

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**WEB ÖNERİ SİSTEMLERİNDE
ANLAMSAL WEB MADENCİLİĞİNİN
UYGULANMASI**

YÜKSEK LİSANS TEZİ

Müh. Göksel ORAL

Anabilim Dalı: Bilgisayar Mühendisliği

Programı: Bilgisayar Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. Şule Gündüz ÖĞÜDÜCÜ

NİSAN 2008

ÖNSÖZ

Gerek yüksek lisans ders aşamasında; gerek tez çalışması sırasında desteğini, önerilerini ve yol göstermeyi esirgemeyen danışmanım Yrd. Doç. Dr. Şule Gündüz ÖĞÜDÜCÜ'ye öncelikli olarak teşekkür ederim. Benim için yeni olan birçok konuda hem sorularıyla hem de ilettikleriyle bana ışık tutmuştur.

Hayatımın her aşamasında bana güvenen ve desteğini hiç esirgemeyen aileme ayrıca hem sevgisini, hem de anlayışını ve desteğini hiç esirgemeyen sevgili nişanlım Özge'ye minnet duyuyorum. Onların varlığıyla kendimi şanslı hissediyorum.

Tez çalışması boyunca gerek örnek veri kümelerinin hazırlanması gerek yapılan diğer öneri sistemleri hakkında yardımcı olan Veri Madenciliği ve Makine Öğrenmesi Proje Grubu üyeleri Gül Nildem Demir ve Murat Göksedef'e şükran borçluyum. Ayrıca Wu&Palmer Benzerlik çalışması sırasında kısa zamanda destek vermiş Ainura Madylova'ya da teşekkür ediyorum.

Yüksek lisans öğrenimim sırasında; çalıştığım firmalardaki yöneticilerime verdikleri destek ve izin için teşekkür ederim. Aynı sıraları ve iş ortamını paylaştığım bitmeyen arkadaşlığını ve veri madenciliği sevgisini gösteren Emre Akarsu'ya, bir ağabeyim ve örnek aldığım bir mühendis olarak gördüğüm Alp Şehiç'e ayrıca teşekkür ediyorum.

"Alet işler; el övünür" sözünden yola çıkarak, favori arama motorum Google'a yorulmayan çalışma ortaklığı ve temel konuların anlatımında aydınlatıcı bilgiler aldığım Wikipedia'ya teşekkür ediyorum.

Son olarak; bilim ışığını bize miras bırakan Ulu Önder Mustafa Kemal Atatürk'ü onun veciz sözüyle hatırlıyor ve ömür boyu minnet duyuyorum. "Ben size manevi miras olarak hiçbir ayet, hiçbir dogma, hiçbir donmuş ve kalıplaşmış kural bırakmıyorum. Benim manevi mirasım akıl ve bilimdir".

Nisan 2008
Göksel ORAL

İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	viii
ÖZET	ix
SUMMARY	xi
1. GİRİŞ	1
1.1. Anlamsal Web	2
1.2. Tezin Amacı	6
1.3. Tezin Yapısı	6
2. KONUYLA İLGİLİ ÇALIŞMALAR	7
2.1. Web Öneri Sistemleri	7
2.2. Anlamsal Web Madenciliği	8
3. SemRec ÖNERİ SİSTEMİ	10
3.1. Veri Hazırlama	11
3.1.1. Oturum Verisi Hazırlama	11
3.1.2. İçerik Verisi Hazırlama	14
3.1.3. Eşleştirme İşlemleri	19
3.2. Anlamsal Web Yapısı Oluşturmak	20
3.2.1. Ontoloji Oluşturmak	20
3.2.2. Sayfaları Ontolojiye Yansıtmak	22
3.3. SemRec Öneri Sistemi	23
4. DENEYSEL SONUÇLAR	31
4.1. Veri Hazırlama Sonuçları	31
4.1.1. Oturum Verisi Sonuçları	31

4.1.2.	İçerik Verisi Sonuçları	32
4.1.3.	Eşleştirme Sonuçları	34
4.2.	Anlamsal Web Yapısı Sonuçları	35
4.2.1.	Ontoloji Oluşturma Sonuçları	36
4.2.2.	Sayfaların Ontolojiye Yansıtılması Sonuçları	36
4.3.	Öneri Sistemi Sonuçları	36
4.3.1.	Vuru Sayısı, Kapsama ve Kısa yol Kazanç Sonuçları	36
4.3.2.	Karşılaştırma Matrisleri	43
4.3.3.	Diğer Web Öneri Modelleriyle Karşılaştırma	45
4.3.4.	Öneri Zamanlama Sonuçları	46
5.	SONUÇ ve İLERİ ÇALIŞMALAR	47
	KAYNAKLAR	50
A.	EKLER	55
A.1	ITU Veri Kümesine ait Anlamsal Web Yapısı	56
A.2	CS Veri Kümesine ait Anlamsal Web Yapısı	62
	ÖZGEÇMİŞ	67

KISALTMALAR

API	: Uygulama Programlama Arayüzü (Application Programming Interface)
CSV	: Virgül Ayraçlı Dosya
DAML+OIL	: DARPA Ajan İşaretleme Dili + Ontoloji Dönüştürme Dili (DARPA Agent Markup Language + Ontology Interchange Language)
DDİ	: Doğal Dil İşleme
HTML	: Zengin Metin İşaretleme Dili (HyperText Markup Language)
OWL	: Web Ontoloji Dili (Web Ontology Language)
RDF	: Kaynak Açıklama Ortamı (Resource Description Framework)
URL	: Tekdüzen Kaynak Bulucu (Uniform Resource Locator)
WWW	: Dünya Çapında Web (World Wide Web)
W3C	: Dünya Çapında Web Konsorsiyumu (World Wide Web Consortium)
XML	: Genişletilebilir İşaretleme Dili (Extensible Markup Language)

TABLO LİSTESİ

Tablo 2.1: İstatistik araçları tarafından oluşturulan bazı değişkenler	7
Tablo 3.1: Örnek Sunucu İz Dosyası İçeriği	11
Tablo 3.2: Veri Temizleme İşlemi Sonrasında İz Dosyası İçeriği	12
Tablo 3.3: İz dosyasındaki her isteğin ait olduğu kullanıcının belirlenmesi.....	13
Tablo 3.4: DilTesti Yordamı Bulanık Mantıksal Dönüş Değerleri.....	18
Tablo 4.1: Oturum Verisi Hazırlama Sonuçları	32
Tablo 4.2: Sayfalardaki bağlantı bilgileri.....	33
Tablo 4.3 ITU ve CS Veri kümelerindeki Oturum ve İçerik Veri Sayıları.....	35
Tablo 4.4: ITU ve CS Veri kümelerindeki Bağlantı Eşleştirme Sonuçları	35
Tablo 4.5: İTÜ Veri kümesinde öneri modellerinin vuru sayıları	37
Tablo 4.6: İTÜ Veri kümesinde, öneri modellerinin kısa yol kazancı ve kapsama oranları	38
Tablo 4.7: CS Veri kümesinde öneri modellerinin vuru sayıları	40
Tablo 4.8: CS veri kümesinde, öneri modellerinin kısa yol kazancı ve kapsama oranları	41
Tablo 4.9: ITU Veri kümesi üzerinde, öneri modelleri arasındaki karşılaştırma matrisi.....	43
Tablo 4.10: Anlamsal Önericinin sınıf bazında hata yapma oranı.....	43
Tablo 4.11: CS Veri kümesi üzerinde, öneri modelleri arasındaki karşılaştırma matrisi.....	44
Tablo 4.12: Anlamsal Öneri Modelinin sınıf bazında hata yapma oranı	44
Tablo 4.13: ITU Veri kümesi üzerinde, diğer Web öneri modelleriyle başarımların karşılaştırma	45
Tablo 4.14: CS Veri kümesi üzerinde, diğer Web öneri modelleriyle başarımların karşılaştırma	45

Tablo 4.15: Veri kümelerinde öneri zamanlama sonuçları	46
Tablo A.1: Ontoloji tablo açıklaması	55
Tablo A.2: ITU Veri Kümesi için Anlamsal Web Yapısı.....	56
Tablo A.3: CS Veri Kümesi için Anlamsal Web Yapısı.....	62

ŞEKİL LİSTESİ

Şekil 1.1: Anlamsal Web Katmanlı Yapısı	3
Şekil 3.1: HTMLEditorKit tarafından çağırılan metot örneği	15
Şekil 3.2: Zemberek DilTesi Yordamı Kullanım Şekli.....	17
Şekil 3.3: SemRec Programına Ait Akış Diyagramı.....	25
Şekil 4.1 ITU Veri Kümesindeki Dil Sınıflandırma Skor Dağılımı.....	34
Şekil 4.2 CS Veri Kümesindeki Dil Sınıflandırma Skor Dağılımı	34
Şekil 4.3: İTÜ Veri kümesinde, oturum ziyaret uzunluklarına göre başarımlar yüzdeleri	37
Şekil 4.4: İTÜ Veri kümesinde, Gerçekleşen önerilerin tahmin sırasına göre dağılımı	38
Şekil 4.5: İTÜ Veri kümesinde, Başarımlar X Kapsama oranı karşılaştırılması	39
Şekil 4.6: İTÜ Veri kümesinde, Başarımlar X Kısa yol Kazancı karşılaştırılması	39
Şekil 4.7: CS veri kümesinde, oturum ziyaret uzunluklarına göre başarımlar yüzdeleri.....	40
Şekil 4.8: CS veri kümesinde, Gerçekleşen önerilerin tahmin sırasına göre dağılımı	41
Şekil 4.9: CS veri kümesinde, Başarımlar X Kapsama oranı karşılaştırılması	42
Şekil 4.10: CS veri kümesinde, Başarımlar X Kısa yol Kazancı karşılaştırılması.....	42

WEB ÖNERİ SİSTEMLERİNDE ANLAMSAL VERİ MADENCİLİĞİNİN KULLANILMASI

ÖZET

Günümüzde Dünya Çağında Web (WWW) haber, reklam, eğitim, alışveriş, günlük, dosya paylaşım gibi bir çok servis ve hizmetin ucuz ve kolay sunulabildiği bir ortam halini almıştır. Servis sunmanın kolaylaştığı bu ortamda artan global rekabet, bireylere ya da gruplara özel farklılaşan, etkili çözümler üretilmesini gerektirmiştir. Bu açıdan müşteri memnuniyeti, eğilimleri ve kullanım profilleri baz alınarak, dinamik içerik ve bağlantılar içeren yapılar oluşturulmaktadır. Bu noktada Web öneri sistemleri hem içeriksel hem de istenilen içeriğe daha kolay ve çabuk erişmeyi sağlayan çözümler üretmeye çalışmaktadır.

WWW'ın hızla genişlemesi, uygulamalar açısından yararlı bilginin elde edilmesini zorlaştırmaktadır. Ama buna karşılık Web alanları bu çalışmaları teşvik edecek kullanım, etkileşim, sayfa içeriği ve Web yapısı gibi çok yönlü kullanışlı bilgiler sunabilmektedir. Bu noktada Web madenciliği, Web uygulamaları ve servislerince sunulan veriden otomatik olarak kullanılabilir bilginin elde edilmesiyle ilgilenen bir çalışma alanı olarak ortaya çıkmıştır.

Web öneri sistemleri, kullanıcıların Web sayfa gezimlerinde karar vermelerini kolaylaştıran yararlı bilgiler ya da ilgilenebilecekleri ürünleri, içeriği sunan öneriler üretmektedir. Önerilerin oluşturulmasında kullanım bilgileri sıkça kullanılmaktadır buna ek olarak içerik ve yapısal bilgilerin kullanılması istatistiksel modelleri daha açıklayıcı, nedensel ve başarılı kılmaktadır. Ancak hem çok fazla Web dokümanının olması hem de bu dokümanların yazılımlarca işlenebilir formatta olmaması bu çalışmalar için alt yapı eksikliğini işaret etmektedir. Bu noktada, WWW konusunda standartları belirleyen Dünya Çağında Web Konsorsiyumu (W3C), Web içeriklerinin makine okunabilir olmasını sağlayan bir teknolojiyi sunmuştur.

Anlamsal Web, bilgi düzeyini katmanlı bir yapıda zenginleşen makine-okunabilir veri yapıları sunmaktadır. Bu veri yapılarının oluşturulmasında DDİ (Doğal Dil İşleme) yöntemlerinden faydalanılmakta ve Web dokümanları üzerinde anlamsal açıklamalar oluşturulmaya çalışılmaktadır. Oluşturulan bu veri yapıları sayesinde yapısal ya da yarı yapısal içerikler yazılımlarca işlenerek Web madenciliği çalışmalarında kullanılabilir hale gelmektedir.

Bu tez çalışmasında, öncelikle; Web sayfalarına ait içerikler temel anlamsal Web yapısı olan ontolojiye aktarılması sağlanmaktadır. Burada Web alanına ait ontolojinin oluşturulması ve ardından her sayfanın bu ontolojiye yansıtılması işlemleri yapılmaktadır. Sonrasında; oluşturulan ontolojiyle kullanıcı oturum bilgileri bir araya getirilerek iki aşamalı bir öneri sistemi oluşturulmuştur. İlk aşama sınıf öneri modeli olarak adlandırılır. Bu aşamada sayfa bazında oluşturulan oturum bilgileri ontolojideki her sayfaya karşılık gelen sınıf bilgisine dönüştürülerek sınıfsal

bir gösterilim oluşturulur. Bu bilgiyle sınıf önerisi üreten model, herhangi bir sınıftan bir sayfayı ziyaret eden kullanıcının, bir sonraki ziyaret edeceği sayfanın hangi sınıflardan olacağını önermektedir. İkinci aşama, anlamsal öneri modeli olarak adlandırılır ve kullanıcının bir sonraki ziyaret edebileceği sayfalar bu model tarafından belirlenmektedir. Bu model, sınıf öneri modeli tarafından önerilen sınıflara ait sayfalar üzerinden bir sonraki yüksek olasılıklı ziyaret edilebilecek sayfalar oluşturulmaktadır. Deneylerimizde her öneri sisteminde üretilen ilk dört tahmin esas alınmaktadır ama tahmin sırasına göre başarımların dağılımına da dikkat edilmektedir.

Önerilen yöntemlerimize ait sonuçlar, gerçekleştirilen diğer bir istatistiksel öneri sistemiyle karşılaştırılmaktadır. Sonuçların genel geçerliliği, boyut ve içerik olarak farklı iki veri kümesi üzerinde sınanmıştır.

Önerdiğimiz yöntemler; doğruluk, vuru sayısı, kısa yol kazancı, örtme yüzdesi gibi metriklerle karşılaştırılmaktadır. İki farklı veri kümesi üzerinden sonuçlara baktığımızda; sınıf bazlı öneri sistemi, bir sonraki sayfaya ait sınıfın belirlenmesinde etkili sonuçlar üretmektedir. Elde edilen sonuç, hem Anlamsal Web'e iletilen tahminlerdeki hem de sınıf bazında kullanıcıya yönlendirilebilecek önerilerdeki başarımları göstermektedir. Anlamsal öneri sisteminde ise karşılaştırdığımız öneri sistemine nazaran daha iyi bir netice elde edilememiştir. Buna karşın, bu öneri sistemi daha iyi bir önbellekte tutma işlevi sunabilecek kısa yol kazancı sunmaktadır. Ayrıca doğru önerilerin, oluşturduğumuz modellerin ikisinde de daha öncelikli tahminler arasından geldiğini gözlemleyebildik. Son olarak; anlamsal öneri modelinin, sınıf öneri modeli tarafından önerilen sınıflara ait sayfalar içerisinden tahmin üretmesi nedeniyle; önerilen modeller kesin bir süre ve bellek kazancı sunmaktadır.

SEMANTIC WEB MINING ON WEB RECOMMENDER SYSTEMS

SUMMARY

Nowadays *World Wide Web* (WWW) became an easy servable, cheap environment for news, advertisements, education, e-commerce, diary, file sharing etc. With the increasing global compition caused by easy serving utilities of internet, it necessitates a distinctive, effective solutions for individuals and groups. For this point of view, Web pages are built by dynamic contents and link based on customer care, propensity and user profile data. At this point, Web recommender systems provide solutions for dynamic content and link infrastructure.

With the extending structure of WWW, it becomes though for applications to extract useful information. But on the other hand, Web domains include usage, interaction, content and structural information motivating the extraction technologies be improved. At this point, Web mining is defined as the use of data mining techniques to automatically discover and extract information from Web documents and services. In this research field, there are three sub study fields: Web Content Mining, Web Usage Mining and Web Structure Mining.

Web Recommender Systems purposes recommendation at decision-making through click-streams with providing interesting and useful products and contents. Recommendation is frequently based on usage data but additionally, usage of content and structure data make the probabilistic recommenders more descriptive, reasonable and accurate. But lack of processing power of software agents on huge amount of not-formatted Web documents, points us the absence of infrastructure for content mining works. At this point, *World Wide Web Consortiun* (W3C) , specifying WWW standarts, purposes a new technology named Semantic Web serving an machine-readable environment for Web contents.

Semantic Web purposes machine-readable data structures where information gain is enriching by increasing level. While building up these data structures, NLP techniques are being used and semantic annotations are applied on each page. With these data-structures, un-formatted or semi-formatted contents can be usable by software agents.

In this thesis work, firstly Web contents are mapped to Web domain ontology. At that point, Web domain ontology creation and ontology mapping processes are done. Laterly, we purpose two step recommendation system using ontology structure weighted with session data. First step is called as class recommender model. In the first step, page based sessions are transformed into class based sessions by matching each page with mapped ontology concept. For the user visiting a page member of an ontology concept, class recommender model generates most probable next Web page's ontology concept. Second phase is called as semantic recommender model and for the user it generates most probable next web page using the outputs of class

recommender model. In our experimental results; for accuracy, first four recommendation is taken as basis and also the hit ratios due to recommendation order is measured.

Our purpose models are compared with a probabilistic recommender system. We conducted our experiments on two different contented and sized Web site datasets.

We evaluate our recommender model using four different metrics namely accuracy, hit-ratio, shortcut gain, coverage ratio. Due to results from two different datasets, class recommender produce effective results. The accuracy of this model shows the success of standalone class recommendation and also input class parameters passed to semantic recommender. But on the other hand, our semantic recommender can not produce better recommendations in comparison with other probabilistic recommender. However, this recommender produce better shortcut gains pointing to a better caching strategy. Additionally; in both purposed models, correct recommendations came from more priorite recommendations in comparision with other probabilistic recommender. Lastly, since page recommender Works only on a subset of Web pages due to the class recommender recommendation, our recommender increases the recommender speed and decreases memory usage while maintaining recommendation accuracy reasonable.

1. GİRİŞ

Dünya Çapında Ağ (WWW) insanlara sınırların olmadığı bir dünya sunarken; logaritmik olarak artan [1,2] eğitim, kamusal ve haber gibi çok çeşitli alanları kapsayan bilgi servislerini bizlere sunmaktadır. Eski tip servis yapılarının yerine; daha başarılı, etkili, zamandan ve maliyetten kazanç sağlayan servisler kazandırmıştır.

WWW reformuyla beraber yayıncı ve kullanıcı sayılarındaki inanılmaz artış, verilen hizmetler açısından yeni gereksinimlerin karşılanmasını gerektirmiştir. Örnek olarak arama motorları açısından, aranılan sözcüklerle hızla artan Web sayfaları arasından hedef sayfaların hızlı ve doğru olarak bulunması, sıralanması gittikçe zorlaşmaktadır. Bu açıdan Web sayfalarının daha fazla makine okunabilir olması, Web sayfalarındaki bilgilerinin birbirleriyle örtüştürülmesi gerekmektedir.

Bu noktada, Anlamsal Web teknolojisi Berners Lee ve diğ. (2001) tarafından bilgi servislerinin genişletilebilmesini sağlamak ve Web sayfa içeriklerinin daha fazla makine okunabilir olması açısından öngörülmüştür [11]. Bu öngöründe, Kaynak Tanımlama Çerçevesi (RDF), Ontoloji gibi yeni yapısal veri yapıları ortak Genişletilebilir İşaretleme Dili (XML) formatında tanımlanmıştır. Bu veri yapıları, Web alanına ait ortak kategorisel tanımlamaları ve anlamsal ilişkileri içermektedir. Bu şekildeki ek tanımlamalar “Bulunduğum ilçede saat 21’den sonra açık, kredi kartıyla ödeme yapılabilen en uygun DVD oynatıcı fiyatına sahip mağaza bilgisini getir” şeklindeki karmaşık sorgulara da cevap verebilen bir altyapı ortamını sunmaktadır. Anlamsal Web teknolojisinin karmaşık, büyük Web alanlarında da kullanılabilmesi için Doğal Dil İşleme (DDİ) görevlerinin ve bazı yapay zeka(AI) tekniklerinin uygulandığını görebilmekteyiz [11].

İçerikle ilgili çalışmalarla beraber Web alanının yapısı ve sayfa erişimine ilişkin Web madenciliği çalışmaları da bulunmaktadır. Kısaca Web alanının yapısıyla ilgili çalışmalarda, birbirine iki yönlü bağlı sayfalar arasında frekans bağlantı örüntülerinin oluşturulmaya çalışılmaktadır. Sayfa kullanımına ilişkin çalışmalar da ise, kullanıcı tercihleri üzerinden öngörme ve öneri yapılmasına çalışılmaktadır.

Tez çalışmasında önerilen modellerde, hem Web madenciliğinin öngörme ve yapısal analizleri hem de Anlamsal Web yapısının bize sunduğu kategori ve kategoriler

arasındaki anlamsal ilişkiler kullanılmaktadır. Bu şekilde çok yönlü bir Web madenciliği yapısıyla, oluşturulan modellerin daha başarılı sonuç vereceği düşünülmekte ve uygulanacakları farklı ortamlardaki sapmaların minimum düzeyde olacağı tahmin edilmektedir.

1.1. Anlamsal Web

2001 yılında, Berners-Lee, Hendler ve Lassila gelecek Web için Anlamsal Web adını verdikleri bir vizyon ortaya attılar. Bu vizyona göre; sunulan bilgilerin global bir veritabanı bilgisi gibi kullanılabilmesi planlanmaktadır. Bu yapının oluşturulması için çok farklı ortamlarda dağılmış bilgilerin ortak yapılara yansıtılması ve bu yapılarında yeni bilgilerle genişletilmesi önerilmektedir [5]. Sunulan teknoloji, sadece Web sayfalarındaki bilgilerin değil ayrıca veritabanlarındaki, Web servislerindeki bilgilerin de bu yapıya geçirilebilir bir altyapı oluşturmaktadır [8]. Bunlarla beraber, Berners-Lee Anlamsal Web teknolojisinin, insandan insana olan bilgi paylaşım ortamını daha makine okunabilir hale getirmek hedefinde olduğunu belirtmektedir [4]. Ama hali hazırda, Web ortamında çoğunluğu yapısal ya da yarı-yapısal içerik bulunmaktadır ve bu durum da makine okunabilir formata geçimi zorlaştırmaktadır. Bu sorunu aşmak üzere, Bernard-Lee Anlamsal Web teknolojisi içerisinde katman katman anlamsal olarak zenginleşen makine okunabilir veri yapıları sunmaktadır. Şekil 1.1’de W3C tarafından yayımlanan standartta belirttiği Anlamsal Web’i oluşturan meta-dillerin kapsamlı ve fonksiyonel katmanlardan oluşan mimarisi sunulmaktadır [5].

Tanımlanan meta-dillerin oluşturduğu katmanlar şunlardır:

(1) *Evrensel Kod (Unicode)/ Düzgün Kaynak Belirteci (URI)*

Yapısal olmayan Web verisi

(2) *XML/İsim Uzayı (Namespaces) / XML Şeması (Schema)*

Yarı-yapısal Web verisi

(3) *Kaynak Tanımlama Çerçevesi (RDF)/ RDF Şeması (Schema)*

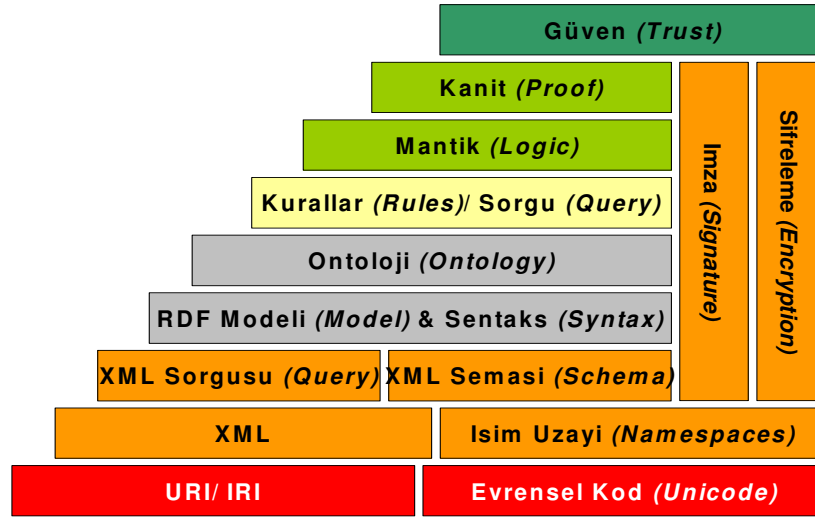
Kaynak Tanımlama Çerçevesi (RDF) (Resource Description Framework), meta-dil olarak tasarlanmış World Wide Web Consortium (W3C) tarafından önerilen bir tanımlamadır. RDF meta-dil modeli, kaynaklar hakkında, RDF terminolojisinde üçlüler (triples) olarak adlandırılan, özne-yüklem-nesne şeklinde ifadeler üretme fikri

üzerine kurulmuştur. Özne kaynağı, yüklem, kaynağın özelliğini veya durumunu gösterir ve özne ve nesne arasındaki ilişkiyi açıklar [6]. RDF yapısı aşağıdaki işlemlerle oluşturulmaktadır [9,10]:

a. Kavram (concept) oluşturma

Kavram hedef alandaki sınıflara karşılık gelmektedir ve kavramların oluşturulmasında aşağıdaki kısıtlar bulunmaktadır.

- i. Kavramlar birden fazla sınıfın alt ya da üst sınıfı olabilir. Ama tek bir sınıfın üst sınıfı olarak tanımlanması yararlı bir tanımlama değildir.
- ii. Alt sınıflar üst sınıfların özellik tanımlarını ezebilmektedir.
- iii. Dairesellik içeren (ortak üst ve alt sınıflara sahip) bir hiyerarşik tanımlamaya izin verilmemektedir, bunun yerine bu sınıfların denk olduğu belirtilmelidir.
- iv. Kavram adlarının hepsi ya çoğul ya da tekil isim içermelidir.



Şekil 1.1: Anlamsal Web Katmanlı Yapısı

b. Özellik (Slot) oluşturma

Kavrama ait nitelikler ve hiyerarşik olmayan ilişkiler, kavram altında özellikler tanımlanarak belirtilmektedir. Özelliklere ilişkin aşağıdaki kısıtlar ve durumlar bulunmaktadır:

- i. Alt sınıflar üst sınıflarda belirtilen bütün özellikleri içermektedir.
- ii. Her özelliğe ait etki alanı ve aralık tanımlamalarının yapılması gerekir. Bu tanımlar eğer tüm alt sınıflarda ortaksa üst sınıflarda yapılması önerilmektedir.
- iii. Bazı ilişkiyi ifade eden özellikler iki-yönlü tanımlanır.

Örn: tüketmek - üretmek

RDF şeması, ortak RDF geliřtirmelerini saęlamak için; önceden tanımlı RDF kaynaklarını ve sınıf ve ilişki yapısını içeren şematik tanımlamadır [12].

(4) Ontoloji (Ontology)

Ontoloji belirli bir konuya ait kavramları, bu kavramlar arasındaki ilişkileri ve kavramlara ait örnekleri içeren bir veri modelidir [7]. Ontolojinin oluşturulmasında RDF tanımlamalarına ek olarak aşağıdaki ek tanımlamalar ve düzeltmeler yapılmaktadır:

a. Yön (Facet) oluşturma

Yön (facet) tanımları özellięe ait kısıtları içermektedir. Yöne ait aşağıdaki tanımlar yapılmaktadır:

- i) Yön tipi: Katar (String), Boolean, Sayı (Number), Sıra (Enumerated), Sınıf (Class) tipleri
- ii) Yön değeri: Tekil yön değeri için tek değeri, çoklu yön değeri için birden fazla değeri girilebilir.
- iii) Kardinalite (Cardinality): Kardinalite (Cardinality) bir değişenin alabileceęi minimum ve maksimum değeri sayısını belirtmektedir.
- iv) Varsayılan (Default) değeri: Özellięe ait varsayılan değeri belirtmektedir.

b. Örnek (Instance) oluşturma

Örnekler ait oldukları kavrama ait sınıfın altında yer alırlar. Örneklerin oluşturulmasında sınıflara ve üst sınıflara ait özelliklere ait bilgiler doldurulmaktadır.

c. Aykırı (disjoint) sınıf tanımlama

Aykırı sınıflar aynı seviyede tanımlanan ama ortak alt sınıfa sahip olmayan sınıflardır. Bu tanımla bir örneğin aykırı sınıflardan sadece birine ait olabileceęi garanti edilmiş olmaktadır.

d. Sınıflara ait bileşim (union) ve kesişimlerin (intersections) tanımlanması

Tanımlanan sınıf bilgileri varolan sınıflardan bazılarının bileşimi ya da kesişimi olabilir.

Örnek:

```
<owl:Class rdf:ID="Meyve">  
  <owl:unionOf rdf:parseType="Collection">  
    <owl:Class rdf:about="#TatlıMeyve" />  
    <owl:Class rdf:about="#TatsızMeyve" />  
  </owl:unionOf>  
</owl:Class>
```

```
</owl:unionOf>  
</owl:Class>
```

Meyve nesnesi hem TatlıMeyve sınıfının hem de TatsızMeyve sınıfının özelliklerini içerdiği yukarıdaki bileşim ifadesiyle (owl:unionOf) belirtilmektedir.

(5) Mantık (Logic)

Bu katmanda ontoloji ilişkilerine göre kuralların oluşturulduğu bölümdür.

(6) Kanıt (Proof)

Mantık katmanında oluşturulan kurallar bu katmanda çalıştırılmaktadır.

(7) Güven (Trust)

Kanıt katmanında oluşturulan sonuçlara güvenilip güvenilemeyeceğinin değerlendirildiği katmandır.

Yukarıda bahsedilen Anlamsal Web yapısı katmanları, her aşamada yeni tanımlamalar eklenerek geliştirilmektedir. Web alanlarında bulunan yapısal olmayan ya da yarı-yapısal sayfalar öncelikle RDF'lere çevrilmektedir. RDF yapısının oluşturulmasında gerekli özne-yüklem-nesne üçlülere (triples) temel NLP işlemleriyle ve anlamsal açıklamaların oluşturulmasıyla yapılmaktadır. Bir sonraki katman olan ontolojinin oluşturulmasında; RDF'teki üçlülerden özne ontolojideki sınıfa; yüklem ontolojideki hiyerarşik olmayan ilişkiye, nesne de ontolojideki sınıf özelliğine yansıtılmaktadır. Bu aşamada elde edilen sınıflar arasındaki hiyerarşik ve denklik ilişkisinin oluşturulması açısından ontoloji öğrenme işlemi gerçekleştirilmeli ve ardından son ontolojiye örneklerin yansıtılması işlemi yapılmalıdır.

Ontoloji öğrenme işlemi, sınıflar arasındaki hiyerarşik ilişkiler Karoui, L. ve diğ. (2006)'nin uyguladığı hiyerarşik demetleme yöntemiyle elde edilebilmektedir [34]. Denklik (thesaurus) ilişkisi ise Hearst, M. (1992)'in belirttiği şekilde birlikte bulunma sıklığı gösteren kelimeler ve bu kelimelerin işaret ettiği kavramlar arasında tanımlanabilmektedir [35].

Örneklerin ontolojiye yansıtılması işleminde ise sayfalardan elde edilen örneklerin en yakın olduğu kavrama atanması yapılmaktadır. Buradaki yakınlık kavramı, Prasenjit M. ve diğ. (2005)'nin OMEN [37] adı verdikleri çalışmalarındaki gibi olasılıksal bir

yaklaşım ile ifade edildiği gibi A.Doa ve diğ. (2001)'nin GLUE [36] adını verdikleri çalışmalarında sınıflandırma sonucu olarak değerlendirilmektedir.

1.2. Tezin Amacı

Bu tezin amacı, sadece oturum bilgisinin kullanıldığı öneri sistemlerine göre; hem oturum bilgisinin hem de oluşturulan Anlamsal Web yapısıyla içerik bilgisinin kullanıldığı daha iyi sonuç veya kısa yol kazancı ayrıca hız ve az bellek kullanımı sağlayan bir öneri sisteminin oluşturulmasıdır. Oluşturulacak öneri sisteminden, hem sayfa kategorisi hem de sayfa bazından tahminlerin oluşturulması ve bunun yanında önerilerin Anlamsal Web'de karşılık gelen ilişkiyle kullanıcıya ifade edilerek açıklayıcı bir önerme yapılması amaçlanmaktadır.

1.3. Tezin Yapısı

Tez çalışması beş bölümden oluşmaktadır. Bu çalışmada, anlamsal Web yapısının Web öneri sistemlerinde nasıl uygulanabileceğini oluşturduğumuz iki aşamalı öneri sistemiyle tanıtmaktayız. Önerdiğimiz öneri sistemiyle karşılaştırdığımız öneri sistemlerine ilişkin çalışmalar 2. Bölümde sunulmaktadır. Öneri sistemimize ait temel bileşenler ve bunların oluşturulmasında uygulanan yöntem ve kabuller 3. Bölümde tanıtılmaktadır. 4. Bölümde birbirinden içerik, yapı ve boyut olarak farklı iki test veri kümesi üzerinden elde edilen deneysel sonuçlar sunulmaktadır. Son olarak, 5. Bölümde sonuçlar ve ileriki çalışmalara ait yorumlar belirtilmektedir.

2. KONUSYLA İLGİLİ ÇALIŞMALAR

2.1. Web Öneri Sistemleri

Gerçek hayatta öneri sistemleri, konuşmalar, yayımlanan eleştiriler, haberler ve araştırmalar aracılığıyla yapılmaktadır. Tıpkı gerçek hayattaki gibi Web öneri sistemlerinde yeni bağlantıların ve alternatiflerin kullanıcıya sunulması istatistiksel analizler, oylamalarla yapılmaktadır. Bunlardan en basiti, Tablo 2.1'deki gibi istatistiksel değişkenlerin tanımlandığı WebTrends [46] ve SurfAid [47] gibi araçlardır. Buna ek olarak PageGather en çok ziyaret edilen sayfalarla kullanıcılar arasında bir ilişki kurulmaktadır. Buradaki ilişki, sayfalar arasındaki geçimleri gösteren çizgede kullanıcının tercihleri işaretlendikten sonra benzer tercihlere sahip diğer kullanıcıların profilleriyle gösterilmektedir [21].

Tablo 2.1: İstatistik araçları tarafından oluşturulan bazı değişkenler

İstatistikler	Detaylı Bilgi
Web sayfası aktivite istatistikleri	Toplam ziyaret sayısı
	Ortalama vuru sayısı
	Başarılı/başarısız/yönlendirilen/önbelleklenmiş vurular
	Ortalama gözlem zamanı
	Ortalama gezinim uzunluğu
Hata istatistikleri	Sunucu hataları
	Sayfa Bulunamadı hatası
Sunucu istatistikleri	En çok ziyaret edilen sayfalar
	En fazla giriş/çıkış sayfaları
	En fazla tek erişilen sayfalar
İşaretçi istatistikleri	En çok işaret edilen sayfa
	En çok aranan sayfa
	En çok kullanılan arama anahtar kelimeleri
Demografik istatistikler	En iyi coğrafik konum
	En aktif ülke/il/organizasyon
İstemci istatistikleri	Ziyaretçinin Web tarayıcısı, işletim sistemi ve çerezler

İlişkilendirme kuralları öneri sistemlerinde de kullanılabilir. Lin ve diğ. (2002)'nin çalışmasında aynı işlem sırasında birliktelik ilişkisindeki sayfalara dayanılarak öneri üretilebileceği belirtilmektedir [48].

Demetleme veri analizlerinde, veri kümesinden kendi içinde homojen, birbirleri arasında farklı alt kümeler oluşturmaktır. Bu teknik, Web öneri sistemlerinden oturum, kullanıcı ve sayfa verisi üzerinde uygulanmaktadır. Yan ve diğ, (2001)'nin çalışmasında kullanıcıları, oturumlarında ziyaret ettikleri sayfalara bakarak demetlemektedir. Burada her kullanıcı oturumu bir vektör ile temsil edilmektedir. Bu vektörün boyutu site içindeki sayfa sayısına eşittir. Vektörde sayfaya ait değer, kullanıcının o sayfayı ziyaret sayısı, ziyaret etmediyse 0 olarak belirlenmektedir. Bu şekilde oluşturulan vektörler demetlenerek kullanıcıların demetlenmesi sağlanmaktadır [22].

Web kullanım madenciliğinde kullanılan sekans verisinden örüntü elde etme tekniği, öneri sistemleri içinde kullanılabilir. Burada birbirinden ayrılan iki yaklaşım bulunmaktadır: Apriori-tabanlı ve (Web Erişim Örüntüsü) WAP-Ağacı tabanlı. Apriori algoritması orijinal olarak Agrawal (1994) tarafından işlemsel veritabanlarında sıklıkla geçen elemanların ilişkisel kurallar oluşturularak bulunmasıyla ortaya çıkmıştır [45]. Web öneri modelleri açısından; Agrawal ve Srikant (1994) çalışmasında oturum bilgisini içerisindeki en sıklıkla bulunan öğelerle ifade ettikten sonra AprioriAll adını verdiği sekans örüntü bulma yöntemini kullanmıştır [23]. WAP-Ağacı ise sekans bilgilerinin üst üste bindirilerek oluşturulan ağaç yapısıdır. Pei ve diğ. (2000)'leri tarafından WAP-mine algoritması Web erişim örüntülerini WAP-Ağacı oluşturularak bulmaktadır [24].

Yukarıdaki yöntemlerden farklı olarak Goldberg ve diğ. (2001) “collaborative filtering” adını verdikleri farklı kişilerin düşüncelerini harmanlayan bir yöntem sunmuştur. Yöntemde kullanıcılar oylamaya sunulan içeriklerle ilgili verdikleri cevaplara göre demetlemekte ve bir kullanıcı için oluşturulan öneri ona benzer diğer kullanıcıların önerdiği içerik olmaktadır [20].

2.2. Anlamsal Web Madenciliği

Anlamsal Web Madenciliği genel itibariyle içerik madenciliği açısından bir yöntem olarak kullanıldığı gibi Web dokümanları arasındaki ilişkileri ifade edilmesi açısından da kullanılmaktadır [38-40].

Eirinaki,M. ve diğ. (2003) yaptıkları çalışmada, Web içerik bilgisinin kullanım bilgisiyle beraber işlenmesi için C-Log adını verdikleri bir veri yapısı

oluřturulmuřtur. Bu veri yapısında oturum bilgisine ek olarak; ziyaret edilen sayfayı anlatan anahtar kelimeler listesi oluřturulmaktadır. Eklenecek kelime listesi; sayfa ierisindeki kelimeler varolan bir taksonomide Wu&Palmer [49] benzerlięi kullanarak ait olduęu dęümü tanımlayan kelimelerle bunlarla eř anlama gelen kelimeler kullanarak oluřturulmaktadır [38].

Oberle,D. ve dię. (2003) ise C-Log'a benzer ama anahtar kelime yerine ontoloji kavramlarını kullanıldıęı bir veri yapısı onererek, kullanım bilgisiyle ierik bilgisinin birleřtirmeye alıřmıřtır [39].

Anlamsal Web'in farklı bir kullanımı da, Ziegler, C. (2004) tarafından yapılan alıřmada, sayfaya ait ilgi skorunun oluřturulan ontolojideki iliřkili olduęu dięer sayfalara daęıtılmasıyla sayfalar arasında ierięe baęlı bir baę kurulmasını saęlamıřtır [40].

3. SemRec ÖNERİ SİSTEMİ

SemRec Öneri Sistemi önerilerini istatistiksel olarak ağırlıklandırılmış Anlamsal Web ilişkilerine dayandırmaktadır. Bu açıdan öneri sistemi temel olarak üç ana bileşenden oluşmaktadır: Veri altyapısı, Ontoloji yapısı ve Öneri yapısı.

İlk bileşen olan veri altyapısı, Web alanına ait kullanıcı oturum ve içerik verisini içermektedir. Kullanıcı oturum bilgisi, Web sunucularına ait erişim kayıtlarından elde edilmektedir. Her kullanıcı oturumu, bir kullanıcının web sitesini tek bir ziyareti sırasında istek yaptığı sayfalardan oluşmaktadır. İçerik verisi ise Web sayfalarının etiketlerden (tag) oluşan yapısı içerisinde belirtilmiş metin, bağlantı ve başlık gibi bilgilerden oluşmaktadır. İçerik verisi, Web alanına ait ontolojinin oluşturulmasında ve her sayfanın bu ontolojiye yansıtılması sırasında kullanılmaktadır.

İkinci bileşen olan ontoloji yapısı; web alanında belirtilen kavramları, bu kavramlara ait sayfaları ve bu kavramlar arasındaki ilişkileri içermektedir. Oluşturulan ontolojide, Web alanına ait hiyerarşik ve hiyerarşik olmayan ilişkiler bulunmaktadır. Sonuç ontoloji kullanılarak, sayfaların ait olduğu kavram elde edilmiş olup sayfalar arasındaki geçimler anlamsal ilişki bazında açıklanabilmektedir.

Son bileşen olan öneri yapısında; ilk iki bileşende oluşturulan istatistiksel ve anlamsal Web yapıları kullanılmaktadır. Bu yapı içerisinde iki aşamalı bir öneri modeli uygulanmaktadır. İlk aşamada, bir sonraki sayfanın ait olabileceği ontoloji sınıflarını öneren sınıf öneri sistemi çalıştırılmaktadır. İkinci aşamada, sınıf önericisinin önerdiği sınıflara ait sayfalar üzerinden sayfa önerici uygulanmaktadır. İki aşamada da öneriler, istatistiksel sayfa ve sınıf geçim bilgileri üzerinden hesaplanan olasılıklarla hesaplanmaktadır. Bu bölümde, yukarıda bahsedilen bileşenlerin oluşturulmasına ait uygulanan yöntem ve kabuller tanıtılmaktadır.

3.1. Veri Hazırlama

Bu tez çalışmasında, Web alanına ait sunucu erişim izleri hem de sayfa içerikleri kullanılacaktır. Bu kaynak verileri işlenmemiş halde Web sunucusundan alınıp, sonraki aşamalarda kullanılabilecek biçime getirilmesi aşağıdaki Bölüm 3.1.1 ve Bölüm 3.1.2’de anlatılmaktadır.

3.1.1. Oturum Verisi Hazırlama

Kullanıcıya ait kullanılabilir Web bilgileri; gezim bilgisi, sorgular, çerez (cookie) bilgisi, kayıt bilgisi ve uygulamada ek olarak tutulan etkileşim bilgileri olabilir. Bu bilgiler sunucu izlerinde, vekil (Proxy) sunucu izlerinde ya da istemci tarafında bulunuyor olabilir.

Sunucu ve vekil sunucu izlerinde, Web sayfasına yapılan isteğe ait IP adresi, istek zamanı, temel http protokol bilgisi bulunmaktadır. Bu dosya da yer alabilecek bilgiler genelde Web sunucuları tarafında konfigürasyonu yapılabilmektedir. Örnek bir sunucu iz dosyası içeriği Tablo 3.1’de sunulmaktadır.

Tablo 3.1: Örnek Sunucu İz Dosyası İçeriği

İstek Kaynağı	İsteğin Günü ve Zamanı	Metot URL, (http Protokolü)	Durum Kodu	Transfer Edilen Bayt
216.35.116.28	[01/01/2007:00:00:01+0200]	“GET/http/1.0”	200	656
216.35.116.28	[01/01/2007:00:00:10+0200]	“GET a.gif http/1.0”	200	5462
216.35.116.28	[01/01/2007:00:00:30+0200]	“GET a.html http/1.0”	200	1231
216.35.116.28	[01/01/2007:00:01:30+0200]	“GET b.html http/1.0”	200	2342
216.35.116.29	[01/01/2007:00:01:25+0200]	“GET bb.html http/1.0”	404	1232
216.35.116.29	[01/01/2007:00:01:30+0200]	“GET c.html http/1.0”	200	5652

İstemci tarafında tutulan bilgiler çerez bilgisi ve etkileşim verisi oluşmaktadır. Bu verilerin kullanılabilmesi açısından tarayıcı (browser)’da ek yazılımlarla bu verinin ortak bir sunucuya aktarılması gerekmektedir. Bu sayede kullanıcının ilgilendiği temel alanlar, fare ve klavye hareketleri izlenebilmektedir.

Kullanıcıya ait bu işlenmemiş verilerin ileriki Web madenciliği işlemlerinde kullanılması açısından temizlenmesi ve ön işlenmesi gerekmektedir.

Veri temizleme

Veri temizleme aşamasında iz dosyasında Web dokümanı ile ilişkili olmayan dosya, çoklu ortam (multimedia) içerik istekleri ve hatalı sayfa cevaplarına ait satırlar elenmektedir. Tablo 3.1’de belirtilen Durum Kodu 200 olmayan ya da Metot URL alanında Web sayfa istemi olmayanlar silinmektedir. Veri temizleme sonucunda aşağıdaki örnek iz satırları kullanılacaktır.

Tablo 3.2: Veri Temizleme İşlemi Sonrasında İz Dosyası İçeriği

İstek Kaynağı	İsteğin Günü ve Zamanı	Metot URL, (http Protokolü)	Durum Kodu	Transfer Edilen Bayt
216.35.116.28	[01/01/2007:00:00:30+0200]	“GET a.html http/1.0”	200	1231
216.35.116.28	[01/01/2007:00:01:30+0200]	“GET b.html http/1.0”	200	2342
216.35.116.29	[01/01/2007:00:01:30+0200]	“GET c.html http/1.0”	200	5652

Kullanıcı Tanımlama

Web sayfası üzerindeki kullanıcı aktivitelerinin izlenebilmesi açısından, iz dosyalarında tekil kullanıcıya ait bir tanımlama getirilmesi gerekmektedir. Eğer Web sunucusuna belirli bir kullanıcı adıyla erişim sağlanıyorsa, doğrudan bu tanımlama kullanılabilir. Eğer bu şekilde bir erişim yoksa tüm istekler isimsiz (anonymous) kullanıcıya ait istekler olarak görülmektedir. Bu durumda eğer vekil sunucu üzerinden bu sunucuya erişim sağlanıyorsa IP adresi tekil kullanıcı tanımı olarak kullanılabilir. Eğer vekil sunucusu üzerinden bağlantılar varsa, bu bağlantılar için istemci tipi ve bazı diğer ek değişkenlerle (işletim sistemi vb.) bir tanımlama yapılabilir. Aşağıdaki tabloda Veri Temizleme işleminin ardından elde ettiğimiz Tablo 3.2’ye eklenen Kullanıcı ID kolonuyla, kullanıcı tanımlamaları her istek için IP adresi üzerinden aşağıdaki Tablo 3.3’deki gibi yapılmaktadır.

Tablo 3.3: İz dosyasındaki her isteğin ait olduğu kullanıcının belirlenmesi

Kullanıcı ID	İsteğin Günü ve Zamanı	Metot URL (http Protokolü)	Durum Kodu	Transfer Edilen Bayt
1	[01/01/2007:00:00:30 +0200]	“GET a.html http/1.0”	200	1231
1	[01/01/2007:00:01:30 +0200]	“GET b.html http/1.0”	200	2342
2	[01/01/2007:00:01:30 +0200]	“GET c.html http/1.0”	200	5652

Veri temizleme aşamasında elde edilen aşağıdaki sunucu erişim verisi aşağıdaki şekilde ifade edilirse:

$$L = L_1, L_2, \dots, L_{|L|}$$

$$L_i = (KaynakIP_i, ZAMAN_i, URL_i, CODE_i, BYTE_i)$$

$$L_i \in L, i \in [1 \dots |L|]$$

kullanıcı tanımlama aşamasında, bu bilgi aşağıdaki T kullanıcı işlemine dönüştürülmektedir.

$$T = T_1, T_2, \dots, T_{|T|}$$

$$T_i = (KID_i, ZAMAN_i, URL_i, CODE_i, BYTE_i)$$

$$T_i \in T, i \in [1 \dots |T|]$$

Oturum Tanımlama

Kullanıcı oturumu sayfayı ziyaret eden kullanıcının etkileşimde olduğu süredeki yaşam süresidir. Web sunucundaki oturum nesnesinden daha geniş bir zaman aralığını ifade ettiği gibi uygulama tarafından değerlerle sonlanmamaktadır. Sadece etkileşim açısından oturumu değerlendirdiğimizde, aynı kullanıcıya ait ardışık iki isteğin aynı oturumda değerlendirilmesi açısından maksimum ne kadar süreye izin verileceği belirtilmelidir. Kullanıcının belirtilen maksimum zaman aralığını aşan iki ardışık isteği farklı oturumlarda değerlendirilmektedir. Bu zaman aralığı uygulamamızda 30dk. olarak alınmıştır. Bir oturum verisi T kullanıcı işlemleri cinsinden aşağıdaki şekilde ifade edilebilir.

$$\begin{aligned}
S &= S_1, S_2, \dots, S_{|S|} \\
S_i &= T_1 \cup T_2 \cup \dots \cup T_{|T|} \\
S_i &\in S, i \in [1 \dots |S|] \\
T_i &\in T, i \in [1 \dots |T|]
\end{aligned} \tag{3.1}$$

3.1.2. İçerik Verisi Hazırlama

Web içerik madenciliği için, yapısal olmayan ya da yarı-yapısal Web sayfalardan içeriklerin çıkarılması gerekmektedir. Bir sayfadaki içerik metinden, bağlantılardan, şekillerden ya da akan video verisinden oluşabilir. İçeriğin anlaşılmasında, incelenmesinde ve sınıflandırılmasında metin içerikleri yazılımlar tarafından daha kolay ve doğru işlenebilir olduğundan dolayı içerik madenciliğinde sadece metin tabanlı içerikler ele alınmıştır.

Bu tez çalışmasında, GenerateLink adı verdiğimiz program ile Web sunucusunda yayımlanan Web dokümanları alınarak, bu dokümanlardan kullanılabilir bilgilerin çıkartılması, hazırlanması ve veritabanına entegrasyonu işlemleri gerçekleştirilmektedir. Kullanılabilir bilgi olarak temelde metinsel içerikler alınmaktadır. Buna ek olarak; S. Chakrabarti ve diğ. (1999) çalışmalarında, bir sayfanın içeriğinin anlaşılmasında ona referans veren bağlantıların kullanılması tavsiye edilmektedir [41]. Bu bilgi ileriki aşamalarda sayfaların yaratılan ontolojide hangi sınıfa ait olduğunun belirlenmesinde kullanılabileceği düşünülmüş ve alınmıştır. Bu açıdan her sayfadaki bağlantıların, bağlantı metni ve bağlantı hedef sayfa URL'i kaydedilmektedir. Metin halindeki bağlantı hedef URL'in kullanabilmesi açısından sistemdeki sayfaların URL'leriyle birebir eşlenmesi de gerekmektedir, bu iş Bölüm 3.1.3'de anlatılmaktadır. Son olarak; M. Eirinaki ve diğ. (2003) çalışmasında sayfanın içeriğini sınıflandırmak açısından sayfanın başlığını kullanılabileceği belirtilmiştir [38]. Bu bilgi de yine kullanılabilir bir bilgi olarak sayfadan çıkartılması işlemi yapılmaktadır.

GenerateLink Modülü tek başına çalışan bir java [50] uygulamasıdır. Uygulama parametre olarak klasör ya da dosya fiziksel yolunu almaktadır. Klasör parametresiyle belirtilen klasördeki ve alt klasördeki(öz-yinelemeli olarak) tüm dosyaların listesi alınırken, dosya parametresiyle sadece belirtilen dosya

alınmaktadır. Diğer bir parametre olarak da sonuç dosyalarının ve entegrasyon dosyalarının yazılacağı klasörlerin fiziksel yolunu almaktadır.

Uygulama, içerik verisinin hazırlanmasını 3 aşamada gerçekleştirmektedir: İşlenmemiş Bilginin Çıkarılması, Bilginin Ön İşlenmesi ve Veritabanı Entegrasyonu.

İşlenmemiş Bilginin Çıkarılması

GenerateLink modülünde üzerine çalışılacak dosyaların listesi kaynak klasör ya da dosya adı ile verilmektedir. Oluşturulan işlenecek dosya listesinde Web içeriği olmayan resim, film, sunum gibi dosyalar da mevcut olabilmektedir. Bu tür dosyalar, dosya uzantısına bakılarak anlaşarak listeden çıkarılmaktadır. Tez çalışmamızda, kullandığımız veri kümelerindeki analizimize göre .htm, .html, .txt ve uzantısız dosyaların dışındaki dosyalar listeden çıkarılmaktadır.

Bir Web dokümanı; HTML etiketlerinden, istemci tarafında çalışan komutlardan (script) ve gömülü nesnelerden(flash, akan video) oluşabilmektedir. Bölüm başında belirtildiği gibi araştırmamızdaki içerik madenciliği açısından sadece metin tabanlı içerikler yani salt metinler, bağlantılar ve sayfa başlığı bilgileri kullanılacaktır. Bu açıdan sayfadaki HTML etiketlerin ayrıştırılması gerekmektedir. Bu amaç için HTMLEditorKit adı verilen bir java Uygulama Programlama Arayüzü (API) kullanılmaktadır. Bu API, HTML etiketlerini içeren bir dosya okuyucu nesnesini alarak, algıladığı html etiketlerine uygun metodu çağırır. Örnek yordam imzası Şekil 3.1’de görülmektedir. Eğer dosya da HTML etiketleri bulunmuyorsa hata verilmekte bu da Web içeriği olmayan dosyaların uzantıya bakılmaksızın filtrelenmesini sağlamaktadır.

```
public void handleStartTag(HTML.Tag t, MutableAttributeSet a, int pos);  
public void handleEndTag(HTML.Tag t, int pos);  
public void handleText(char[] data, int pos);
```

Şekil 3.1: HTMLEditorKit tarafından çağırılan metot örneği

HTML etiketlerinden aşağıdaki bilgilerin çıkarılması sağlanmaktadır:

1. Sayfa Başlık Bilgisi

<Title> etiketinin içindeki değerde belirtilmektedir.

2. Sayfadaki Bağlantıların Bilgisi

Bağlantılarda görünen metin ve bu metine basıldığında gidilen sayfanın yolu bilgileri bulunmaktadır. <A> etiketinin href özelliğinden hedef sayfanın yolu, etiketin içindeki değerle metin bilgisini sunmaktadır.

3. Sayfadaki Salt Metin İçeriğe ait Kelime Listesi

Özel anlamlı etiketler dışındaki(<script>,<embed>) etiketlerin içindeki değerler ya da salt metinler alınmaktadır. Metinlerdeki bazı alfabetik olmayan karakterlerde düzenli ifade süzgeçleriyle temizlenmektedir.

Aşağıdaki örnek Web dokümanına ait HTML etiket içeriğinden, yukarıdaki bilgilere ait numaralarla belirtilen bilgilerin elde edilmesine çalışılmaktadır.

```
<html>
<head>
<title>ITU Main Page</title> (1)
</html>
<body>
<table>
....
<a href="(2)anasayfa.htm">
(2)(3)Main Page</a> (2)
....
Faculties (3)
- Electric& Electronic Faculty (3)
```

Bilginin Ön İşlenmesi

Bu tez çalışmasında, içerik madenciliği Web sayfalarının oluşturulacak ontolojiye yansıtılması sırasında uygulanmaktadır. Bu yansıtılma işlemi dosyaların ait olduğu kavramlara sınıflandırmasıyla yapılmaktadır. Bu işlem bir doküman sınıflandırma görevi olarak görülmektedir ve kullandığımız sınıflandırma yöntemi açısından öğrenme kümesindeki dosyaların kök olarak ifade edilen kelime listesi ve bunların frekansı kullanılmaktadır. Bu açıdan işlenmemiş bilgilerin çıkarılması sırasında elde edilen kelime listesi, dosya sınıflandırma işlemi öncesinden ön işlenmesi gerekmektedir.

Web alanlarında hatta tekil bir Web sayfasında birden fazla dil desteği ya da çeşitli dillere ait içerik bulunabilmektedir. Bu durumda içerik madenciliğinin yapılabilmesi için her dile ait içerik saptanmalı ve eğer mümkünse farklı dillerdeki bu içerikler ortak bir içeriğe yansıtılmalıdır.

Tez çalışmamızda kullandığımız örnek veri kümelerinde hem farklı sayfalarda hem de aynı sayfa içerisinde İngilizce ve Türkçe içerik bulunabilmektedir. Bu nedenle her

dosya için İngilizce ve Türkçe kelime listeleri oluşturulmaktadır. Bu noktada, her kelimenin ait olduğu dilin belirlenmesi gerekmektedir. Doküman sınıflandırma yöntemimiz için gereken frekans bilgisi sözcük kökleri üzerinden hesaplanmaktadır. Bu açıdan her dile ait kelime listesi, o dile ait kök ayrıştırıcı tarafından ön-işlenmesi de gerekmektedir. Bununla beraber, tek bir doküman sınıflandırıcının olması için farklı kelime listelerindeki içeriğin ortak bir içeriğe dönüştürülmesi gerekir. Yalnız hem başarımlar hem de boyutsal olarak bunun yapılması yerine her dile ait ayrı bir doküman sınıflandırıcı oluşturulması tercih edilmiştir. Her dile ait sınıflandırıcı öğrenme kümesindeki sınıfların kendi dilindeki kelime listesini kullanmaktadır. Bu nedenle her Web dokümanının hangi dile ait olduğu belirlenmektedir. (İşlem-4 ve 5). Burada belirtilen ön-işlemlere ait uygulama detayları şu şekildedir:

İşlem-1: Her kelimenin ait olduğu dilin bulunması

Bir kelimenin hangi dile ait olduğunu anlamak için kullanılan dillere ait test yordamları kullanılarak bu yordamlardan dönen değerler normalize edilerek en yüksek değerdeki dile atanması yapılabilir. Tez çalışmamızda kullanılan veri kümeleri İngilizce ve Türkçe içeriğe sahip olabilmektedir. Türkçe için bize dil işleme araçları sağlayan Zemberek API'si kullanılmıştır.[26,27] Bu API içerisindeki DilTesti yordamı aşağıdaki Şekil 3.2'de gösterilen kolay bir kullanımı bulunmaktadır.

```
int result=zemberek.dilTesti(kelime);
```

Şekil 3.2: Zemberek DilTesti Yordamı Kullanım Şekli

DilTesti yordamına iletilen kelime ya da kelime listesinin Türkçe dil durumu, bulanık mantıkla ifade edilen aşağıdaki değerlerle belirtmektedir. İki den büyük dönüş değerleri Türkçe kelime olarak ele alınmaktadır.

İngilizce için ise dil test yordamını sunan kullanışlı bir API bulunamadığından, baştaki karşılaştırmalı yaklaşımımız değişmiştir. Test veri kümelerinde iki dile ait içerik olduğundan Türkçe dil testi yordamından Türkçe olmadığı belirtilen kelimelerin İngilizce olduğu varsayımı yapılmaktadır.

Bu aşamada her kelime için elde edilen DilTesti skoru sayfa dil sınıflandırılmasında, kök ayrıştırılmasında ve bu sınıflandırmanın başarımının ölçülmesi sırasında kullanılacaktır.

Tablo 3.4: DilTesti Yordamı Bulanık Mantıksal Dönüş Değerleri

Skor	Skor Açıklaması
0	Türkçe Kelime Değil
1	Genel itibariyle Türkçe Değil
2	Türkçe kelime ama çokça yabancı kök ya da bozulmalar var
3	Türkçe kelime ama birkaç yabancı kök ya da bozulmalar var
4	Kesinlikle Türkçe Kelime

İşlem-2: Her dile ait kelime listesinin oluşturulması

İşlem-1’de elde edilen dil test değerlerine göre; sayfada bulunan kelimeler ait oldukları dile ait kelime listelerine dağıtılırlar.

İşlem-3: Her kelimenin köklerine ayrıştırılması

Doküman sınıflandırıcı olarak kullanacağımız Naive Bayes Sınıflandırıcı için, kelime ve kelimelere ait frekans bilgisinin hesaplanması gerekmektedir. Kelimeler içerisindeki ekler frekans hesabını yanıltmaktadır, bu açıdan her kelime bir kök ayrıştırma işlemine tabi tutulmaktadır. İşlem-2 de oluşturulan her dile ait kelime listesine, o dile ait kök ayrıştırma algoritması uygulanmaktadır.

Türkçe kelimeler için Zemberek API sindeki KelimeCozumle yordamı kullanılırken [26,27], İngilizce kelimeler için PorterStemming algoritması uygulanmaktadır [33].

İşlem-4: Sayfa Dil Sınıflandırması

Web dokümanları, tanımlanan ontoloji kavramlarına sınıflandırılırken; her dile ait bir sınıflandırıcının oluşturulmaktadır. Her dile ait sınıflandırıcı oluşturulurken, o dile ait öğrenme kümesinde yer alacak dosyaların belirlenmesi gerekmektedir. Bu açıdan sayfalar web alanında kullanılan dil seçeneklerine göre sınıflandırılmaktadır. Türkçe ve İngilizce olarak iki farklı dile sahip web alanları için; S web alanındaki doküman listesini, w_i de dokümanda bulunan bir kelimeyi ifade ediyorsa, j ’de her dokümana ait sınıflandırma yöntemi formülize edilmiştir.

$$S = \{s_1, s_2, s_3, \dots, s_{|S|}\}$$

$$S_i = \{w_1, w_2, w_3, \dots, w_{|w_{Len}|}\}$$

$$\begin{aligned}
& \forall Si \in S, i \in [1 \dots |S|]; \\
& Skor_i = \sum_{i=1}^{wLen} zemberek.dilTesti(wi) / wLen \\
& SayfaDili = \begin{cases} Skor_i \geq 2; Türkçe \\ Skor_i < 2; İngilizce \end{cases}
\end{aligned} \tag{3.1}$$

İşlem-5: Sayfa Dil Sınıflandırması Ölçümü

İşlem-4'te gerçekleştirilen sayfa dil sınıflandırılmasının başarımı, sayfanın içerisindeki skor dağılımıyla anlaşılmaktadır. Bu açıdan her sayfaya ait bu dağılım kaydedilmektedir.

3.1.3. Eşleştirme İşlemleri

Farklı modüllerle, farklı yerlerden elde edilen bilgilerin beraber kullanılabilmesi açısından bu bilgiler eşleştirme işleminin yapılması gerekmektedir. Tez çalışmamızda iki adet eşleştirme işlemi yapılmaktadır: İçerik ve Oturum Bilgilerinin Eşleştirilmesi, Bağlantı Hedef URL'in Sayfayla Eşleştirilmesi.

İçerik ve Oturum Bilgilerinin Eşleştirilmesi

Bölüm 3.1.1'de oluşturulan oturum verisiyle Bölüm 3.1.2'de oluşturulan içerik verisi sayfa bazından eşleştirilmesi gerekmektedir. Oturum verisindeki sayfa sunucu izlerinden alınan URL bilgisiyle adlandırılırken, içerik verisindeki sayfa sunucu üzerinde yayımlanan dosya ile adlandırılmaktadır. Bu adlandırmalar arasından eşleştirme aşağıdaki URL üzerinde yapılan bazı işlemlerle sonrasında gerçekleştirilmektedir.

- (1) Sayfa içi bağlantı(inline) kısmının kaldırılması (Örn: URL#**documentClass**)
- (2) Aynı yeri işaret eden URL eklentilerinin kaldırılması (Örn: URL/**e/./**)
- (3) İçeriği değiştirmeyen istek parametrelerinin kaldırılması (Örn: RL&**sort_order=reverse**)
- (4) Gidilen sayfayı değiştirmeyen ek karakterlerin kaldırılması
(Örn: www.ce.itu.edu.tr/ ⇔ www.ce.itu.edu.tr
ya da http://www.ce.itu.edu.tr ⇔ www.ce.itu.edu.tr)

Bağlantı Hedef URL'in Sayfayla Eşleştirilmesi

S. Chakrabarti ve diğ. (1999) çalışmalarında, bir sayfanın içeriğinin anlaşılmasında ona referans veren bağlantıların kullanılması tavsiye edilmektedir [41]. Bu açıdan sayfadaki kaydedilen bağlantılardaki hedef özelliğinde belirtilen URL bilgisi; (1) Web alanındaki içerisindeki bir sayfaya yönelmiş, (2) Web alanı dışındaki bir sayfaya yönelmiş, (3) istemci tarafında çalışan komutu yönelmiş olabilir. Amacımız (1) numaralı yönelimlerin sayfalarla eşleştirilerek, hangi sayfadaki bağlantının hangi sayfaya hangi bağlantı metniyle referans verdiğinin bulunmasını sağlamaktır.

3.2. Anlamsal Web Yapısı Oluşturmak

Anlamsal içerikli bir öneri sisteminde amacımız, sadece geçiş istatistiklerine bakarak önerme yerine sayfaların içerikleri de dikkate alınarak en yakın anlamsal ilişki içinde olduğu sayfaların önerilmesidir. Buradaki anlamsal ilişki, sayfaların ait olduğu kavramlar arasındaki ilişkiyi göstermektedir. Bu açıdan Web alanına ait kavramların çıkarılması ve bu kavramlar arasında hiyerarşik olan ve olmayan ilişkilerin bulunduğu Anlamsal Web yapısı oluşturulması gerekmektedir. Bu yapının oluşturulmasında iki temel aşama bulunmaktadır: Ontoloji oluşturmak ve sayfaları ontolojiye yansıtmak.

3.2.1. Ontoloji Oluşturmak

Ontolojinin yaratılması Anlamsal Web açısından hayati ve zor bir iş olduğundan yapılan çalışmalar açısından motive edici olmaktadır. Ontoloji öğrenmede, bir Web alanında bulunan dağınık bilgi ve ilişkilerin öğrenilmesi gerekmektedir. Ontoloji öğrenme Bölüm 1.1'de belirtildiği şekilde Doğal Dil İşleme teknikleriyle ve Web dokümanları üzerinde yapılacak anlamsal açıklamalarla (semantic annotation) yapılabilmektedir. Bunun dışında aynı içeriğe sahip diğer Web alanlarından elde edilmiş, kullanıma açık, belirli bir olgunluğa sahip, kabul gören ontolojilerde bir başlangıç noktası kabul edilerek de bir öğrenme süreci başlatılabilmektedir.

Bu tez çalışmasında, boyut olarak geniş, farklı dil içeriklerine sahip, dağınık ve karmaşık veri kümeleri üzerinde çalıştığımızdan dolayı Bölüm 1.1'de belirtilen yöntemlerden iyi sonuç alınamayacağı düşüncesiyle kullanıma açık ontolojilerin kullanılmasına karar verilmiştir. Basit HTML Ontoloji Uzantıları (SHOE) ontoloji paylaşım ortamında [42] yayımlanan "University Ontology" ontolojisi [28] ITU veri

kümesi için kullanılırken, “Computer Science Department Ontology v. 1.1” ontolojisi [29] CS veri kümesi için kullanılmıştır.

Ontoloji öğrenilmesinde başlangıç olarak alınan yukarıdaki ontolojiler, veri kümesindeki kavramlara göre kırılmakta ya da genişletilmektedir.

Genişletilme işlemi Web alanına ait uzman tarafından Protégé ontoloji editörü [43] kullanılarak yapılmaktadır. Ontolojinin genişletilmesi Bölüm 3.1.2’de elde edilen kelime listeleri ve Web alanına ait dosya klasör yapısı kullanılarak yapılmaktadır. Hiyerarşik ilişkiler referans ontolojide bulunduğu gibi ontoloji editöründe eklenecek yeni kavramların bir hiyerarşiye yerleştirilmesi sırasında da belirtilmektedir. Hiyerarşik olmayan ilişkiler, ontoloji oluşturma aşamasında tanımlanabileceği gibi Bölüm 3.2.2’de yapılan ontolojiye yansıtılma işlemi sonrasında da oturum bilgilerindeki sayfa geçişlerinin, kavramlar arasındaki geçişe dönüştürülmesiyle elde edilerek ontolojiye yansıtılabilir. S oturum bilgisini, T kullanıcı işlem sayfasını, C’de kullanıcı işlem sayfasının yansıtıldığı ontoloji kavramını ifade ediyorsa;

$$S = T_1(C_{T_1}) \cup T_2(C_{T_2}) \cup \dots \cup T(C_T) \mid T \mid$$
$$T_i \in T, i \in [1 \dots |T|];$$

Aşağıdaki örnek S oturumuna bakıldığında;

$$S = T_1(C_{T_1} = Student) \cup T_2(C_{T_2} = Student) \cup T_3(C_{T_3} = Lecturer) \cup T_4(C_{T_4} = Lecturer)$$

T₂ ile T₃ sayfaları arasındaki öğrenci-öğretmen ilişkisi görülebilmektedir. Bu ilişki, ontolojiye bu aşamada da bakılarak eklenebilir. Eklenen hiyerarşik olmayan ilişkiler sayesinde yapılan önerilerin nasıl bir anlamsal ilişkiyi yansıttığını da açıklayabiliyor olacağız.

Kırılma işlemi ise referans alınan ontolojilerde bulunan ama veri kümelerinde rastlanılmayan kavramların kaldırılmasını içermektedir. Bu kavramlara yansıtılan sayfalar olmayacağı gibi bu kavramlara ait bir önerme de gerçekleştirilmeyecektir. Bunun dışında ontoloji ağacındaki aynı seviyedeki yaprak sınıflara yansıtılmış hiçbir sayfa bulunmadığında bu yaprak kavramlar ontolojiden çıkarılmaktadır.

3.2.2. Sayfaları Ontolojiye Yansıtma

Bölüm 3.2.1’de ontoloji oluşturulduktan sonra burada tanımlı kavramlara ait üye sayfaların belirlenmesi gerekmektedir. Her sayfa ontoloji yapısındaki en yakın sınıfa atanılması gerekmektedir ve her sayfa sadece bir sınıfın üyesi olması düşünülmektedir. Buradaki varsayımımız; her sayfanın içerik olarak belirli bir kavramı ağırlıklı olarak işlediği düşünülmektedir.

Sayfaların kavramlara atanmasında Bölüm 3.1.2’de ele alınan sayfaya ait sıklıkla geçen kelimeler, sayfayı referans veren bağlantılar ve sayfa başlığı kullanılmaktadır. Doküman sayısının az olduğu Web alanlarında (Örn:İTÜ) bu atamalar elle yapılabilenken; doküman sayısının çoğaldığı Web alanlarında, öğrenme kümesinin yeterli olduğu sınıflar için doküman sınıflandırma algoritmaları kullanılmaktadır. Sınıflandırma yapılabilecek sınıf sayısının fazla olabileceği düşünülerek, ilk aşamada en üst sınıflar arasında sınıflandırma yapılmaktadır. Üst sınıflara sınıflandırılan sayfaların, alt sınıflara dağıtılması sırasında yeterli örnek veri kümesi her alt sınıf için bulunabiliyorsa yeni bir sınıflandırma yapılmaktadır yoksa bu sayfalar alt sınıflara Protégé ontoloji editörü kullanılarak son sınıfa üye olarak atanmaktadır.

Bölüm 3.1.2’de belirtildiği üzere Web alanında birden fazla dile ait içerik olması durumunda eğer bu içerikleri ortak kavramlara dönüştüremezsek her dile ait sınıflandırıcı oluşturmamız gerekmektedir. Bu dönüştürme yapılamadığından Bölüm 3.1.2’de sayfa dil sınıflandırılması yapılmıştır. Sınıflandırma yapılacak örnek “Person” kavramı için hem Türkçe hem de İngilizce sınıflandırıcı oluşturulacağından, bu dillere ait dokümanlar içerisinden öğrenme kümeleri oluşturulmaktadır.

Tez çalışmamızda; sayfaların birbirinden farklı kelime sayısı fazla olmadığı gözlemlenmiştir. Bu açıdan öğrenme kümesinde yer alacak doküman sayısı artırılarak 50 olarak seçilmiştir. Ayrıca test veri kümelerimizde her sayfada bulunan portal menüleri ve navigasyon bağlantılarına ait kelimeler temizlenmiştir.

Doküman sınıflandırma işlemi aşağıda tanıtilen Naïve Bayes Sınıflandırıcı tarafından gerçekleştirilmiştir.

Naïve Bayes Sınıflandırıcı

Naïve Bayes Sınıflandırıcı, Bayes teoremini temel alan, olasılığa dayalı etkili ve önde gelen sınıflandırıcılardan biridir. Bayes teoremi gereğince; koşulların hedef sınıfa bağlılık olasılıkları hesaplanmaktadır yalnız naïve koşullu bağımsızlık varsayımı kabul edilerek koşulların kendi aralarında bir bağıllığı olmadığı düşünülmektedir [30]. Buna göre koşullu olasılık ifadesi (3.2)'deki gibi olmaktadır.

$$p(C | F_1, \dots, F_n) = p(C) \prod_{i=1}^n p(F_i | C) \quad (3.2)$$

Doküman Sınıflandırma problemi için, verilen D dokümanının C sınıfına ait olma koşullu olasılığı (3.3)'deki denklemle ifade edilmektedir [30].

$$p(C | D) = \frac{p(C)}{p(D)} p(D | C) \quad (3.3)$$

(3.3)'deki ifadesinin sağındaki koşullu olasılık (3.4) ifadesiyle hesaplanmaktadır. Buradaki w_i öğrenme verisetindeki i. kelimeyi ifade etmektedir [30].

$$p(D | C) = \prod_i p(w_i | C) \quad (3.4)$$

3.3. SemRec Öneri Sistemi

SemRec Öneri Sistemi, önerilerin Anlamsal Web'de oluşturulan istatistiksel olarak ağırlıklandırılmış ilişkileri baz almaktadır. Bu nedenle Bölüm 3.2'de Web alanlarına dair ontoloji oluşturulup, sayfaların buradaki kavramlara yansıtılması gerçekleşmiştir.

Son halini alan bir ontoloji, sayfalar arasındaki bağlantıların hangi anlamsal ilişkiyi ifade ettiğini belirtebilecektir. Anlamsal ilişkiler, sayfalar arasındaki geçiş olasılıklarıyla ağırlıklandırılmıştır. Bu yaklaşım ile, öneri sistemi anlamsal ilişkiyi açıklarken, bu ilişkinin ziyaretçiler tarafından izlenmesine göre kuvvetli/zayıf olduğu belirlenebilmekte ve bu bilgiler ekranlarda kullanıcılara sunulabilmektedir. Bunun dışında sayfa bazından ifade edilen oturum bilgisi, bunun dışında bir de sayfalara ait

kavramlarla ifade edilmekte ve bu ifade şekliyle oturum içinde izlenen anlamsal ilişkiler analiz edilebilmektedir.

Bu tez çalışmasında, iki aşamalı bir öneri sistemi önerilmektedir. İlk aşamada sınıf bazındaki oturum dizileri, sayfaların ait olduğu kavramlarla ifade edilerek kavram dizisi haline getirilmektedir. Oluşturulan kavram dizileri üzerinden; geçiş olasılıkları baz alınarak kavram bazlı bir öneri oluşturulmaktadır. İkinci aşamada, birinci aşamada önerilen kavramlara ait sayfalar üzerinden geçiş olasılıkları baz alınarak öneri oluşturulmaktadır. Bu şekilde bir öneri sistemiyle, anlamsal ilişkinin kuvvetli olduğu sayfalar önerilmekte ve ayrıca öneri oluşturulacak küme daraltıldığından hem zamandan hem de bellek kullanımından kazanç sağlanacağı beklenmektedir.

Oluşturulan modellerden üretilen sonuçlar, gerçekleşen istatistiksel öneri sistemiyle tüm metrikler kullanılarak karşılaştırılmaktadır. Bunun dışında önerilen modeller başarımları ve kapsama oranları bakımından, aynı veri kümeleriyle çalışan diğer başka öneri modelleriyle de karşılaştırılmıştır.

SemRec Öneri Sistemi tek başına çalışan bir java uygulamasıdır. Programa ait akış ve adımlar Şekil 3.3'deki gibidir:

Adım 1. Ara Tabloların Oluşturulması

Programın ilk çalıştırılmasında bazı ara tablolar veritabanında oluşturulmaktadır. Yaratılan ara tablolar sorgu karmaşıklığını azaltmak program veri yapısına uygun az aşamada tüm gereken bilginin elde edilmesini sağlamak için oluşturulmaktadır. Bu aşamada iki alt aşama bulunmaktadır.

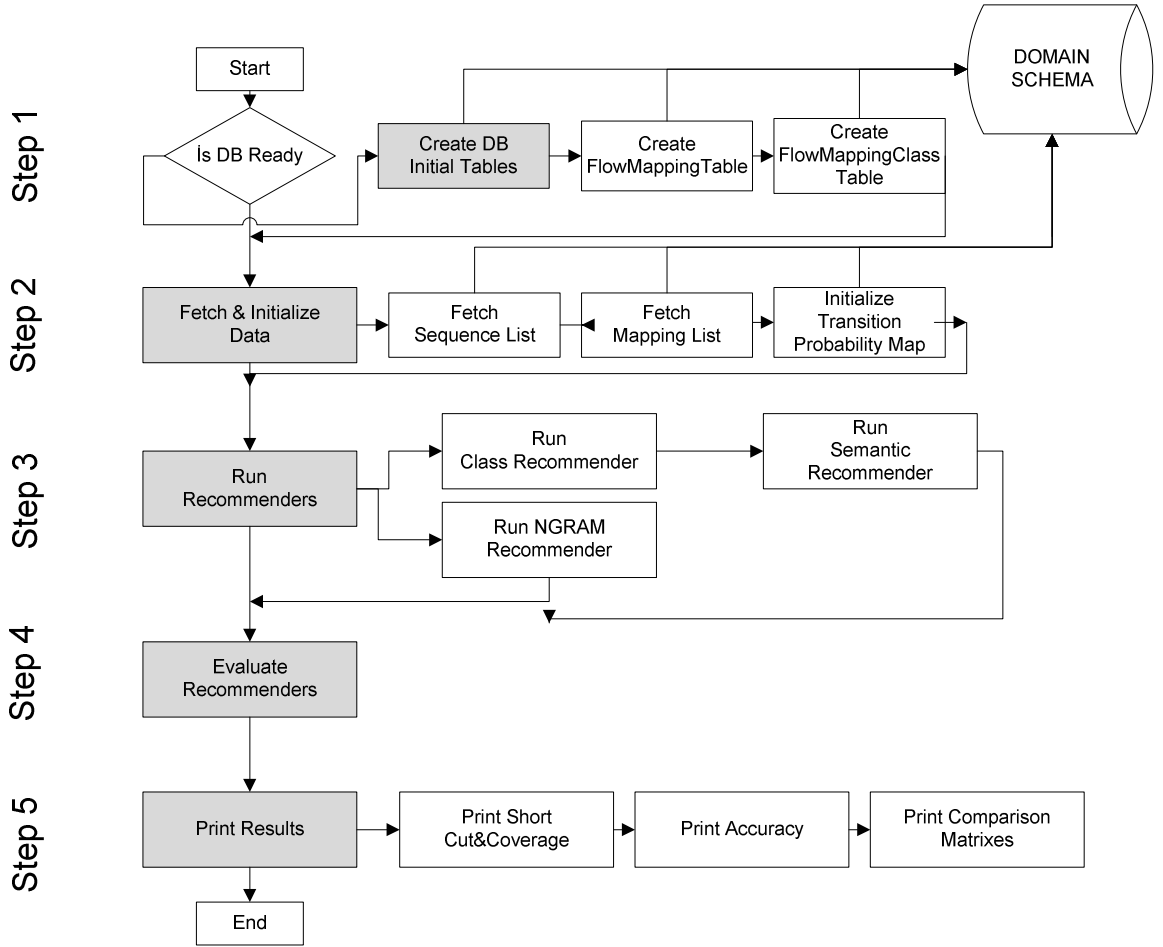
Adım 1.1 Sayfa gezim haritasının çıkarılması

Bölüm 3.1.1'de oluşturulan oturum bilgileri veritabanında her satırda oturuma ait bir sayfa gezimini içermektedir. Satır bazında verilen oturuma ait gezimler, maksimum sütun bilgisi sistemden alındıktan sonra kolon bazında gösterilimi yapılmaktadır. Bu şekilde oluşturulan ara tabloda satır bazında bir oturuma ait tüm gezimler kolon bazından görülebilmektedir.

Adım 1.2 Sınıf gezim haritasının çıkarılması

Sınıf bazlı öneri sisteminde, sayfaların ait olduğu kavram dizi üzerinden öneri oluşturulmaktadır. Bu dizinin oluşturulması program tarafından yerine veritabanında yapılması performans kazandıracak düşünülmektedir. Bu açıdan Adım 1.1'den sonra

oluşturulan ara tablo ve sayfaların ait olduğu kavram bilgisinin tutan tablolar birleştirilerek, kolon bazından oturum ve kavram bilgisinin içeren yeni bir ara tablo oluşturulmaktadır.



Şekil 3.3: SemRec Programına Ait Akış Diyagramı

Adım 2 Program ilk değer atamasının yapılması

Program akışı içinde gereken bazı kaynak tablo bilgileri uygulama hafızasına alınarak, uygulama içinde genel kullanılması amaçlanmaktadır. Bu aşamada üç alt aşama bulunmaktadır.

Adım 2.1 Oturum bilgisinin alınması

Hem öneri modellerinin eğitilmesinde hem de sonuçların test edilmesi açısından oturum bilgisi kullanılacağından, tüm veri program belleğine alınmaktadır. Modellere ait eğitim ve test kümelerinin rasgele olarak oluşturulması veritabanında değil bellekte yapılmaktadır.

Adım 2.2 İçerik ve oturum bilgisine ait eşleşme bilgisinin alınması

Bölüm 3.1.1’de elde edilen oturum bilgisiyle, Bölüm 3.1.2’de elde edilen içerik bilgisinin beraber kullanılması açısından Bölüm 3.1.3’de eşleştirilmeleri yapılmaktadır. Bu eşleştirme programda da kullanılabilir olması gerektiğinden program belleğine eşleşme tablosu da alınmaktadır.

Adım 2.3 Geçim olasılıklarının hesaplanması

Öneri modellerinin kullanması açısından bigram olasılıklar hesaplanmaktadır. Hesaplanan bu olasılıklar program içerisindeki HashMap veri tipinde kaynak sayfa ve hedef sayfa ikilisi şeklindeki anahtarlarla tutulmaktadır.

Adım 3 Öneri Sistemlerinin Çalıştırılması

Öneri modelleri öğrenme ve test üzere iki aşamada çalışmaktadır. Öğrenme aşamasında her modelin ihtiyacı olduğu geçiş olasılıkları, Adım2’de ayrılan öğrenme kümesi üzerinden hesaplanmaktadır. Test kümesinde bulunan oturum bilgileri üzerinden de her öneri adımında belirtilen sayıda sıralı tahmin listesi oluşturmaktadır. Bu listenin boyutu çalışmamızda dört olarak seçilmiştir.

Adım 3.1 Sınıf önerici modelinin çalıştırılması

Sınıf öneri modeli öğrenme ve uygulama olmak üzere iki aşamada çalışmaktadır. Model öğrenme aşamasında; Adım1.2’de oluşturulan sınıf gezim haritasını kullanılmaktadır. Bu tabloya ait bilgiler program belleğine yüklenerek Adım2.3’de kavramlar arası geçim olasılıkları hesaplanmaktadır. Bu olasılıklar hesaplandıktan sonra modelin uygulama aşamasında geçilmektedir. Test kümesindeki sayfa bazında oturum bilgisi öğrenme kümesi gibi kavram bazında ifade edilen bir dizi haline getirilmektedir. Her bir dizi elemanı için kavram bazındaki öneri listesi, diğer kavramlarla arasındaki geçiş olasılığına bakılarak oluşturulmaktadır. Sınıf önericisine ait öneriler direk olarak kullanıcıya yansıtılabileceği gibi, bir sonraki adımda oluşturacağımız anlamsal önericiye de aktarılarak, sayfa bazından öneri oluşturulması sağlanabilmektedir.

Adım 3.2 Anlamsal önerici modelinin çalıştırılması

Anlamsal öneri modeli öğrenme ve uygulama olmak üzere iki aşamada çalışmaktadır. Model öğrenme aşamasında; Adım1.1’de oluşturulan gezim haritasını kullanılmaktadır. Bu tabloya ait bilgiler program belleğine yüklenerek Adım2.3’de

sayfalar arasındaki geçim olasılıkları hesaplanmaktadır. Bu olasılıklar hesaplandıktan sonra modelin uygulama aşamasına geçilmektedir. Uygulama aşamasında, diğer sayfa öneri modellerinden farklı olarak; sadece sınıf önerici modelden öneri listesinde olan kategoriye ait sayfalar baz alınmaktadır. Bu şekilde bulunan sayfayla en kuvvetli ilişkideki kavrama ait sayfalar kullanılarak, içerik madenciliği sonucu kullanılmaktadır. Bununla beraber önerinin oluşturulmasında; geçiş olasılığına göre karşılaştırılan sayfaların sayısı azaldığından daha hızlı öneri oluşturulurken, bellekten de kazanç sağlanmaktadır.

Adım 3.3 N-GRAM öneri modelinin çalıştırılması

Önerdiğimiz öneri modeli, hızlı sonuç elde edebileceğimiz istatistiksel N-GRAM öneri modeliyle karşılaştırılması düşünülmüştür. N-GRAM, n eleman içeren pencerelerin sekans üzerinde kaydırılmasıyla elde edilen sekanslar listesidir. N-Gram olasılığı denklem (3.5)'de görüldüğü gibi n. elemana kadarki tüm koşullu olasılıkların çarpımıyla ifade edilmektedir.

$$p(w_1, w_2, w_3, \dots, w_{N-1}, w_N) = p(w_1)p(w_2 | w_1)p(w_3 | w_1, w_2)p(w_4 | w_3, w_1, w_2) \dots p(w_N | w_1, w_2, w_3, \dots, w_{N-1}) \quad (3.5)$$

Markov kabulüne göre; önceki tüm kelimeleri almak yerine, sadece bir önceki kelimeye bağlı koşullu olasılığın alınmaktadır. Denklem 3.5'deki ifade bu varsayımla denklem 3.6'ya dönüşmektedir.

$$p(w_1, w_2, w_3, \dots, w_{N-1}, w_N) = p(w_1)p(w_2 | w_1)p(w_3 | w_2)p(w_4 | w_3) \dots p(w_N | w_{N-1}) \quad (3.6)$$

Tez çalışmasında; N-Gram Modeli için Markov kabulüne göre (3.6) olasılığı hesaplanmaktadır. Yalnız bu değer çok küçük olacağı nedeniyle her ifadenin logaritması alınarak değer okunabilir hale getirilmektedir. Uygulamada tüm N-GRAM olasılıkları hesaplanırsa da, kolaylık açısından öneri modelinde bigram olasılığı kullanılmaktadır.

Adım 4 Öneri Sistemlerinin Sonuçlarının Değerlendirilmesi

Adım3'te çalıştırılan her öneri sistemi; test kümesinde bulunan her sekans içindeki ilk eleman dışındaki her sekans elemanı için 4 adet öneri üretmektedir. Sayfa öneren

modeller için sayfa numarası, sınıf öneren model için sınıf bilgisi aşağıdaki şekilde öneri listesine eklenmektedir.

[SekansNumara][SekansElemanNo][ÖneriNo] <= Önerilen sınıf/sayfa

Her öneri modeli tarafından oluşturulan yukarıdaki dizi şeklindeki öneri listesi, test sekansı ile karşılaştırılmaktadır. Öneri listesiyle aynı yapıda olan ama başarılı tahminler için 1, olmayanlar için 0 olarak atanan sonuç listesi oluşturulmaktadır. Bu şekildeki bir liste yapısı Adım5'te hesaplanacak sonuç ve metrikler için uygun bir format sunmaktadır.

Adım 5 Sonuçların Oluşturulması

Adım 4'te oluşturulan önerilerin gerçekleşme durumunu belirten sonuç listesine bakılarak; öneri modellerinin süreleri, vuru sayıları, kıs ayol kazancı ve kapsama oranları ölçülmekte ve öneri modelleri arasında karşılaştırma matrisleri oluşturulmaktadır. Bu ölçümlerle ilgili detaylar aşağıdaki alt adımlarda verilmektedir:

Adım 5.1 Vuru sayısı

Bir öneri sisteminin başarısı, yapılan önerinin gerçekleşme sayısı olan vuru sayısı ile basit bir şekilde ölçülebilir. Adım 4'te oluşturulan sonuç listesinde, her oturum adımı için yapılan dört adet önerinin başarı durumları 1 ya da 0 olarak belirtilmektedir. Dört önerinin kendi içinde önerilme sırası bulunmaktadır ve ek bir başarı kriteri olarak, hangi sıradaki önerinin gerçekleştiği ayrıca ölçülmektedir.

Adım 5.2 Kısa Yol Kazanç Oranı

Öneri sisteminin başarısı sadece bir sonraki adım tahminiyle değil, yapılan tahminin ileriki adımlarda gerçekleşmesiyle de ölçülebilir. Kısa yol kazancı denilen bu ölçüm, yapılan önerinin oturum içerisindeki ileriki adımlarda gerçekleşmesi durumunda kaç adım kazanç sağlandığını belirtmektedir.

Örneğin elimizde a, b, c, d, e şeklinde bir oturum listesi olsun. Öneri sistemi b sayfasındayken e'yi önerdiğinde iki adım kazanç sağlamaktadır. Önerilen sayfanın ara bir sayfa olması ya da zaman harcanan bir sayfa olması sonucu değiştirir mi? Eğer aynı oturum listesi içerisinde, önerilen sayfa birden fazla kere bulunduğu durumda ölçüm nasıl yapılacaktır?

İlk soruya cevap olarak; sayfalar kullanıcının bakış açısından anahtar (key) ve yedek sayfa (auxiliary) olarak sınıflandırılmaktadır. Bu sınıflandırma kullanıcının sayfada harcadığı zamana bakılarak yapılmaktadır. Sayfada harcanan zaman, Bölüm 3.1.1’de oturum verisinden elde edilmektedir. Tez çalışmasında, Anahtar – yedek sayfa ayrımının yapılması açısından 30sn.’lik bir zaman seçilmiştir [31].

İkinci soruya cevap olarak ise; öneri yapılan sayfanın anahtar-yedek sayfa sınıflandırılması yapıldıktan sonra, anahtar sayfa için ilk gözlenilen, yedek sayfa için ise en son gözlenen oturum elemanı kısa yol kazancının hesaplanılmasında kullanılmaktadır.

Öneri sisteminin yardımıyla oluşturulan kısaltılmış oturum s' ve S oturum listesindeki gerçek oturum s olarak nitelendirilirse, kuvvet kazancı aşağıdaki formülden hesaplanmaktadır

$$KısayolKazancı = \frac{\sum_s \frac{|s| - |s'|}{|s|}}{S} \quad (3.7)$$

Adım 5.3 Kapsama Oranı

Yapılan önerilerin kullanıcı tarafından gezilme ihtimallerinin oranı kapsama oranı olarak nitelendirilmektedir. Bu ölçüm; sınıflandırma sonucunda yapılan geri-çağırım (recall) ölçümüne benzemektedir. Her kullanıcı oturumu s , oturumdaki her sayfa p olarak nitelendirilirse, öneri modeli tarafından oluşturulan öneri listesi $R(p)$ olarak gösterilebilir. $R(p)$ listesi, s dizisindeki p sayfasından sonraki kısmıyla karşılaştırılmakta ve buna $T(p)$ adı verilmektedir. Bu durumda kapsama oranı aşağıdaki denklem (3.8)’le hesaplanmaktadır. [31]

$$KapsamaOranı = \frac{\sum_s \frac{|\bigcup_p (T(p) \cap R(p))|}{|\bigcup_p (T(p))|}}{S} \quad (3.8)$$

Adım 5.4 Karşılaştırma Matrisleri

Her öneri modeli için Adım 4’te oluşturulan sonuç listeleri birbirleriyle karşılaştırılarak karşılaştırma matrisleri oluşturulmaktadır. Bu matrislerde öneri modellerinin birbirinden farklı başarılı önermeleri gözlenebilmektedir. Buradaki fark fazla ise modeller arasında hibrid öneri sisteminin kazanç getireceği

düşünülmektedir. Ayrıca anlamsal öneri modelinin eksiklerini anlamak açısından; bu farklı önermeler incelenerek sınıf tipi bazında analiz yapılmaktadır.

Adım 5.5 Zamanlama Ölçümü

Öneri modelleri için çalışılan veri seti büyüdükçe öneri üretme zamanı ve bellek kullanımı önem kazanmaktadır. Bu açıdan her öneri modeline ait öğrenme aşaması geçilip uygulanma aşamasında gelindiğinde zaman ölçümü başlatılır ve öneri listesi oluşturulduğunda zaman ölçümü sonlandırılmaktadır. Anlamsal öneri modeli, sınıf öneri modelinin sonucunu kullandığı için; bu modelin zamanlama ölçümüne sınıf öneri modelinin süresi eklenmiştir.

4. DENEYSEL SONUÇLAR

Bu bölümde, Bölüm 3.3'te tanıtılan SemRec Öneri Sisteminin oluşturulması aşamaları ve veri kümeleri üzerindeki önermelere ait sonuçlar sunulmaktadır. Sonuçların genel geçerliliği, boyut ve içerik olarak farklı iki Web alanı olan İstanbul Teknik Üniversitesi (İTÜ) ve İTÜ Bilgisayar Mühendisliği (CS)'ne ait veri kümeleri üzerinde sınanmıştır. Aşağıdaki sonuç başlıklarında her iki veri kümesi için de sonuçlar verilmektedir.

4.1. Veri Hazırlama Sonuçları

Bölüm 3.1'de aktarılan veri hazırlama işlemleri olan oturum verisi hazırlama, içerik verisi hazırlama ve eşleştirme sonucunda elde edilen ölçümler her veri kümesi için aşağıda verilmiştir.

4.1.1. Oturum Verisi Sonuçları

Web sunucudan elde edilen oturum bilgisinde, ziyaret edilen sayfa ve sayfada harcanan zaman bilgisi bulunmaktadır. Bu bilgilere ek olarak oturum ait temel istatistiksel bilgiler ölçülmüştür. Oturum Sayısı (OS), erişim bilgilerinde yapılan oturum belirleme işleminden sonra elde edilen farklı oturum sayısını içermektedir. Oturum URL Sayısı(OUS) ise web alanındaki toplam farklı URL sayısını belirtmektedir. Ziyaret Sayısı (ZS), veri temizleme işleminden sonra; erişim bilgisinde bulunan toplam kullanıcı işlem sayfa sayısını belirtmektedir. Erişim bilgisindeki farklı URL sayısı Ziyaret Edilen Farklı URL Sayısı(ZUS) olarak alınırken; bu değerin OUS ile orantısı, bize kaynak erişim bilgisinin web alanını kapsama oranını verecektir. Bu değerle beraber kullanıcının sayfada ortalama harcadığı süre ve oturumun uzunluğuyla ilgili ölçümler Tablo 4.1'de yer almaktadır. Tablo 4.1'de her iki veri kümesinde de; ziyaret edilen sayfaların, Web alanındaki tüm sayfalara göre çok küçük bir kısmını içerdiğini görmekteyiz. Bunun temel nedeni; alınan erişim bilgilerinin, web alanına ait tüm sayfaları içerecek kapsamda uzun dönemde bir kaydının alınmamasıdır. Bununla beraber; CS veri kümesinde, sayfada harcanan zaman sağlıklı olmadığından ölçülememiştir. Oturum uzunluğuna

bakıldığında, ortalama ve minimum uzunluklar birbirine yakın olduğundan dolayı genellikle oturum uzunluğu kısa diyebiliriz.

Tablo 4.1: Oturum Verisi Hazırlama Sonuçları

Sonuç Metrikleri	İTÜ Veri Kümesi	CS Veri Kümesi
Oturum Sayısı (OS)	5,308	3,484
Ziyaret Sayısı (ZS)	28,765	23,476
Oturum URL Sayısı (OUS)	1005	11872
Ziyaret Edilen Farklı URL Sayısı (ZUS)	141	460
ZUS/OUS (%)	14.03	3.87
Ort. Sayfada Harcanan Zaman(sn.)	158.97	Bilgi yok
Ort. Oturum Ziyaret Uzunluğu	5.92	6.74
Maks. Oturum Ziyaret Uzunluğu	15	15
Min. Oturum Ziyaret Uzunluğu	4	4

4.1.2. İçerik Verisi Sonuçları

Web sayfalarından içerik verisi olarak; sayfa başlığı, bağlantılar ve salt metin alınmaktadır. Sayfa bağlantıları, Web alanı içerisindeki doğal bağlantıların nasıl yapıldığını, bağlantıda bulunan hedef sayfa (href) bilgisiyle anlamaktayız. Hedef sayfa, Web alanımız içindeki ya da dışındaki bir sayfayı işaret ediyor olabilir. Buna göre hedef sayfa iç ya da dış olarak sınıflandırılır. Bunun dışından iç URL’ler açısından; URL bilgisi “Açık Hedef” olarak verilebileceği gibi (Örn: “http://www.ce.itu.edu.tr/calisanlar/sgunduz”) “/calisanlar/sgunduz” şeklinde “Rölatif Hedef” olarak da verilebilir. Bu tanımlamalara göre iki veri kümesine ait bağlantı ölçümleri Tablo 4.2 gibidir.

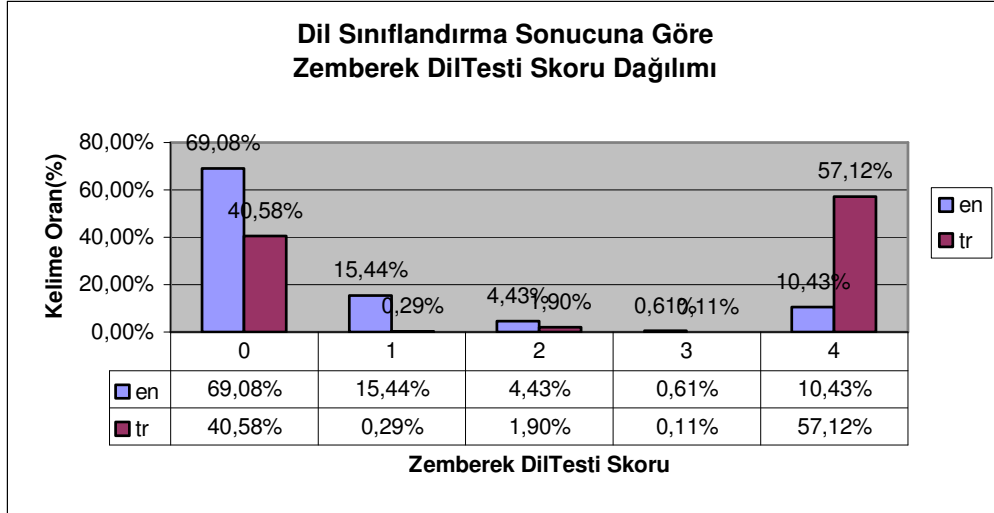
Tablo incelendiğinde, iki veri kümesinde de “Açık Hedef” tanımlamaları çoğunlukla bulunmaktadır. Bir başka açıdan da iç hedef sayısının fazla olması eşleştirilmesi gereken bağlantı sayısının fazla olduğunu göstermektedir. Ayrıca bağlantı sayısı ile sayfa sayısı arasında çok katlı bir fark bulunmaktadır. Bu duruma da genelde sayfalarda bulunan bağlantı içeren menüler sebep olmaktadır.

Tablo 4.2: Sayfalardaki bağlantı bilgileri

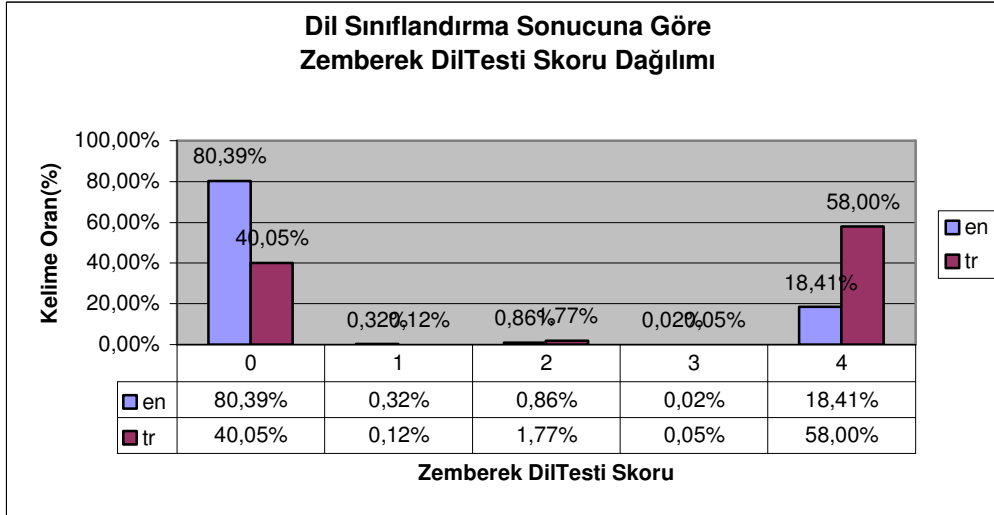
	ITU Veri Kümesi	CS Veri Kümesi
# Link Sayısı	169383	245248
# Farklı Hedef Url Sayısı	7741	13572
# Açık Hedef İç Bağlantı	7431	12254
# Açık Hedef Dış Bağlantı	189	752
# Rölatif Hedef İç Bağlantı	112	369
# İç Satır Bağlantı	2	12
# JavaScript Bağlantı	4	150
# E-Posta Adres Bağlantı	3	35

Salt metin içeriklerinin içerik madenciliğinde kullanılabilmesi için kök ayrıştırılmasına tabi tutulması gerekmektedir ama veri kümemizde hem İngilizce hem de Türkçe kelimelerin olması nedeniyle kelimelerin öncelikle dil testinden geçirilmesi gerekmektedir. Veri kümelerimizde Türkçe ve İngilizce olmak üzere iki dile ait içerik olduğundan kolaylık olması açısından sadece Türkçe diline ait dil testi kullanılmakta, Türkçe olmayan kelimelerin İngilizce olduğu düşünülecektir. Türkçe dil testi için Zemberek API'sindeki DilTesti yordamı kullanılmakta ve bu yordam bize kelimenin Türkçe olup olmadığını Tablo 3.4'de belirtilen 0 ile 4 arasında değerlerle dönmektedir. Her kelime için yapılan dil testinin sonucuna göre dokümanın geneline bakarak bir dil sınıflandırılması yapılmaktadır. Bu sınıflandırma doküman düzeyindeki Türkçe ve İngilizce kelime bulunma oranlarına göre karar verilmektedir.

Yapılan dil sınıflandırmanın doğruluk analizi, sayfa içerisindeki dil testi skor dağılımına bakarak anlaşılmaya çalışılmıştır. Şekil 4.1'de İTÜ veri kümesi için Türkçe ve İngilizce olarak sınıflandırılan dokümanlara ait dil testi skorunun yüzdesel dağılımı gösterilmektedir. Aynı ölçüm CS veri kümesi için de yapılmış ve Şekil 4.2'de gösterilmektedir. CS veri kümesindeki İngilizce olarak sınıflandırılan dokümanlarda, İTÜ veri kümesine göre daha fazla Türkçe kelime bulunmaktadır (%18-%10). Bununla beraber; her iki veri kümesinde de Türkçe olarak sınıflandırılan sayfalarda, Türkçe olmayan kelimelerin yüzdesi 40%'larda seyretmektedir bu durumda Web sayfalarında hem İngilizce hem de Türkçe kelimenin birlikte bulunduğu anlaşılabilmektedir.



Şekil 4.1 ITU Veri Kümesindeki Dil Sınıflandırma Skor Dağılımı



Şekil 4.2 CS Veri Kümesindeki Dil Sınıflandırma Skor Dağılımı

4.1.3. Eşleştirme Sonuçları

Veri hazırlama işleminde iki eşleştirme işlemi yapılmaktadır:

Birinci eşleştirme farklı ortamlardan elde edilen içerik ve oturum bilgisindeki sayfaların birbirleriyle URL üzerinden eşleştirilmesidir. Her iki veri kümesi için eşleştirilen Oturum ve İçerik verilerine ait sayılar Tablo 4.3’de gösterilmektedir. Burada sınıflandırılmayan olarak belirtilenler, oturum verisinde bulunan ama içerik verisinin bulunamadığı sayfalardır. Tabloda da görüldüğü üzere eşleştirilen veriler arasında yaklaşık 1:4 – 1:5 bir oran söz konusudur. Bu şekildeki bir oran; web sayfasına birden fazla çağırım metoduyla ulaşılabilmesi ve uygulama ve web sunucusu tarafından bu yönlendirmenin olmasından kaynaklanmaktadır.

Örneğin: www.ce.itu.edu.tr/lisans ile www.ce.itu.edu.tr/lisans/index.html aynı sayfaya yönlendirilmektedir.

Sunucu erişim bilgisi, kullanıcının çağırım yordamını tuttuğundan bu şekildeki bir yönlendirme kaydedilememektedir.

Tablo 4.3 ITU ve CS Veri Kümelerindeki Oturum ve İçerik Veri Sayıları

		# Sayfa	# İng. Sayfa	# Tr. Sayfa	#Sınıflandırılmayan
ITU	Oturum Verisi	1005	445	30	530
	İçerik verisi	206	190	16	0
CS	Oturum Verisi	11872	8695	3177	0
	İçerik verisi	3736	2641	1095	0

İkinci eşleştirme, içerik bilgisi olarak alınan bağlantılardaki “Hedef Sayfa” bilgisinin sistemdeki Web sayfalarıyla eşleştirilmesidir. Burada sadece iç bağlantılar işleme alınmaktadır. Her iki veri kümesinde de yapılan bu işleme ilişkin sonuçlar aşağıdaki Tablo 4.4’de sunulmaktadır. Eşleştirme iki veri kümesinde de yüksek oranda yapılmıştır. Bu işlemle web alanına ait bağlantı çizgesi oluşturulabilmekteyken, öneri modellerinin sonuçlarına göre bu çizge üzerinde yeni bağlantılar oluşturulabilmektedir.

Tablo 4.4: ITU ve CS Veri Kümelerindeki Bağlantı Eşleştirme Sonuçları

	ITU Veri Kümesi	CS Veri Kümesi
İç Bağlantıların Toplamı	7543	12623
# Eşleştirme Yapılan Bağlantı	7391	12520
# Eşleştirme Yapılamayan Bağlantı	152	103

4.2. Anlamsal Web Yapısı Sonuçları

Anlamsal Web yapının oluşturulmasında iki aşama izlenmektedir: Ontolojinin oluşturulması ve sayfaların ontolojiye yansıtılmasıdır. Bu işlemler sırasında elde edilen sonuçlar aşağıdaki gibidir:

4.2.1. Ontoloji Oluşturma Sonuçları

Bölüm 3.2.1’de belirtildiği üzere, ontoloji oluşturma işlemine Web alanına ait kullanılabilir referans ontolojiler seçilerek başlanmıştır. Bu ontolojiler üzerinde ontoloji editör kullanılarak ekleme ve budamalar yapılmıştır. Ontoloji üzerine yapılan eklemeler veri kümelerinde bulunan özel kavramları içermektedir.

ITU Veri kümesine ait ontoloji Tablo A.2’de gösterilirken, CS Veri kümesine ait ontoloji Tablo A.3’de gösterilmektedir.

4.2.2. Sayfaların Ontolojiye Yansıtılması Sonuçları

Bölüm 3.2.12’de belirtildiği üzere, sayfaların ontolojiye yansıtılması, küçük olan İTÜ veri kümesi için, içerik bilgileri kullanılarak ontoloji editörü vasıtasıyla yapılmıştır. Daha büyük olan CS veri kümesinde ise 50’den fazla hem İngilizce hem de Türkçe öğrenim kümesi olan Worker, Course ve Announcement sınıfları için her dilin kendi kelime listesini kullandığı Naive Bayes Doküman Sınıflandırıcı oluşturulmuştur. İngilizce dokümanlar için %72, Türkçe dokümanlar için de %61 başarımla sınıflandırma yapılmıştır.

4.3. Öneri Sistemi Sonuçları

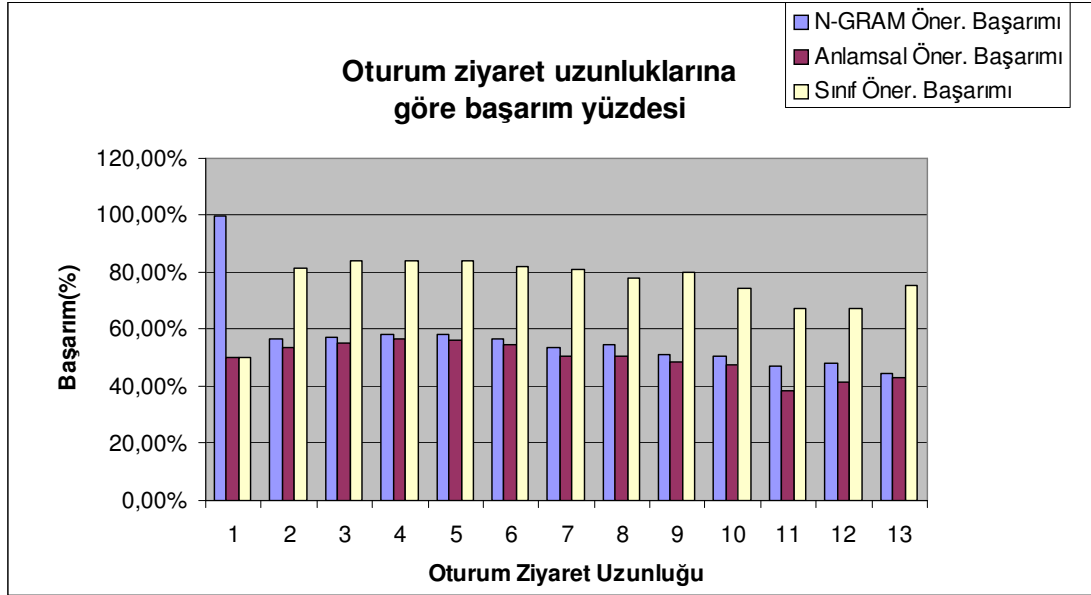
SemRec Öneri Sisteminde oluşturulan öneri modellerinin sonuçları N-GRAM öneri modeliyle karşılaştırılmaktadır. Karşılaştırmalar aşağıdaki sonuçlar üzerinden yapılmaktadır. Öneri ve sınaama kümeleri %70-%30 olarak ayrılmış ve sonuçların tutarlılığını anlamak açısından 10-kat çapraz geçерleme yapılmıştır.

4.3.1. Vuru Sayısı, Kapsama ve Kısa yol Kazanç Sonuçları

Bir öneri modelinin başarımı, bir sonraki adım için yapılan önerinin doğruluğıyla hesaplanan vuru sayısıyla belirlenebilir. İTÜ veri kümesi için vuru sayılarına bakıldığında, Sınıf önerici etkili bir oranı bulunurken, N-GRAM ve Anlamsal öneri modellerinde yakın sonuçlar elde edilmiştir (Tablo 4.5).

Tablo 4.5: İTÜ Veri kümesinde öneri modellerinin vuru sayıları

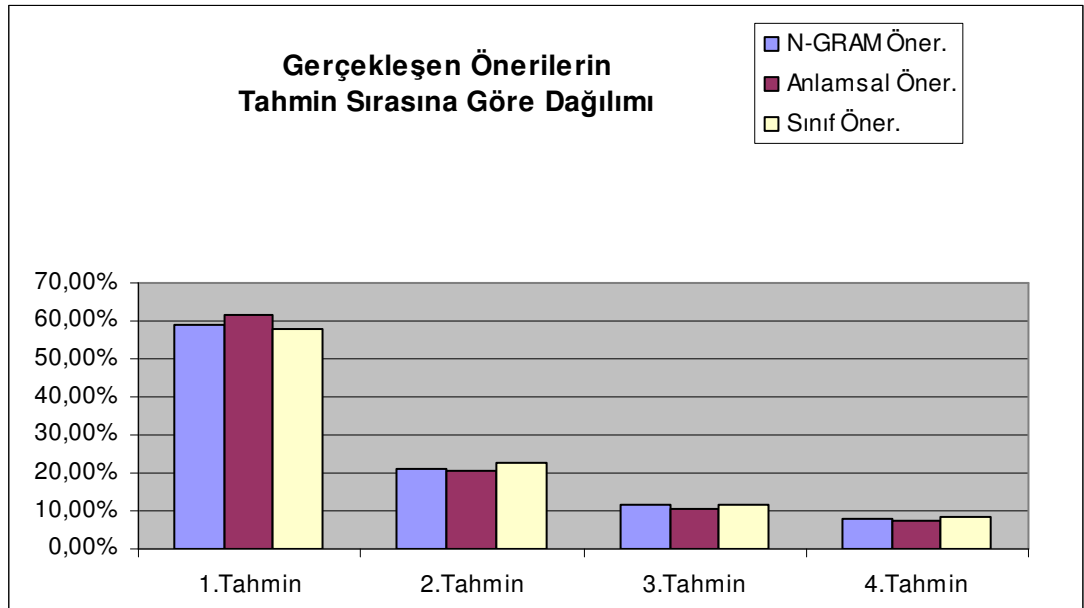
Öneri Liste Uzun.= 4 Test Küme = 30%	Vuru Sayısı Vuru Oranı	
N-GRAM Öner.	3744	%53.88
Anlamsal Öner.	3601	%51.82
Sınıf Öner.	5715	%82.25
Toplam	6948	%100.00



Şekil 4.3: İTÜ Veri kümesinde, oturum ziyaret uzunluklarına göre başarımlar yüzdesi

Oturumların uzunluğunun başarıma etkisi Şekil 4.3’de görülmektedir. Burada her oturum uzunluğu için, oturuma ait vuru oranlarının ortalaması alınmıştır. Buradan da görüldüğü üzere N-GRAM ve Anlamsal öneri modelleri daha uzun oturumlarda oranları azalmaktadır.

Şekil 4.4’te ise yapılan başarılı önerilerin sıra bazında dağılımı görülmektedir. Burada anlamsal öneri modelinin 1. tahminlerde diğer modellere göre daha başarılı olduğunu görebiliyoruz.



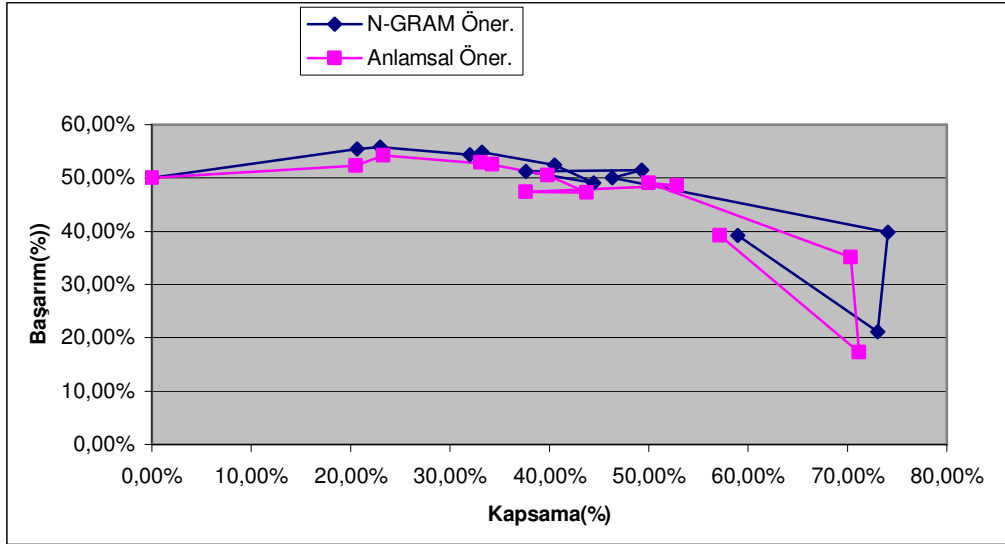
Şekil 4.4: İTÜ Veri kümesinde, Gerçekleşen önerilerin tahmin sırasına göre dağılımı

Öneri modelleri için diğer başarı kriterleri olarak kısa yol kazancı ve kapsama verilebilir. Kısa yol kazancı sayfa bazında karşılaştırma Tablo 4.6’ da verilmektedir. Burada görüldüğü üzere modeller arasında İTÜ veri kümesi için yakın sonuçlar elde edilmiştir.

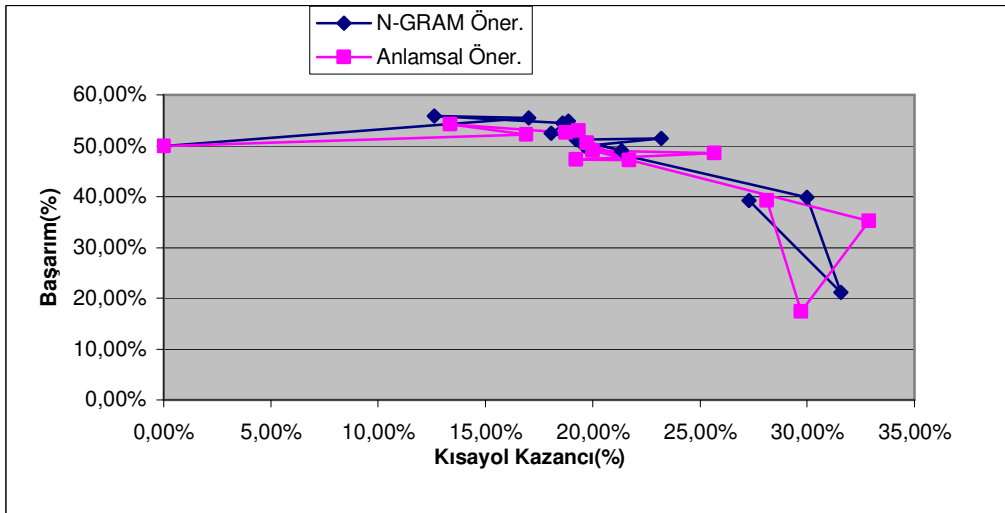
Tablo 4.6: İTÜ Veri kümesinde, öneri modellerinin kısa yol kazancı ve kapsama oranları

	N-GRAM ÖNER.	ANLAMSAAL ÖNER.
Toplam Kısa yol Sayısı	2071	2074
Toplam Kısa yol Durumu	384	385
Kısa yol Oranı	%18.54	%18.56
Kapsama Oranı	%29.80	%29,85

Şekil 4.5 ve Şekil 4.6’da; yukarıda ölçülen kısa yol ve kapsamanın başarımla ilişki grafikleri sunulmaktadır. Bu grafiklerde de görülebileceği gibi bu değerlerin, başarımla ters bir orantı çizdiği görülmektedir. Başka bir deyişle, başarımın arttığı yerde diğer başarımların azalmasıdır.



Şekil 4.5: İTÜ Veri kümesinde, Başarım X Kapsama oranı karşılaştırılması

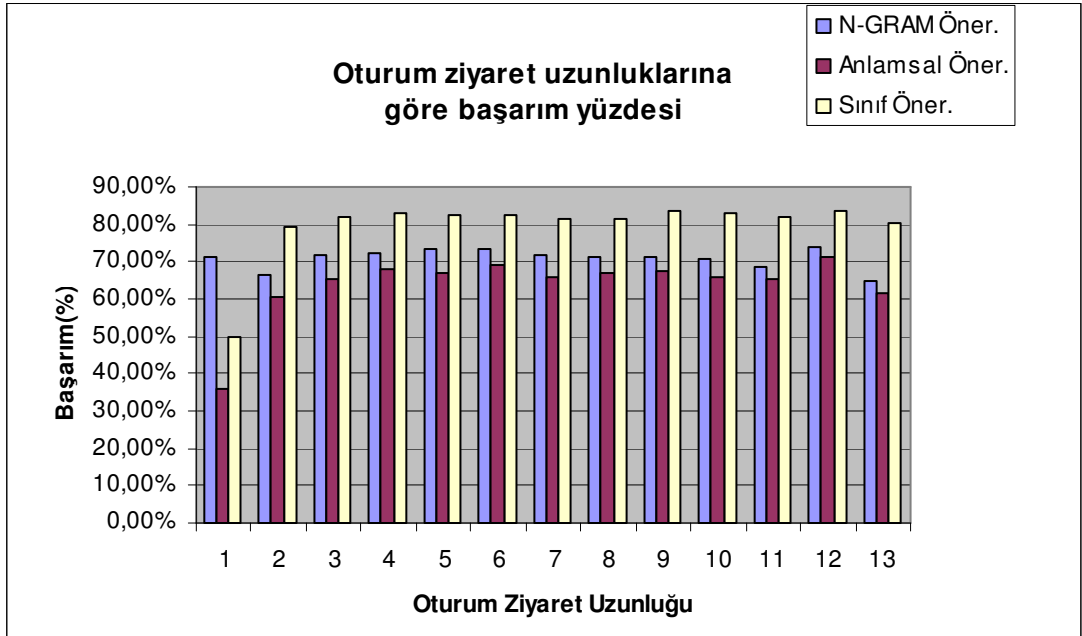


Şekil 4.6: İTÜ Veri kümesinde, Başarım X Kısa yol Kazancı karşılaştırılması

CS veri kümesi için vuru sayılarına bakıldığında, sınıf öneri modelinin etkili bir oranı bulunurken, N-GRAM ve Anlamsal öneri modellerinde yakın sonuçlar elde edilmiştir (Tablo 4.7).

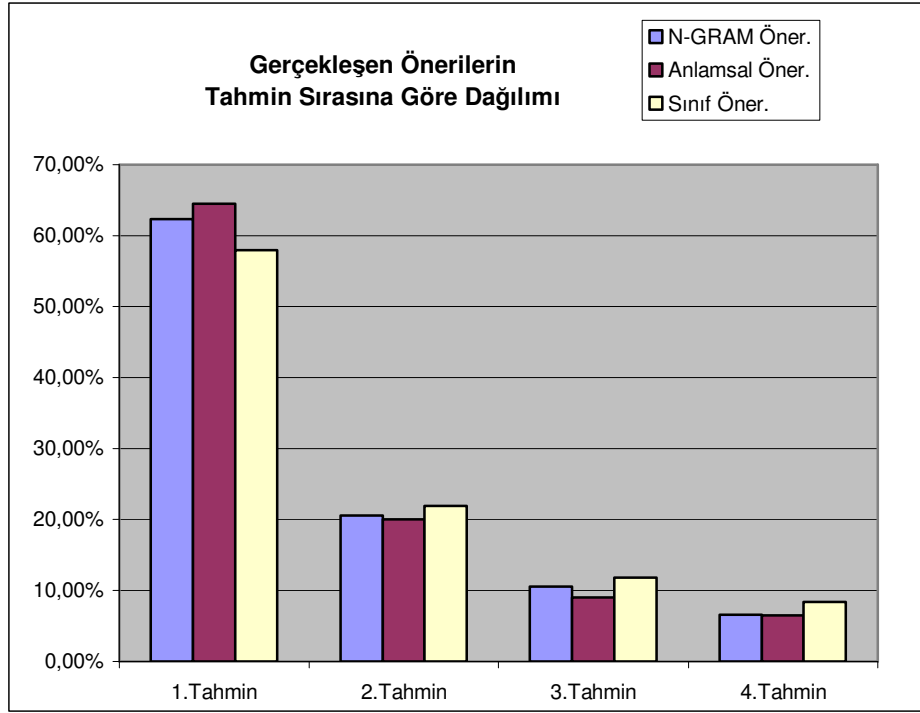
Tablo 4.7: CS Veri kümesinde öneri modellerinin vuru sayıları

Öneri Liste Uzun.= 4 Test Küme = 30%		
	Vuru Sayısı	Vuru Oranı
N-GRAM Öner.	3814	%63.32
Anlamsal Öner.	3621	%60.12
Sınıf Öner.	4937	%81.97
Toplam	6023	%100.00



Şekil 4.7: CS veri kümesinde, oturum ziyaret uzunluklarına göre başarımlar yüzdesi

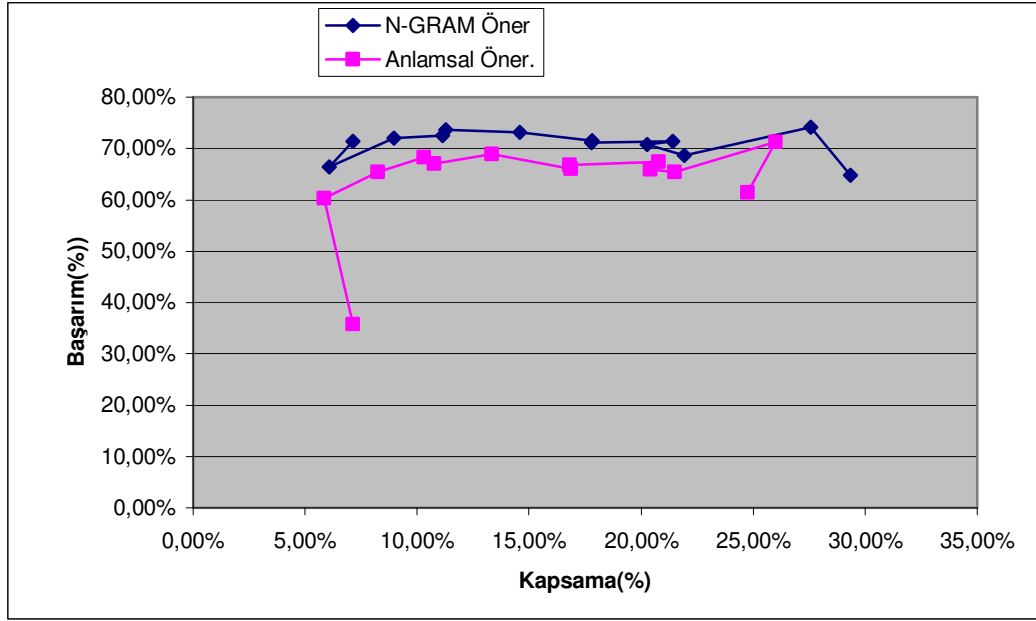
Öneri modelleri için diğer başarı kriterleri olarak kısa yol kazancı ve kapsama verilebilir. Kısa yol kazancı sayfa bazında karşılaştırma Tablo 4.8’de verilmektedir. Burada görüldüğü üzere modeller arasında CS veri kümesi için yakın sonuçlar elde edilmiştir.



Şekil 4.8: CS veri kümesinde, Gerçekleşen önerilerin tahmin sırasına göre dağılımı

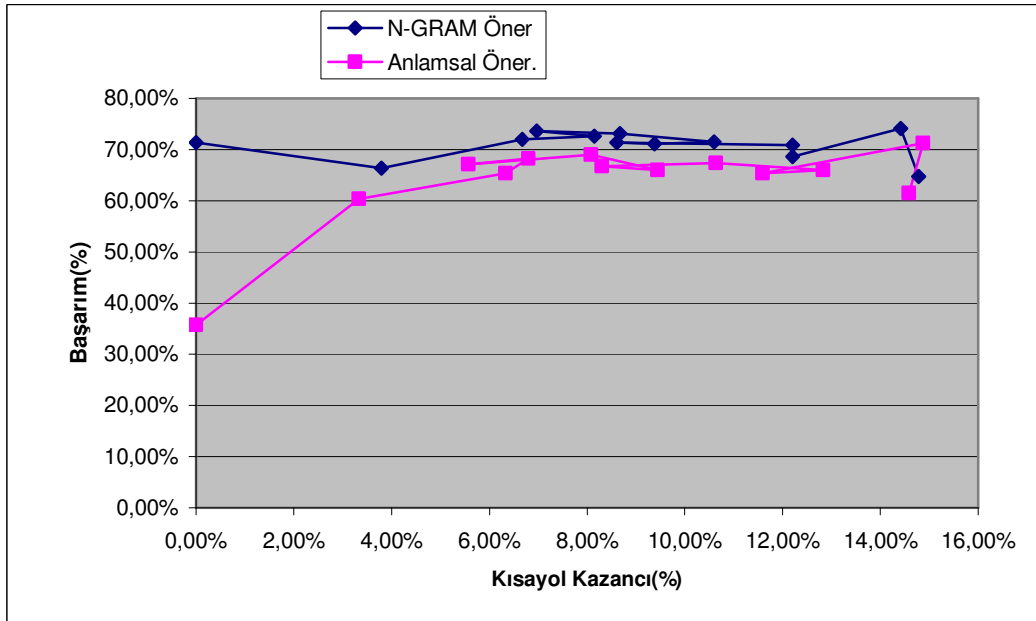
Tablo 4.8: CS veri kümesinde, öneri modellerinin kısa yol kazancı ve kapsama oranları

Anahtar sayfa bulunmamaktadır	N-GRAM ÖNER.	ANLAMSAL ÖNER.
Toplam Kısa yol Sayısı	944	893
Toplam Kısa yol Durumu	106	91
Kısa yol Oranı	%11.22	%10.19
Kapsama Oranı	%15.67	%14.82



Şekil 4.9: CS veri kümesinde, Başarım X Kapsama oranı karşılaştırılması

Ölçülen kısa yol kazancı ve kapsama oranının başarımla ilişki grafikleri Şekil 4.9 ve Şekil 4.10'da sunulmaktadır. Bu grafiklerde de görülebileceği gibi bu değerlerin, İTÜ veri kümesinde alınan sonuçların tersine, başarımla doğru bir orantı çizdiği görülmektedir. Başka bir deyişle, başarımın arttığı yerde diğer başarımların da artma olmaktadır.



Şekil 4.10: CS veri kümesinde, Başarım X Kısa yol Kazancı karşılaştırılması

4.3.2. Karşılaştırma Matrisleri

Oluşturulan modellerin birbirinden farklı ve aynı öneri üretme durumlarını anlamak açısından karşılaştırma matrisleri oluşturulmuştur. Buradaki pozitif-negatif ve negatif-pozitif hücrelerinde bulunan sayılar modeller arasındaki farklı tahminleri göstermektedir. Farklı tahminlerin fazla olması iki modelin birleştirilerek hibrid bir öneri sistemi yapılması açısından motive edici olacaktır. Bunun yanında, anlamsal öneri modelinin N-GRAM'ın doğru öneride bulunduğu sayfalar için, hangi sınıflarda ne kadar hata yaptığı sınıf içindeki orana bakılarak ölçülmektedir.

Tablo 4.9 'da İTÜ veri kümesinde, oluşturduğumuz modellerin N-GRAM modeliyle karşılaştırma sonuçları sunulmaktadır. Burada N-GRAM modeliyle anlamsal modelin yakın tahminler yaptığını görebilmekteyiz, öbür yandan sınıf önericisiyle N-GRAM modeli arasında hibrid sınıf önericisi oluşturmaya uygun sonuçlar elde edilmiştir.

Tablo 4.9: İTÜ Veri kümesi üzerinde, öneri modelleri arasındaki karşılaştırma matrisi

	N-GRAM ÖNER.	
ANLAMSAL ÖNER.	3016	331
	188	3413

	N-GRAM ÖNER.	
SINIF ÖNER.	901	332
	2303	3412

Tablo 4.10: Anlamsal Önericinin sınıf bazında hata yapma oranı

Sınıf İsmi	Hata Oranı
Course	16,16%
AdministrativeStaff	12,69%
Club	10,25%
Dean_Professor	10,10%
Faculty	10,08%
Foundation	9,90%
Organization	9,68%
EducationOrganization	8,85%
ResearchGroup	7,48%
Program	7,14%
Chair-Professor	6,97%

İTÜ veri seti için; anlamsal öneri sisteminin en fazla hangi sınıfta ve ne oranda yaptığı Tablo 4.9'da belirtilmiştir. En yüksek hata oranı kabul edilebilir düzeyde kaldığı görülmüştür.

Tablo 4.11 'de CS veri kümesinde, oluşturduğumuz modellerin N-GRAM modeliyle karşılaştırma sonuçları sunulmaktadır. Burada N-GRAM modeliyle anlamsal modelin yakın tahminler yaptığını görebilmekteyiz, öbür yandan sınıf önericiyle N-GRAM modeli arasında hibrid sınıf önericisi oluşturmaya uygun sonuçlar elde edilmiştir.

Tablo 4.11: CS Veri kümesi üzerinde, öneri modelleri arasındaki karşılaştırma matrisi

	N-GRAM ÖNER.	
	ANLAMSAL ÖNER.	SINIF ÖNER.
ANLAMSAL ÖNER.	2124	264
	83	3536

	N-GRAM ÖNER.	
	ANLAMSAL ÖNER.	SINIF ÖNER.
SINIF ÖNER.	822	250
	1382	3550

Tablo 4.12: Anlamsal Öneri Modelinin sınıf bazında hata yapma oranı

Sınıf İsmi	Hata Oranı
Undergraduate Not-department Lessons	18,81%
Information	17,65%
Undergraduate Courses	11,59%
Undergraduate Department Lessons	10,99%
Introductory information	10,59%
Exam	9,28%
Graduate Courses	8,71%
News	7,18%
Anouncement	6,67%
Undergraduate Elective Lessons	6,62%
Student announcement	6,48%
Event	5,88%

CS veri kümesi için; anlamsal öneri sisteminin en fazla hangi sınıfta ve ne oranda yaptığı Tablo 4.12'de belirtilmiştir. En yüksek hata oranı kabul edilebilir düzeyde kaldığı görülmüştür.

4.3.3. Diğer Web Öneri Modelleriyle Karşılaştırma

Oluşturulan anlamsal model önceki bölümlerde N-GRAM modeliyle karşılaştırılmaktadır. Bu bölümde; aynı veri kümeleri üzerinde çalıştırılan başka Web öneri modellerine ait sonuçlar sunulmaktadır. Tablo 4.13’de İTÜ veri kümesine ve Tablo 4.14’de de CS veri kümesine ait sonuçlar gösterilmektedir. Tüm veri kümelerinde kural bazlı önerici diğer modellere göre kötü sonuç vermektedir. Önerdiğimiz model, N-GRAM’a ve diğer modellere yakın sonuçlar üretmektedir.

Tablo 4.13: ITU Veri kümesi üzerinde, diğer Web öneri modelleriyle başarımların karşılaştırma

Web Öneri Model Adı	Vuru Yüzdesi(%)	Kapsama Yüzdesi(%)
N-Gram	53,88	29.8
Anlamsal	51,82	29.85
CST[44]	50	21
Markov	54,8	26
Kural Bazlı	35	19

Tablo 4.14: CS Veri kümesi üzerinde, diğer Web öneri modelleriyle başarımların karşılaştırma

Web Öneri Model Adı	Vuru Yüzdesi(%)	Kapsama Yüzdesi(%)
N-Gram	63,32	15.67
Anlamsal	60,12	14.82
CST[44]	60	21
Markov	58,6	26
Kural Bazlı	39	16

4.3.4. Öneri Zamanlama Sonuçları

N-GRAM öneri modelinde; bulunulan sayfadan, ziyaret edilebilecek tüm sayfalarla arasındaki bigram geçiş olasılığını hesaplamakta ve bunları kendi içinde sıralamaktadır. Bu arada anlamsal öneri modeli, sınıf öneri modeli tarafından önerilen kavramlara ait sayfalar üzerinden geçim olasılıkları baz alınarak öneri oluşturmaktadır. Bu şekilde öneride oluşturulacak küme daraltıldığından hem zamandan hem de bellek kullanımından kazanç sağlayacağı beklenmektedir. Tablo 4.15’de veri kümelerindeki modellere ait zamanlama sonuçları gösterilmektedir. Anlamsal Öneri modeli bir önceki modelden veri aldığından bu öneri modeli için zamanlama iki modelin toplamı olarak alınmaktadır. Bu durumda küçük boyutlu İTÜ veri kümesi için sonuçlar yakinken, CS veri kümesi için sonuçlar arasında 1:2’den fazla fark bulunmaktadır.

Tablo 4.15: Veri kümelerinde öneri zamanlama sonuçları

	İTÜ Veri Kümesi	CS Veri Kümesi
N-GRAM ÖNER.	23.5	292,60
ANLAMSAL ÖNER.	27.9(14.1+13.8)	105.44(9.73+95.71)
SINIF ÖNER.	13.8	9.73

5. SONUÇ ve İLERİ ÇALIŞMALAR

Bilindiği gibi Web madenciliği, Web dokümanlarından yararlı bilginin çıkarılmasını üç inceleme alanından yapmaktadır: Web İçerik, Yapı ve Kullanım Madenciliği. Öneri modelleri açısından, bu çalışma alanlarının birlikte kullanıldığı çalışmalar çok yönlü modeller oluşturulmasını sağlamaktadır. Bu açıdan önerdiğimiz öneri modelinde, hem kullanım hem de içerik madenciliği teknikleri kullanılmaktadır.

Veri madenciliğinde en temel işlem olan veri hazırlama işlemi İstanbul Teknik Üniversitesi (İTÜ) ve İTÜ Bilgisayar Mühendisliği Departmanı (CS)'na ait veri kümeleri üzerinde yapılmıştır. Bu Web alanlarındaki hem kullanım hem de içerik verisi alınmaktadır. Kullanım verisi, Web sunucusuna ait erişim dosyalarından çeşitli temizleme, önişleme yöntemleri uygulanmaktadır, ardından kullanıcı ziyaretleri belirlenmekte ve son olarak, iki ziyaret arası 30dk. lık bir zaman aralığı konularak oturum bilgileri oluşturulmaktadır. İçerik bilgisi ise Web alanında bulunan dokümanların işlenmesiyle elde edilmektedir. Web dokümanları uzantılarına ve HTML etiketi içermelerine göre anlaşılabilir, dokümanlardaki sayfa başlık, bağlantı ve salt metin bilgilerine ulaşılmaktadır. Burada elde edilen metinlerin kullanılması için metindeki kelimelerin köklerinin bulunması gerekmektedir. Yalnız kullandığımız veri kümelerinde İngilizce ve Türkçe içerikler hem aynı sayfa içerisinde hem de Web alanının farklı sayfalarında bulunmaktadır. Bu nedenle önce kelimelerin hangi dile ait olduğu anlaşılmalı ardından sayfaların dil sınıflandırılması yapılmaktadır.

Anlamsal Web teknolojisi, makine okunabilir olmayan Web dokümanlarının katmamsal olarak zenginleşen veri yapılarıyla işlenebilir hale getirmektedir. Tez çalışmasında; bu teknoloji karmaşık, yapısal olmayan, dağınık Web içeriğine sahip Web alanlarının; organize edilmiş, birbirleriyle ilişki içindeki kavramlar ağı haline getirilmesi sağlamıştır. Öneri modelleri için bu içerik altyapısı, sadece oturum bilgileri üzerinden yapılan öneri modelleri için zenginleştirici bir anlam katacağı düşünülmüştür ama buna karşın; karşılaştırılan sadece erişim istatistiklerini kullanan

N-GRAM öneri modeli ile yakın başarımlar ve tahmin grafikleri çizmişlerdir. Ama buna rağmen üretilen öneriler için ilişkiler bazında açıklanabilir bir ortam oluşturulmaktadır.

Anlamsal Web Yapısının oluşturulmasında; ontoloji öğrenme ve sayfaların ontolojiye yansıtılması işleri bulunmaktadır. Ontolojinin öğrenilmesi Doğal Dil İşleme ve yapay zeka yöntemleriyle yapılabilirken; bu adımsal öğrenme süreci yerine Web alanlarına uygun kabul görmüş ontolojiler başlangıç olarak seçilmesi ve bu ontolojiler üzerinde geliştirme yapılması hem zaman hem de kapsam bakımından fayda sağlamaktadır. Tez çalışmasında, İTÜ ve CS'ye ait veri kümeleri için yayımlanan kabul görmüş ontolojiler baz alınmıştır. Bu ontolojiler üzerinde yapılan ekleme ve çıkarma işlemleri, Web alanına ait yapılan analizlere göre ontoloji editörü kullanılarak gerçekleştirilmiştir.

Ontoloji yapısı oluşturulduktan sonra, sayfaların ontolojide ait oldukları kavramların belirlenmesi işlemi yapılmaktadır. Bu işlem İTÜ veri kümesi gibi az sayıda doküman içeren Web alanları için elle yapılabilirken, CS veri kümesinde 3000'i aşan doküman sayısı nedeniyle elle yapılamamaktadır. Elle sınıflandırma sırasında sayfalardan elde edilen başlık, frekansı yüksek kelimeler ve bu sayfayı gösteren bağlantılara ait metin bilgisi kullanılırken; otomatik sınıflandırmada ise yeterli öğrenme kümesi olan sınıflar için; Naive Bayes doküman sınıflandırıcı kullanılmaktadır. Web alanında birden fazla dile ait içerik olduğu durumlarda her dil için ayrı bir sınıflandırıcı oluşturulmaktadır. Veri kümesindeki içeriklerde sayfalar arasında tekrar eden kelimeler filtrelenmesinden sonra geriye kalan kelimeler kullanılmaktadır. Kelime listesinin azalması nedeniyle de öğrenme kümesi için seçilen dosya sayısı arttırılmıştır. Tez çalışmasında her dile ait 50 öğrenme dosyası çıkarabilen temel sınıflar için doküman sınıflandırma tekniği kullanılırken, diğer sınıflar için elle atama yapılmıştır.

Oluşturulan anlamsal ağdaki ilişkiler oturum bilgileriyle ağırlıklandırılarak sınıf bazından öneri üreten bir model oluşturulmuştur. Bu modelin ürettiği önermeler, kendi başına kullanılabildiği gibi sayfa bazından önerme yapmak için de kullanılabileceği düşünülmüştür. Sayfa bazından önerme yapan anlamsal öneri sistemi, önerilen sınıflara ait sayfalar üzerinden geçiş olasılığına göre sıralı öneri listesi oluşturmaktadır. Bu şekilde daraltılmış sayfa listesi kullanılarak; hem ağırlıklı kavramsal ilişkiye sahip olduğu sayfalar arasından önerme yapılmakta ve bu şekilde

içeriksel ilişkisi az olan sayfalar elimine edilmekte; hem de zaman ve bellek açısından büyük veri setleri için fayda sağlamaktadır.

Oluşturulan öneri modelleri, basitlik açısından sadece bigram geçiş olasılığının kullanıldığı N-GRAM öneri modeliyle karşılaştırılmıştır. Sınıf bazından öneri üreten model her iki veri kümesinde de %80 üzerinde etkili bir başarımla elde etmektedir, bu hem kendi başına kullanılabilirliğini gösterirken; anlamsal öneri modelinin bir sonraki aşamada kullanılmak üzere sınıf bilgisini hangi başarıyla beslediğini görebilmekteyiz. Anlamsal öneri modeli, N-GRAM modeline öneri başarımları açısından çok yakın sonuçlar elde etmiştir. Önerilen model, kısmi olarak kısa yol kazancı ve doğru tahminlerin daha öncelikli tahminlerden elde edilmesi gibi artılar sunmakta ayrıca büyüyen veri kümelerinden süre ve bellek açısından ciddi kazançlar sağlamaktadır.

İleri çalışma olarak, elle gerçekleştirilen prosedürler hem iyileştirilebilir hem de otomatize edilebilir. Bu açıdan Ontoloji oluşturulmasında, sayfalardaki içeriğin anlamsal açıklamalarının yapılarak kavramların otomatik olarak belirlenmesi sağlanmalı ve bu kavramlar üzerinden oluşturulacak ontolojiye, sayfaların daha başarılı şekilde yansıtılması sağlanabilir. Öneri modelleri açısından da, anlamsal öneri modeli başka öneri modelleriyle birlikte hibrid bir yapıda değerlendirilebilir ayrıca öneri oluşturmada geçim olasılıklarının yerine ilişki kuralları kullanılabilir.

KAYNAKLAR

- [1] **Pitkow, J.; Krishna, K. Bharat.,** 1994. Webviz: A tool for world-wide Web access log analysis, First International WWW Conference
- [2] **Maarek, Y. S.; Shaul, I.Z. Ben,** 1996. Automatically organizing bookmarks per content, Proc. of 5th International World Wide Web Conference
- [3] **Zhao, Q.; Bhowmick, S. S.,** 2006. Sequential Pattern Mining: A Survey, Nanyang Technological University, Singapore
- [4] Semantic Web RoadMap,
<http://www.w3.org/DesignIssues/Semantic.html>
- [5] Semantic Web Activity,
<http://www.w3.org/2001/sw/>
- [6] RDF/XML Syntax Specification
<http://www.w3.org/TR/rdf-syntax-grammar/>
- [7] OWL Web Ontology Language
<http://www.w3.org/TR/owl-features/>
- [8] Integrating Applications on the Semantic Web
<http://www.w3.org/2002/07/swint>
- [9] **Manola, F.; Miller, E.,** 23 January 2003. "RDF Primer." W3C Working Draft
- [10] **DuCharme, B.; Cowan, J.,** October 2002. "Make your XML RDF-Friendly", O Reily Media
- [11] **Lee, T. Berners; Hendler, J.; Lassila, O.,** May 2001. "The Semantic Web.", Scientific American
- [12] Resource Description Framework
<http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>

- [13] DAML+OIL Reference Description, March 2001.
<http://www.w3.org/TR/daml+oil-reference>
- [14] OWL Web Ontology Language Overview, February 2004
<http://www.w3.org/TR/owl-features/>
- [15] **Omelayenko, B.**, 2001, Learning of Ontologies for the Web: the Analysis of Existent Approaches, Proceedings of the International Workshop on Web Dynamics
- [16] **Hearst, M.A.**, 1992, Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th International Conference on Computational Linguistics. Nantes, France.
- [17] **Cimiano, P.; Hatho, A.; Staab, S.**, 2005, Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis
- [18] **Sánchez, D.; Moreno A.**, 2006, Discovering Non-taxonomic Relations from the Web. IDEAL 2006: 629-636
- [19] **Gulli, A.; Signorini, A.**, 2005, "The Indexable Web is more than 11.5 billion pages" <http://www.cs.uiowa.edu/~assignori/Web-size/>
- [20] **Goldberg, K.; Roeder, T.; Gupta D.; Perkins, C.**, 2001, A Constant Time Collaborative Filtering Algorithm, Kluwer Academic Publishers, pp 135-151
- [21] **Perkowitz, M.; Etzioni, O.**, 1998, Adaptive Web sites: Automatically synthesizing Web pages, Fifteenth National Conference On Artificial Intelligence, pp. 727- 733
- [22] **Yan, T.; Jacopsen, W.; Garcia-Molina, M.; Dayal, H. U.**, 1996, From user access patterns to dynamic hypertext linking. The Fifth World Wide Web Conference (WWW5), Paris, France, May 6-10, pp. 1007-1014
- [23] **Agrawal, R.; Srikant, R.**, 1994, Fast algorithms for mining association rules. In Proc. 20th Int. Conf. Very Large Data Bases, VLDB, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, pp. 487- 499.
- [24] **Han, J.; Pei, J.; Mortazavi-Asl, B.; Chen, Q.; Dayal, U.; Hsu, M.-C.** 2000, Freespan: frequent pattern-projected sequential pattern mining. In Proceedings

of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM Press, pp. 355- 359.

[25] MySQL,

<http://www.mysql.org>

[26] Zemberek: Java Kullanım,

https://zemberek.dev.java.net/java_kullanim.html

[27] Zemberek Project Home Page,

<https://zemberek.dev.java.net>

[28] University Ontology 1.0 (draft),

<http://www.cs.umd.edu/projects/plus/SHOE/onts/univ1.0.html>

[29] Computer Science Department Ontology (draft),

<http://www.cs.umd.edu/projects/plus/SHOE/onts/cs1.1.html>

[30] **McCallum, A.; Nigam K.** 1998, A Comparison of Event Models for Naive Bayes Text Classification". In AAAI/ICML-98 Workshop on Learning for Text Categorization, Technical Report WS-98-05. AAAI Press. pp. 41-48.

[31] **Li, J.; Zaïane, O. R.**, 2004 August 30- September 3, Combining usage, content, and structure data to improve Web site recommendation, 5th International Conference on Electronic Commerce and Web Technologies (EC-Web 04), Springer Verlag LNCS 3182, Zaragoza, Spain, pp. 305-315.

[32] **Eirinaki, M.; Vazirgiannis, M.; Varlamis, I.** 2003, SEWeP: Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process. In Proceedings of the 9th SIGKDD Conference

[33] **Jones, S.K.; Willet, P.**, 1997, *Readings in Information Retrieval*, San Francisco: Morgan Kaufmann, ISBN 1-55860-454-4.

[34] **Karoui, L.; Aufaure, M.-A.; Bennacer, N.**, 2006. Web Intelligence 2006. IEEE/WIC/ACM International Conference on Volume , Issue , 18-22 Dec. 2006 Page(s):420 – 427, Digital Object Identifier 10.1109/WI.2006.55

- [35] **Hearst, M.**, 1992, Automatic acquisition of hyponyms from large corpora. In Proceedings of the 14th International Conference on Computational Linguistics (COLING'92), pages 538–545, Nantes, France
- [36] **Doan A., Madhavan , J.; Domingos, P.; Halevy, A.**, 2001, Learning to Map between Ontologies on the Semantic Web, University of Washington
- [37] **Prasenjit M., Natasha F. Noy, Jaiswals A.**, 2005, "OMEN: A Probabilistic Ontology Mapping Tool", International Semantic Web Conference 2005: 537--547.
- [38] **Eirinaki, M.; Vazirgiannis, M.; Varlamis, I.**, 2003, SEWeP: Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process. In Proceedings of the 9th SIGKDD Conference
- [39] **Oberle, D.; Berendt B.; Hotho, A.; Gonzalez, J.**, 2003, Conceptual User Tracking, to appear in Proceedings of the Atlantic Web Intelligence Conference (AWIC) Madrid, Spain (2003)
- [40] **Ziegler, C.Nicolas**, 2004, Semantic Web Recommender Systems. EDBT Workshops 2004: 78-89
- [41] **Chakrabarti, S.; Dom B.; Gibson D.; Kleinberg J.; Kumar R.; Raghavan P., Rajagopalan S., Tomkins A.**,1999. Mining the Link Structure of the World Wide Web, IEEE Computer (1999) Vol.32 No.6
- [42] SHOE Ontologies
<http://www.cs.umd.edu/projects/plus/SHOE/onts/>
- [43] Protege: Ontology Editor and Knowledge Acquisition System,
<http://protege.stanford.edu/>
- [44] **Göksedef, M.; Gündüz, Ş. Öğüdücü**, 2007. "A Consensus Recommender for Web Users", ADMA 2007: The Third International Conference on Advance Data Mining and Applications, Lecture Notes in Computer Science, Springer, pages 287-299.
- [45] **Agrawal,R.; Srikant, R.**, 1994. Fast Algorithms for Mining Generalized Association Rules. In Proceedings of the 20th International Conference on Very Large Databases (VLDB'94), Santiago

- [46] WebTrends Marketing Web Analytics and Web Statistics
<http://www.webtrends.com>
- [47] IBM – IBM SurfAid Analytics – United States
<http://surfaid.dfw.ibm.com/>
- [48] **Weiyang, L.; Alvarez, S.; Ruiz, C.**, 2002. Efficient Adaptive-Support Association Rule Mining for Recommender Systems. *Data Mining and Knowledge Discovery*, Springer Netherlands, ISSN: 1384-5810, Volume 6, Number 1 / January, 2002, pp. 83-105
- [49] **Wu, Z.; Palmer, M.**, 1994. Verb Semantics and Lexical Selection *Proc. 32nd Ann. Meetings of the Assoc. for Computational Linguistics*, pp. 133-138
- [50] Developer Resources for Java Technology
<http://java.sun.com/>

A. EKLER

Bölüm 4.2.1’de veri kümelerine ait oluşturulan ontolojilere ait kavramların ve bu kavramlar arasındaki hiyerarşik ilişkiler aşağıdaki tablolarda sunulmaktadır. Bunlara ek olarak; tablolarda, kavramlara yansıtılan doküman sayıları ve bu dokümanlara yapılan ziyaret sayıları da belirtilmektedir. Bu değerler anlamsal öneri modeli tarafından kullanılmaktadır.

Tablo A.2’de İTÜ veri kümesine ait ontoloji, Tablo A.3’de de CS veri kümesine ait ontoloji sunulmaktadır. Aşağıdaki Tablo A.1’de ise bu tablolarda sunulan kolonlara ait açıklamalar bulunmaktadır.

Tablo A.1: Ontoloji tablo açıklaması

Alan Değeri	Açıklama
Sınıf Hiyerarşisi	Sınıf Hiyerarşisinin Sayısal Gösterimi
SınıfId	Sınıfa ait anahtar numara
ÜstSınıfID	Bir üst sınıfa ait anahtar numara
#İng_Doküman	Sınıfa ait İngilizce doküman sayısı
#Tr_Doküman	Sınıfa ait Türkçe doküman sayısı
#Sınıflandırılmayan	Sınıfa ait dil sınıflandırılması yapılamayan doküman sayısı(Web dokümanı bulunamayan oturum URLleri)
#Toplam Sınıf Doküman	Sınıfa ait toplam doküman sayısı
Toplam(#AltSınıf)+ #Toplam Sınıf Doküman	Sınıfın alt sınıflarındaki dokümanlarıyla beraber toplam doküman sayısı
#Sayfa Ziyareti	Oturum verisinde bu sınıfa yapılan gezimlerin sayısı

A.1 ITU Veri Kümesine ait Anlamsal Web Yapısı

Tablo A.2: ITU Veri Kümesi için Anlamsal Web Yapısı

ITU Veri Kümesi için Anlamsal Web Yapısı								
Sınıf Hiyerarşisi	Sınıflıd	ÜstSınıflıD	(#İng_ Dokuman)	(#Tr_ Dokuman)	(#Sınıflandırılmayan)	(#Toplam Sınıf Dokuman)	[toplaml(#AltSınıf)+ #Toplam Sınıf Dokuman]	# sayfa ziyaret sayısı
1.Conference	1	-1	17	1	11	29	29	15
2.Organization	2	-1	0	0	10	10	639	32
2.1.EducationOrganization	3	2	6	1	42	49	481	1370
2.1.1.Department	4	3	2	0	6	8	8	131
2.1.2.Institute	5	3	1	0	7	8	8	75
2.1.3.Program	6	3	14	0	20	34	34	288
2.1.4.ResearchGroup	7	3	4	0	22	26	26	225

Sınıf Hiyerarşisi	Sınıflıd	ÜstSınıflıD	(#İng_ Dokuman)	(#Tr_ Dokuman)	(#Sınıflandırılmayan)	(#Toplam Sınıf Dokuman)	[toplaml(#AltSınıf)+ #Toplam Sınıf Dokuman]	# sayfa ziyaret sayısı
2.1.5.School	8	3	0	0	0	0	0	0
2.1.6.University	9	3	71	3	268	342	342	15381
2.1.7.Library	54	3	0	1	13	14	14	32
2.2.SocialOrganization	51	2	5	1	33	39	148	2650
2.2.1.Foundation	52	51	5	0	29	34	34	1256
2.2.2.CulturalActivityCenter	53	51	7	1	13	21	21	0
2.2.3.Services	55	51	3	0	27	30	30	1750
2.2.4.Club	56	51	6	0	18	24	24	351
3.Person	10	-1	0	0	1	1	270	11
3.1.Student	11	10	14	3	25	42	56	282

Sınıf Hiyerarşisi	SınıfID	ÜstSınıfID	(#İng_ Dokuman)	(#Tr_ Dokuman)	(#Sınıflandırılmayan)	(#Toplam Sınıf Dokuman)	[toplaml(#AltSınıf)+ #Toplam Sınıf Dokuman]	# sayfa ziyaret sayısı
3.1.1.GraduateStudent	12	11	3	0	6	9	9	0
3.1.2.UndergraduateStudent	13	11	0	0	5	5	5	0
3.2.Worker	14	10	5	0	22	27	213	2781
3.2.1.AdministrativeStaff	15	14	2	0	16	18	97	135
3.2.1.1.Chair-Professor	16	15	2	0	14	16	16	205
3.2.1.2.ClericalStaff	17	15	2	0	5	7	7	99
3.2.1.3.Dean_Professor	18	15	1	0	7	8	8	104
3.2.1.4.Director	19	15	0	0	0	0	0	0
3.2.1.5.SystemsStaff	20	15	2	5	41	48	48	77
3.2.2.Assistant	21	14	0	0	1	1	1	0

Sınıf Hiyerarşisi	Sınıflıd	ÜstSınıflıD	(#İng_ Dokuman)	(#Tr_ Dokuman)	(#Sınıflandırılmayan)	(#Toplam Sınıf Dokuman)	[toplaml(#AltSınıf)+ #Toplam Sınıf Dokuman]	# sayfa ziyaret sayısı
3.2.3.Faculty	22	14	17	0	64	81	88	716
3.2.3.1.Lecturer	23	22	0	0	2	2	2	0
3.2.3.2.PostDoc	24	22	0	0	1	1	1	0
3.2.3.3.Professor	25	22	0	0	3	3	4	0
3.2.3.3.1.AssistantProfessor	26	25	0	0	0	0	0	0
3.2.3.3.2.AssociateProfessor	27	25	1	0	0	1	1	0
3.2.3.3.3.FullProfessor	28	25	0	0	0	0	0	0
3.2.3.3.4.VisitingProfessor	29	25	0	0	0	0	0	0
4.Publication	30	-1	0	0	0	0	2	0
4.1.Article	31	30	0	0	0	0	0	0

Sınıf Hiyerarşisi	SınıfID	ÜstSınıfID	(#İng_ Dokuman)	(#Tr_ Dokuman)	(#Sınıflandırılmayan)	(#Toplam Sınıf Dokuman)	[toplaml(#AltSınıf)+ #Toplam Sınıf Dokuman]	# sayfa ziyaret sayısı
4.1.1.ConferencePaper	32	31	0	0	0	0	0	0
4.1.2.JournalArticle	33	31	0	0	0	0	0	0
4.2.Book	34	30	0	0	2	2	2	0
4.3.Manual	35	30	0	0	0	0	0	0
4.4.Periodical	36	30	0	0	0	0	0	0
4.4.1.Journal	37	36	0	0	0	0	0	0
4.4.2.Magazine	38	36	0	0	0	0	0	0
4.5.Proceedings	39	30	0	0	0	0	0	0
4.6.Specification	40	30	0	0	0	0	0	0
4.7.TechnicalReport	41	30	0	0	0	0	0	0

Sınıf Hiyerarşisi	SınıfID	ÜstSınıfID	(#İng_ Dokuman)	(#Tr_ Dokuman)	(#Sınıflandırılmayan)	(#Toplam Sınıf Dokuman)	[toplaml(#AltSınıf)+ #Toplam Sınıf Dokuman]	# sayfa ziyaret sayısı
4.8.Thesis	42	30	0	0	0	0	0	0
4.8.1.DoctoralThesis	43	42	0	0	0	0	0	0
4.8.2.MastersThesis	44	42	0	0	0	0	0	0
4.9.UnofficialPublication	45	30	0	0	0	0	0	0
5.Schedule	46	-1	0	0	0	0	0	0
6.Software	47	-1	0	0	0	0	0	0
7.Work	48	-1	0	0	0	0	18	0
7.1 Course	49	48	0	0	18	18	18	789
7.2 Research	50	48	0	0	0	0	0	0

A.2 CS Veri Kümesine ait Anlamsal Web Yapısı

Tablo A.3: CS Veri Kümesi için Anlamsal Web Yapısı

CS Veri Kümesi için Anlamsal Web Yapısı							
Sınıf Hiyerarşisi	Sınıflıd	ÜstSınıflıD	(#İng_Dokuman)	(#Tr_dokuman)	(#Toplam Sınıf Dokuman)	[Toplam(#AltSınıf)+ #Toplam Sınıf Dokuman]	#Sayfa Ziyaret
1.Event	1	-1	8	6	14	20	68
1.1 Conference	2	1	1	5	6	6	0
1.2 Presentation	3	1	0	0	0	0	0
2.Person	4	-1	0	0	0	455	
2.1 Student	5	4	11	7	18	18	324
2.1.1 Graduate Student	6	5	0	0	0	0	0
2.1.2 Undergraduate Student	7	5	0	0	0	0	0
2.2 Worker	8	4	13	4	17	437	1496

Sınıf Hiyerarşisi	SınıflD	ÜstSınıflD	(#Ing_Dokuman)	(#Tr_dokuman)	(#Toplam Sınıf Dokuman)	[Toplam(#AltSınıf)+ #Toplam Sınıf Dokuman]	#Sayfa Ziyaret
2.2.1 Administrative Staff	9	8	0	0	0	0	0
2.2.1.1 Director	10	9	0	0	0	0	0
2.2.1.2 Systems Staff	11	9	0	0	0	0	0
2.2.1.3 Clerical Staff	12	9	0	0	0	0	0
2.2.2 Assistant	13	8	112	63	175	175	1026
2.2.3 Faculty Member	14	8	0	0	0	245	0
2.2.3.1 Lecturer	15	14	20	24	44	44	862
2.2.3.2 PostDoc	16	14	70	40	110	110	1052
2.2.3.3 Professor	17	14	56	35	91	91	612
3.Publication	18	-1	8	1	9	496	57
3.1 Article	19	18	0	0	0	0	0

Sınıf Hiyerarşisi	SınıflD	ÜstSınıflD	(#Ing_Dokuman)	(#Tr_dokuman)	(#Toplam Sınıf Dokuman)	[Toplam(#AltSınıf)+ #Toplam Sınıf Dokuman]	#Sayfa Ziyaret
3.2 Book	20	18	0	0	0	0	0
3.3 Lecture Notes	21	18	363	76	439	439	1762
3.4 Thesis	22	18	0	0	0	0	0
3.5 Other Publication	23	18	25	23	48	48	778
4. Work	24	-1	0	0	0	1175	0
4.1 Course	25	24	0	0	0	992	0
4.1.1 Undergraduate Courses	26	25	15	1	16	834	69
4.1.1.1 Undergraduate Department Lessons	27	26	301	156	457	457	5489
4.1.1.2 Undergraduate Elective Lessons	28	26	111	60	171	171	272

Sınıf Hiyerarşisi	SınıflD	ÜstSınıflD	(#Ing_Dokuman)	(#Tr_dokuman)	(#Toplam Sınıf Dokuman)	[Toplam(#AltSınıf)+ #Toplam Sınıf Dokuman]	#Sayfa Ziyaret
4.1.1.3 Undergraduate Not-department Lessons	29	26	111	79	190	190	101
4.1.2 Graduate Courses	30	25	112	46	158	158	700
4.2 Research	31	24	13	8	21	21	36
4.3 Commissions	32	24	15	10	25	25	0
4.4 Abet	33	24	90	47	137	137	46
5. Information	34	-1	29	6	35	1589	51
5.1 Announcement	35	34	9	1	10	1216	30
5.1.1 Course announcement	36	35	0	0	0	1095	0
5.1.1.1 HW	37	36	288	46	334	334	1873
5.1.1.2 Exam	38	36	201	98	299	299	1918

Sınıf Hiyerarşisi	SınıflD	ÜstSınıflD	(#Ing_Dokuman)	(#Tr_dokuman)	(#Toplam Sınıf Dokuman)	[Toplam(#AltSınıf)+ #Toplam Sınıf Dokuman]	#Sayfa Ziyaret
5.1.1.3 Practice	39	36	348	114	462	462	2808
5.1.1.4 Other course announcement	40	36	0	0	0	0	0
5.1.2 Student announcement	41	35	53	38	91	91	355
5.1.3 General announcement	42	35	15	5	20	20	0
5.2 News	43	34	11	1	12	12	348
5.3 Private information	44	34	0	0	0	326	0
5.3.1 Introductory information	45	44	96	60	156	156	1143
5.3.2 Presentation information	46	45	135	35	170	170	12

ÖZGEÇMİŞ

Göksel Oral 1983’de İstanbul’da doğdu. 2001 yılında Ümraniye Anadolu Lisesi’nden mezun oldu. 2001 yılı ÖSS sonuçlarına göre; ilk 1000’e girerek İTÜ Bilgisayar Mühendisliğinde okumaya hak kazanmıştır. Üniversite öğrenimi boyunca İTÜ Geliştirme Vakfından Onur Bursu alarak 2005 senesinde mezun olmuştur. Lisans bitirme ödevi olarak “Yatırım Analizi ve Portföy Optimizasyonu” konusunu almış ve bu çalışmada hisselerle ait kazanç ve getiri istatistikleri kullanılarak hisselerin homojen risk gruplarına demetlenmesi sağlanmıştır. Farklı risk gruplarından hisseler seçilerek karı maksimuma çekerken, riski minimuma indiren yatırım sepetlerinin oluşturulmasına çalışılmıştır. Bu çalışmada ayrıca yatırımcı tarafından kullanılabilir analizlerini kullanabildiği bir Web arayüzü gerçekleştirilmiştir. 2005 yılı Eylül ayında Bilgisayar Mühendisliği Yüksek Lisans programına kaydolmuştur.

Yüksek Lisans Öğrenim hayatı sırasında sürdürdüğü iş yaşamında; 2005-2007 yılları arasında IBTECH Finansbank’ın veri madenciliği grubunda görev almış. 2007’den bugüne de Atos Origin Danışmanlık şirketinde ürün destek&geliştirme ekibinde görevini sürdürmektedir.