi olan Zemberek'ten faydalanmak imkanı sağlamaktadır. Sistemin görsel arayüzü Şeki 4.3'deki gibidir.



Şekil 4.3. Çalışmanın görsel arayüzü

Eğer kullanıcı tarafından tanımlanan soru ayrıştırılan sorudan soru sınıfı veya sorulan özellik bilgisi belirlenemezse platform anlamsal ağda odak kelimesiyle ilgili genel bilgiler sunmaktadır (Şekil 4.4).

Sistemin değerlendirilmesinde daha önce kullanılmış soru seti [35] kullanılmıştır. bu aşamada sisteme 30 kitap yazarı sorusu, 30 ülke başkenti sorusu, 30 doğum tarihi sorusu, 30 ölüm tarihi sorusu sorulmakla sistemin ne kadar başarılı olduğu hesaplanmıştır. Soruda farklı teknikler kullanılmakla kullanılan metotların sisteme nasıl etki ettiği incelenmiştir ve gelecek çalışmalara yönelik varsayımlar yapılmıştır. Sisteme sorulan 120 sorunun 110

sorunu cevaplayabilmiştir, bunlardan 81'i doğru cevap olmaktadır. Sorulan bilginin ne kadar yaygın bilinen bilgi olması sistemi de cevaplayabilme niteliğini etkilemiştir.

Şekil 4.4. Örnek bir nesnenin belirlenen ilişkilerinin bir kısmı

Daha çok Türkiye Edebiyatı ile ilgili sorulan "X kitabının yazarı kimdir?" tipindeki sorularda kitap makalelerinin Vikipedi sitesinde ayrıca makalede değinilmediği sorularda veya makalenin İngilizce alternatifi olmadığı sorularda sistem zorlanmıştır. Buna karşılık dünya edebiyatının ünlü kitaplarının, romanlarını anlatan hem İngilizce, hem de Türkçe makaleler bulunduğundan sistem dünya edebiyatına ilişkin soruları kolaylıkla cevaplayabilmektedir. Bu test verisine göre sistemin performansı Çizelge 4.3.'de görülebilir.

Özet olarak sistem ilk olarak internet servisleri aracılığıyla uygun Vikipedi makalesini belirlemeye çalışmaktadır ve sonrasında eşleşen İngilizce makale üzerinden Dbpedia anlamsal ağ alanında nesneyi belirlemektedir. Sonraki aşamada soruda sorulan ilişkinin nesne için tanımlanan ilişki etiketleriyle eşleştirme işlemini gerçekleştirmektedir. Eşleşen üçlü bilgisinden cevap kısmı elde edilmekte ve uygun biçimde kullanıcıya sunulmaktadır.
Problemin çözüme ulaşmasında belli kısıtlamalar vardır ki, bunların başında Türkçe Vikipedia makalelerinin sayısının İngilizce makalelere göre çok daha az olmasıdır ve devamında bu prosedürün Dbpedia veri alanına da etkilediği bellidir. Bundan başka Dbpedia'da Türkçe veri alanının küçük olması yanında İngilizce ile eşleşmeleri (canonicalization) de azdır. 2014 istatistiksel verilere göre, İngilizce nesne sayısı

4,584,616, Türkçe makalelere dayalı nesne sayısı 233,737'dir, hem İngilizce, hem de Türkçe kanonik formu olan nesne sayısı ise 143,914'dür.

Çizelge 4.3. Sistemin120 soruluk test kümesindeki performans sonuçları

ST	#S	#C	#D	K	SP(MRR)
Yazar sorusu	30	26	6	23%	23%
Başkent sorusu	30	30	21	70%	73%
Doğum tarihi sorusu	30	28	28	100%	93%
Ölüm tarihi sorusu	30	26	26	100%	86%
Toplam	120	110	81	73%	68%

Kullanılan veri alanının yeteri kadar geniş olmaması kısıtlama olarak görülse de, diğer yönden soru cevaplama sistemlerinde karşılaşılabilecek veri heterojenliği problemi yoktur ve bilgiler rastgele internet sayfalarından değil, dünyaca bilinen ve kabul gören bir veri kaynağından veri çıkarımı gerçekleştirilmiştir. Ayrıca, yapısal olmayan verileri yapısal veriye dönüştürme, düz metin verilerinden veri çıkarımları yaparak belli bir alanda biriktirme gibi veya önceden sistemin eğitilmesi gibi ek prosedürler gerçekleştirmeye gerek duymamaktadır.

Kullanılan veri kaynağının geniş olmaması bir kısıtlama olarak görülse de, arama motoruyla elde edilen veri kaynakları gibi esnek bir veri setiyle karşı karşıya değiliz. Ayrıca, belli bir zaman sonra klasik çözümlerde başarı oranında çok bir değişiklik olmamakta, bu makalede tercih edilen sistem ise veri alanının sürekli genişlemesinin sistemin soru cevaplama başarısında önemli derecede artışlar öngörülmektedir.

İngilizce soru cevaplama sistemleri ile genel olarak karşılaştırıldığında, verinin özellikle semantik ağ verisinin eksikliği, bundan başka Türkçenin sondan eklemeli dil olması ve bunun İngilizce geliştirilen sistemlere göre daha fazla işlem gerektirmesi bulguları ortaya çıkmaktadır. Ayrıca arama servisleri genelde biçimsel özelliklerden yola çıktığından biçimsel farklılık gösteren benzer anlamdaki kelimeleri farklı kelimeler olarak değerlendirmektedir, İngilizce'de bu problem Türkçedeki kadar önemsenmeyebilir. Diğer bir başka husus DDİ kütüphanelerinin İngilizce için yazılan sistemler kadar dakik ve zengin olmamasıdır. Aynı zamanda, Türkçe WordNet platformı da 25 yıllık İngilizce WordNet kadar kapsamlı değil.

Yukarıda dokunulan kısıtlamalar ve eksiklikler ileride adı geçen çalışmalardaki iyileştirmelersistemin ve tercih edilen yöntemlerin başarı oranının da yükselmesine katkı sağlamış olacaktır. Ayrıca Türkçe alanı için Dbpedia çatısının geliştirilmesi halinde sistemine daha kolaylıkla sonuçları elde etmesi öngörülmektedir. Buna ek olarak sistemde cümle içinde kelimeler arasındaki ilişkileri belirleyen cümle çözümleyicilerin kullanılmasıyla örüntü tanımadan başka bir metot da izlenilerek daha genel kapsamda soru cevaplayabilmek yeteneği kazandırılmasında alternatif seçim olarak değerlendirilebilir.

5. SONUÇ VE TARTIŞMA

Geliştirilmiş platform Türkçe soru cevaplamada çok az sayıda çalışmadan biri olup var olan anlamsalağ bilgilerinden veri çıkarımı yapmakla farklı bir çözüm yolu sunmuştur. Verilerin düz metinlerden DDİ yöntemleri kullanarak değil, hazır sistemler kullanılarak geliştirilmesi sonuç odaklı daha başarılıperformansa ulaşmaya yardımcı olmuştur. Soru şablonu örüntülerinden veya izlenilen diğer yöntemlerle de sonuca ulaşamadığında bile, soru analizi sonucu elde ettiği bulguları değerlendirerek yakın cevaplar, ilgili bilgiler üretmeye çalışmaktadır. Dbpedia'nın diğer veri alanları ile ilişkisinden yola çıkarak sistemin ek bilgiler sunmasını sağlamak, örneğin bir arazinin konumu sorulduğunda harita üzerinde gösterimi gibi çalışmalarla sistem daha kapsamlı hale getirilebilir. Bu yönde bazı denemeler gerçekleştirilerek sistemin esnek yapısı sayesinde başarılı olduğu tespit edilmiştir.

Araştırma tasarımında soruyu veya sorulanı algılamak kısmında veya daha sonra veri erişimi safhasında geliştirilmiş araçlar sistemler, özellikle Türkçe için geliştirilmiş sistemler incelenmiştir. Platform içinde Türkçe metin işlemede kolaylık sağlayacak araçlar dikkate alınmıştır.Bu kapsamda WordNet, Zemberek gibi araçların platformda kullanılabileceği kanaatine varılmıştır.

Platform işleyişi kapsamında süreç işleyiş tarzı olarak ilk kısımda kullanıcıdan soruyu veya sorulanı sormaktır. Eğer tanımlanan şablonlara uyacak şekilde bir şey soruluyorsa, platform açısından algılamak daha kolay olacaktır. Başka durumlarda da sistem kendisi varsayımlar üretmeye çalışmaktadır. Örneğin, Azerbaycan'ın başkenti olarak girdi olursa, program direk Azerbaycan'ı zaten başkent kelimesine göre ülke olması kanaatine gelir ve ülke bilgileri ve özellikleri bilgileri sağlayan alanlara yönelir. Eğer soru biçiminde sorulursa, örneğin "Afrikalı Leo romanını kim yazdı?" tarzında soru ile karşılaşılırsa, sorunun algılanması gerekmektedir. Romanın ismi, "x romanı" tamlamasından yola çıkılarak bulanabilir. Soru türünü belirlemek için de "x'ı kim yazdı?" gibi şablonlar ile de soru analiz edilebilir. Girdi olarak girilen sorunun işlenerek bu hale getirilmesi ve veri kaynakları aracılığıyla denklemlerdeki gibi x="Amin Maalouf" cevabı bulunması ve kullanıcıya cevap olarak sunulması gerekmektedir.

Geliştirilen platform anlamsal ağ projesi gibi doğal dil sorularını algılayan ve uygun SPARQL sorgularına çevirerek Dbpedia servisi üzerinden cevaplandıran bir sistemdir.

Doğal dil işleme yöntemlerimden faydalanarak algılanan sorular, tanımlı örüntüler sayesinde sınıflandırılmıştır. Soru cümlelerinde geçen kelime grupları ile Dbpedia ontolojisinde tanımlı olan sınıflara uygun çevrilmesi ve eşleşmeyen durumlarda doğru cevaba götürecek yakın cevaplar listesi sunmaktadır. Geliştirilen platform tüm doğal dil sorularını cevaplayacak kapasitede olmasa bile, elde ettiğimiz test sonuçları umut verici ve ileriye yönelik çalışmalar için teşvik edici olmuştur. Geliştirme olarak, daha fazla soru sınıfları tanımlanarak sistemin daha fazla soru çeşidine cevap vermesi iyileştirme hedefleri arasında gözükmektedir. Ayrıca birden fazla internet servis ve kaynaklarından faydalanmak, daha kompleks doğal dil sorularını cevaplayabilmek için çalışmaların yapılması hedeflenmektedir.

6. KAYNAKLAR

[1] O'Luanaigh, Cian. *Highlights from CERN in 2013.* **2013**.

[2] Frakes, W. B. *"Introduction to information storage and retrieval systems."* .**1992**.

[3] Bush, Vannevar. "As we may think." *interactions* 3, no. 2, 35-46, **1996**.

[4] White, Howard D., and Belver C. Griffith. "Authors as markers of intellectual space Co-citation in studies of science, technology and society." *Journal of Documentation* 38, no. 4,255-272 , **1982**.

[5] Callan, James P., W. Bruce Croft, and John Broglio. "TREC and TIPSTER experiments with INQUERY." *Information Processing & Management* 31, no. 3, 327-343, **1995**.

[6] Kwok, Cody, Oren Etzioni, and Daniel S. Weld. "Scaling question answering to the web." *ACM Transactions on Information Systems* 19, no. 3, 242-262, **2001**.

[7] Athenikos, Sofia J., and Hyoil Han. "Biomedical question answering: A survey." *Computer methods and programs in biomedicine* , 1-24, **2010**.

[8] Ittycheriah, A., Franz, M., Zhu, W. J., Ratnaparkhi, A., & Mammone, R. J. . "IBM's Statistical Question Answering System." *TREC.* **2000**.

[9] Fellbaum, Christiane. *WordNet.* Blackwell Publishing Ltd, **1998**.

[10] Bilgin, Orhan, Özlem Çetinoğlu, and Kemal Oflazer. "Building a wordnet for Turkish." *Romanian Journal of Information Science and Technology* 7, no. 1-2, 163-172, (**2004**).

[11] Ayşe, Şerbetçi, Orhan Zeynep, and Pehlivan İlknur. "Extraction of semantic word relations in Turkish from dictionary definitions." *Proceedings of the ACL,* **2011.**

[12] El-Kahlout, Ilknur Durgar, and Kemal Oflazer. "Use of wordnet for retrieving words from their meanings." *Proceedings of the global WordNet conference.*118-123, **2004**.

[13] *http://www.*json.org, erişim tarihi: 03.11.2014.

[14] Berners-Lee, Tim, James Hendler, and Ora Lassila. "The semantic web." *Scientific american* 284, no. 5, 28-37, (**2001**).

[15] Guha, Ramanathan, Rob McCool, and Eric Miller. "Semantic search." *Proceedings of the 12th international conference on World Wide Web.* ACM, **2003**.

[16] Cardoso, Jorge, and Amit P. Sheth. *Semantic web services, processes and applications.* Vol. 3. Springer Science & Business Media, **2006**.

[17] http://www.w3.org/TR/rdf-schema/, erişim tarihi: 14.04.2015.

[18] Hebeler, John, ve b. *Semantic web programming.* John Wiley & Sons, **2011**.

[19] Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., & Domingue, J. *Enabling semantic web services: the web service modeling ontology.* Springer Science & Business Media, **2006**.

[20] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. *Dbpedia: A nucleus for a web of open data .* Springer Berlin Heidelberg, **2007**.

[21] *http://wiki.dbpedia.org/news/dbpedia-version-2014-released,* erişim tarihi: 02.08.2014.

[22] *http://www.mediawiki.org/wiki/API:Main_page,* erişim tarihi: 02.08.2014.

[23] *http://www.mongodb.com/nosql-explained,* erişim tarihi: 02.08.2014.

[24] Brill, E., Dumais, S., & Banko, M. "An analysis of the AskMSR question-answering system." *In Proceedings of the ACL-02 conference on Empirical methods in natural language processing.* 257-264, **2002**.

[25] Zheng, Z. "AnswerBus question answering system." *Second international conference on Human Language Technology Research.* Morgan Kaufmann Publishers Inc, 399-404, **2002**.

[26] Lopez, Vanessa ve b. "Poweraqua: Supporting users in querying and exploring the semantic web." *Semantic Web 3.3* , 249-265, **2012**.

[27] Bernstein, A., Kaufmann, E., & Kaiser, C. "Querying the semantic web with ginseng: A guided input natural language search engine." *15th Workshop on Information Technologies and Systems*, December, 112-126, **2005**.

[28] Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., & Weikum, G. "Natural language questions for the web of data." *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational*, 2012.

[29] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., ... & Welty, C. " Building Watson: An overview of the DeepQA project. ." *AI magazine*, 59-79, **2010**.

[30] Katz, B. "Annotating the World Wide Web using Natural Language ." *RIAO*, 136-159, **1997**.

[31] Tunstall-Pedoe, W. " True knowledge: Open-domain question answering using structured knowledge and inference." *AI Magazine*, 80-92, **2010**.

[32] Amasyalı, M. F., & Diri, B. "Bir soru cevaplama sistemi: Baybilmiş. ." *TÜRKİYE BİLİŞİM VAKFI BİLGİSAYAR BİLİMLERİ ve MÜHENDİSLİĞİ DERGİSİ*, 1(1), **2005**.

[33] Celebi, E., Gunel, B., ve Sen, B. "Automatic question answering for Turkish with pattern parsing." *Innovations in Intelligent Systems and Applications (INISTA)* (IEEE), 389-393, June **2011**.

[34] İlhan, S., Duru, N., Karagöz, Ş., & Sağır, M. "Metin Madenciliği ile Soru Cevaplama Sistemi." **2008**.

[35] Er, N. P., ve Cicekli, I. "A Factoid Question Answering System Using Answer Pattern Matching."*Proceedings of the 6th International Joint Conference on Natural Language Processing* (IJCNLP 2013), **2013**.

[36] Schlaefer, N., Gieselmann, P., Schaaf, T., & Waibel, A. "A pattern learning approach to question answering within the ephyra framework." *Text, speech and dialogue.* Springer Berlin Heidelberg, 687-694, **2006**.

[37] Derici, C., Çelik, K., Kutbay, E., Aydın, Y., Güngör, T., Özgür, A., & Kartal, G. . "Question Analysis for a Closed Domain Question Answering System." *Computational Linguistics and Intelligent Text Processing.* Springer International P, 468-482, **2015**.

[38] Biricik, G., Solmaz, S., Özdemir, E., & Amasyalı, M. F. A. "Turkish Automatic Question Answering System with Question Multiplexing: Ben Bilirim." 1, no. 6, 46-51, (June **2013**).

[39] http://tr.wikipedia.org/wiki/Zemberek_%28yaz%C4%B1l%C4%B1m%29. erişim tarihi: 09.26.2014

[40] *Workshop on Relational Models of Semantics.* (Association for Computational Linguistics), **2011**.

[41] Brill, Eric, Susan Dumais, and Michele Banko. "An analysis of the AskMSR question-answering system." *Proceedings of the ACL-02 conference on Empirical methods in natural language processing.* Association for Computational Linguistics, **2002**.

[42] Broekstra, Jeen, Arjohn Kampman, and Frank Van Harmelen. "Sesame: A generic architecture for storing and querying rdf and rdf schema." *The Semantic Web—ISWC 2002* (Springer Berlin Heidelberg), 54-68, **2002**.

7. Ekler

Ek 1:

Belirlenmiş Soru Tipi Şablonları:

YazarKitabı:

[x] + {kitap} + {liste} + {...}

[x]+ {sorukelimesi} + {kitap} + {yazmak} + {...}

[x] + {yazmak} + {kitap} + {...}

[x] + {yazmak} + {olmak} + {kitap} + {...}

[x] + {soru kelimesi} + {kitap} + {...}

FilmOyuncuları:

[x] + {soru kelimesi} + {aktör} + {oynamak} + {...}

[x] + {aktör} + {...}

ÜlkeÖzellikleri:

[x] + {ülke özelliği} + {...}

DoğumTarihi:

[x] + {soru kelimesi} + {zaman} + {doğmak} + {...}

[x] + {soru kelimesi} + {yaş} + {...}

TarihSorusu:

[x] + {soru kelimesi} + {zaman} + {...}

DoğumYeriSorusu:

[x] + {soru kelimesi} + {doğmak}

[x] + {arazi} + {soru kelimesi}

[x] + {doğmak} + {arazi}

[x] + {soru kelimesi} + {arazi} + { doğmak }

[x] + {yer bildiren soru kelimesi} + {...}

KimlikSorusu:

[x] + {kimlik bildiren soru kelimesi} + {...}

[x] + {hakkında} + {...}

KimYazdıSorusu:

[x] + {yazmak} + {eser} + {soru kelimesi} + {...}

[x] + {yazmak} + { soru kelimesi } + {eser} + {...}

8. ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Nicat SÜLEYMANOV
Doğum Yeri :Haçmaz, Azerbaycan
Medeni Hali :bekar
E-posta : nicat.suleymanov@gmail.com
Adresi : Babek cd 85A\54,Nizami ilçesi, Bakü, Azerbaycan

Eğitim

Lise : Bakü Türk Anadolu Lisesi
Lisans : Ankara Üniversitesi Bilgisayar Mühendisliği
Yüksek Lisans :Hacettepe Üniversitesi Bilgisayar Mühendisliği

Yabancı Dil ve Düzeyi Azerice(çok iyi), Türkçe(çok iyi), İngilizce(çok iyi), Rusca(orta)

İş Deneyimi Akgün Yazılım, Cybernet(Azerbaycan)

Deneyim Alanları Doğal Dil İşleme, Java Veb Teknolojileri

Tezden Üretilmiş Projeler ve Bütçesi

Tezden Üretilmiş Yayınlar 2015 2nd Intl. Conference on Soft Computing & Machine Intelligence

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar