T.C.
KIRIKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
DOKTORA TEZİ

ALTERNATİF DÜŞÜK RANKLI MATRİS AYRIŞIMI İLE GİZLİ ANLAMSAL DİZİNLEME

Fahrettin HORASAN

HAZİRAN 2018

Bilgisayar Mühendisliği Anabilim Dalında Fahrettin HORASAN tarafından hazırlanan ALTERNATİF DÜŞÜK RANKLI MATRİS AYRIŞIMI İLE GİZLİ ANLAMSAL DİZİNLEME adlı Doktora Tezinin Anabilim Dalı standartlarına uygun olduğunu onaylarım.

Prof. Dr. Hasan ERBAY
Anabilim Dalı Başkanı

Bu tezi okuduğumu ve tezin **Doktora Tezi** olarak bütün gereklilikleri yerine getirdiğini onaylarım.

Prof. Dr. Hasan ERBAY
Danışmanı

Jüri Üyeleri

Başkan	: Prof. Dr. Fatih BAŞÇİFTÇİ	_____
Üye (Danışman)	: Prof. Dr. Hasan Erbay	_____
Üye	: Doç. Dr. Adem Alpaslan ALTUN	_____
Üye	: Dr. Öğr. Üyesi Cenker BİÇER	_____
Üye	: Dr. Öğr. Üyesi B. Gürsel Emiroğlu	_____

……/…../…….

Bu tez ile Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Doktora derecesini onaylamıştır.

Prof. Dr. Mustafa YİĞİTOĞLU
Fen Bilimleri Enstitüsü Müdürü

Aileme

ÖZET

ALTERNATİF DÜŞÜK RANKLI MATRİS AYRIŞIMI İLE
GİZLİ ANLAMSAL DİZİNLEME

HORASAN, Fahrettin
Kırıkkale Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı, Doktora tezi
Danışman: Prof. Dr. Hasan ERBAY
Haziran 2018, 83 sayfa

Kullanım alanı sürekli genişleyen bilgisayarlar tarafından dijital ortamda depolanan verilerin boyutları günden güne büyümektedir. Ancak bu veriler işlenmediği ya da analiz edilmediği sürece sadece bir arşivden ibarettir. Bu nedenle, istatistikçiler, ekonomistler, iş planlayıcıları, reklam analistleri ve iletişim mühendisleri gibi birçok sektör çalışanları bu depolanan verilerden anlamlı bilgiler elde etmek amacıyla sürekli araştırma ve geliştirme yapmaktadırlar. Araştırmacılar temel olarak büyük veri yığınlarından genel bir sonuca ulaşma, bilinen ya da bilinmeyen problemleri bulma, bu problemleri çözme, problem çözüm yöntemleri geliştirme, yapılabilecek bir değişikliğin etkisini tahmin etme, işlem ve deneylerini zamandan ve veri kaynaklarından bağımsız olarak yapabilmenin yollarını araştırmaktadırlar.

Bu çalışmada ise, devasa doküman yığını içerisinden istenilen dokümanlara ve/veya bilgilere doğru bir şekilde erişmeyi amaçlamayan bilgiye erişim sistemlerinden biri olan Gizli Anlamsal Dizinleme (GAD) yönteminde kullanılan Tekil Değer Ayrışımına (TDA) alternatif bir düşük ranklı matris ayrışımı önerilmektedir. GAD modelinde, doküman yığını içerisindeki her bir terim ve bu terimleri içeren dokümanlar lineer cebir yöntemleri ile sayısallaştırılarak bir vektör uzayında temsil edilmektedir. Vektör uzayının elde edilmesinde kullanılan genel yöntem ise TDA'dır. Ancak TDA ile gerçekleştirilen bu işlemin hesaplama ve hafıza açısından çok maliyetli olması araştırmacıları alternatif yöntemlere yönlendirmektedir Düşük

ranklı matris ayrışımı olarak önerilen Kesik ULV Ayrışımı ile (K-ULVA) vektör uzayının elde edilme sürecindeki maliyet TDA'ya göre daha düşüktür. Ayrıca, doküman yığınına eklenecek yeni dokümanların temsili için yapılan blok güncelleme sürecinin kolay ve maliyetinin az olması K-ULVA'nın bir diğer avantajıdır. K-ULVA ve TDA ile yapılan iki ayrı GAD sistemini karşılaştırılmak amacıyla bilgiye erişim çalışmalarında yaygın olarak kullanılan veri setleri tercih edilmiştir. Son olarak, bir bot yazılımı kullanarak Türkçe haber sayfalarından elde edilen haber metinleri ile Türkçe bir veri seti geliştirilmiş ve bu iki GAD sisteminin bu veri seti üzerindeki performansı da gözlemlenmiştir. Yapılan incelemeler sonucunda K-ULVA ve TDA tabanlı dizinleme modellerinin tüm veri setlerindeki başarılarının oldukça benzer olduğu görülmüştür. K-ULVA yönteminin blok güncelleme yöntemindeki kolaylığı ve maliyetinin az olması sebebiyle TDA yöntemine iyi alternatif matris ayrışımı olduğu sonucuna varılmıştır.

Anahtar kelimeler: Metin Madenciliği, Bilgiye Erişim, Düşük Ranklı Matris Ayrışımı, Kesik ULV Ayrışımı, Tekil Değer Ayrışımı, İçerik Analizi

ABSTRACT

LATENT SEMANTIC INDEXING WITH
ALTERNATE LOW RANK MATRIX APPROXIMATION

HORASAN, Fahrettin
Kırıkkale University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering, Ph. D. Thesis
Supervisor: Prof. Dr. Hasan ERBAY
June 2018, 83 pages

The size of the data stored in the digital environment is increasing day by day by the ever-expanding use of computers. However, this data is only an archive, unless it is processed or analyzed. For this reason, many sector employees, such as statisticians, economists, business planners, advertising analysts and communications engineers, are constantly researching and developing to obtain meaningful information from these stored data. Researchers are basically looking for ways to reach a general outcome from large data sets, finding known or unknown problems, solving these problems, developing problem-solving methods, estimating the effect of a possible change, and performing operations and experiments independently from data sources.

In this work, we propose an alternative low rank matrix decomposition for Singular Value Decomposition (SVD) which is used in the latent semantic indexing (LSI) method, which is one of the information retrieval systems that does not intend to access the desired documents and / or information from the gigantic collection of documents. In the LSI model, each term in the collection of documents and documents containing these terms are represented in a vector space by being digitized by linear algebra methods. The general method used to obtain the vector space is SVD. However, this process performed by the SVD is very costly in terms of calculation and memory, which diverts researchers to alternative methods. The

cost of obtaining the vector space with Truncated ULV Decomposition (T-ULVD), which is proposed as a low-rank matrix decomposition , is lower than TDA. Another advantage of K-ULVA is that the block updating process for the representation of new documents to be added to the collection of documents is easy and low cost. In order to compare two different LSI systems with T- ULVD and SVD, data sets commonly used in information retrieval studies have been preferred. Finally, a Turkish data set has been developed with news texts from Turkish news pages using a bot software and the performance of these two LSI systems on this data set are also observed. Based on the experiments, it is seen that the success of K-ULVA and TDA-based indexing models in all data sets are very similar. Because of the simplicity and low cost of the T-ULVD method in the block updating method, it is the result of a good alternative matrix decomposition to the SVD method.

Keywords: Text Mining, Information Retrieval, Low Rank Matrix Approximation, Truncated ULV Decomposition, Singular Value Decomposition, Content Analysis

TEŞEKKÜR

Tez sürecinde cesaretlendiren, yol gösteren ve her zaman desteğini hissettiğim tez danışmanım Sayın Prof. Dr. Hasan ERBAY'a, tez çalışmalarım esnasında, bilimsel konularda yardımlarını aldığım Sayın Dr. Öğr. Üyesi Cenker BİÇER'e ve Sayın Dr. Öğr. Üyesi Bülent Gürsel EMİROĞLU'na, büyük fedakârlıklarla bana destek olan arkadaşım Sayın Araştırma Görevlisi Fatih VARÇIN'a, doktora öğrenim süresince sonsuz bir anlayış ve sabır gösteren, çalışmalarımı aksatmamam için sayısız fedakârlıkta bulunan sevgili eşime teşekkürlerimi sunarım.

İÇİNDEKİLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

ÇİZELGE Sayfa

ŞEKİLLER DİZİNİ

ŞEKİL Sayfa

KISALTMALAR DİZİNİ

GAA	Gizli Anlamsal Analiz
GAD	Gizli Anlamsal Dizinleme
TDA	Tekil Değer Ayrışımı
K-ULVA	Kesik ULV Ayrışımı
TF	Terim Frekansı
TDF	Ters Doküman Frekansı
HTML	Hyper Text Markup Language
WWW	World Wide Web
IP	Internet Protokol
FTP	File Transfer Protocol
TBA	Temel Bileşen Analizi

1. GİRİŞ

Bilgilerin toplanmasına, işlenmesine, depolanmasına ve ağ teknolojisiyle erişimine olanak sağlayan bilişim teknolojilerindeki gelişmeler bu teknolojilerin kullanım alanını da yaygınlaştırmaktadır. Hayatımızın büyük bir bölümünde rast geldiğimiz bilişim teknolojileri hayatımızı kolaylaştırmanın yanı sıra alacağımız kararlarda da etkin rol oynamaktadır. Herhangi bir alandaki işlem sürecinde performanslı, tutarlı ve duyarlı sonuçlar almak amacıyla bu alanlara özgü uygulamalar kullanılmaktadır. Uygulamaların bu hizmetinin yanı sıra işlenmekte olan verilerin saklanması ve bu verilerin analizi ile anlamlı bilgiler elde edilmekte ve ilgili kararlar alınırken ön bilgi edinilebilmektedir [1,2].

Veri; uygulamalar ya da diğer adı ile programlarda kullanılan işlenmemiş, sayım, ölçüm, deney ya da araştırma yoluyla elde edilen ve tek başına anlamı olmayan gerçek ya da enformasyon kavramıdır. Bir probleme çözüm olabilmek için ilgili veriler üzerinde yapılan analizler sonucunda ortaya çıkan anlamlı ifade ya da ifadeler bütününe de bilgi denilmektedir [3]. Veri madenciliği ise, özelden genele ya da tümevarım mantığı ile büyük boyutlardaki veriler üzerinde işlem yapılarak bu verilerin birbirileri ile bilinmedik ve beklenmedik ilişkilerin keşfedilip irdelenmesi sonucu anlamlı ve yararlı örüntüleri çıkarma işlemidir [4,5]. Literatürde, veri madenciliğine eş değer başka adlandırmalar da bulunmaktadır. Bunlardan bazıları veri tabanlarında bilgi keşfi (knowledge discovery in databases), bilgi harmanlama (information harvesting), bilgi çıkarımı (knowledge extraction), veri ve örüntü analizidir (data / pattern analysis) [1,6].

Kullanım alanı sürekli artmakta olan bilgisayarlar tarafından elektronik ortamda otomatik ya da kontrol edilerek depolanan verilerin boyutları günden güne büyümektedir. Ancak bu veriler işlem ya da analiz yapılmadıklarında sadece arşiv özelliği taşımaktadırlar. İstatistikçiler, ekonomistler, iş planlayıcıları, reklam ajansları ve iletişim mühendisleri gibi birçok sektör çalışanları artık bu depolanan verilerden çıkabilecek anlamların peşine düşmüşler ve bu konularda araştırma geliştirme yapmaktadırlar [7]. Araştırmacılar büyük veri yığınlarından genel bir

sonuca ulaşma, bilinen ya da bilinmeyen problemleri bulma, bu problemleri çözme, problem çözüm yöntemleri geliştirme, yapılabilecek bir değişikliğin etkisini tahmin etme, işlem ve deneylerini zamandan ve veri kaynaklarından bağımsız olarak yapabilmenin yollarını araştırmaktadırlar [3,7].

Bilgilerin kayıt altına alınması kullanışlı olması sebebiyle genellikle metin formunda gerçekleştirilir. Bu sebeple bu alandaki çalışmalar geleneksel bilgiye erişim/bilgi keşfi çalışmalarına nazaran daha yaygın bir şekilde görülebilmektedir. Ancak metinler genellikle yapılandırılmamış verilerden oluştuğu için süreç içerisinde gerçekleştirilen işlemler daha karmaşıktır. Metin formundaki dokümanlardaki en temel öğe olan kelimelerin benzer kökten ya da gövdeden (aynı köke sahip olup yapım eki ya da çekim ekleriyle yeni kelimeler oluşturması) türemiş olması çözülmesi gereken en önemli sorunlardandır. Bunun yanında metinlerde kullanılan dilin bir durumu farklı kelimelerle ya da yöntemlerle ifade edilmesine imkân tanıması, bu tür erişim sistemlerinde en önemli sorun olarak dikkat çekmektedir. Diğer bir ifade ile farklı kelimelerle aynı durumun ifade edilebilmesi ve vurgu amacı gibi tek başına konu ile ilgisi olmayan kelimenin metin içinde anlamlı olması gibi durumlar söz konusudur. Bu durumlarda erişilen ya da analiz edilen dokümanlar içerdikleri kelimelerin eşlemesi ya da yanlış eşleşmeler sebebi ile yanlış sonuçlar vermektedir. Bu tür sorunlarla karşılaşmamak için, kelimelerin eşleşmelerinden ziyade bulundukları her bir dokümanlardaki temsil değerini işleme dâhil eden Gizli Anlamsal Analiz (Latent Semantic Analysis - GAA) yönteminin kullanılması önerilmektedir. GAA terim-terim, terim-doküman ve doküman-doküman arası gizli kalmış ilişkileri ortaya çıkaran bir istatistiksel/matematiksel bir yöntemdir. GAA vasıtası ile doküman yığını içerisindeki dokümanların sorgu cümlecikleri ya da dokümanın benzerliğine göre listelenmesi işlemine Gizli Anlamsal Dizinleme (Latent Semantic Indexing - GAD) denir. Terim-terim ve doküman-doküman arası ilişkilerden dolayı gizli anlamsal dizinleme işlemlerinde, erişilen dokümanların içinde istenen kelimelerin olmamasına rağmen istenilen anlamı taşıyan dokümanların olduğu görülebilir. Örneğin “eser” kelimesi ile “yapıt” kelimesinin anlamını dikkate alan böyle bir sistemde “eser” kelimesi ile ilgili dokümanlara erişilmek istenildiğinde içerisinde “eser” geçmeyen fakat “yapıt” kelimesinin geçtiği dokümanlarla

karşılaşılabilir. Böylece hayatımızda önemli bir katkısı olan farklı anlatma biçimlerinin bilgiye erişim sistemlerindeki olumsuz etkisinin önüne geçilmektedir.

Kelimeler ve kelimelerin bulundukları dokümanlar sırasıyla satır ve sütunlarında her bir satırdaki kelimenin ilgili sütundaki dokümandaki temsil ettiği değeri alarak oluşturduğu matrise terim-doküman matrisi denmektedir. Terim-doküman matrisine matris ayrışımları uygulanarak elde edilen vektörlerle vektör uzayı elde edilmektedir. GAA yönteminde genellikle Tekil değer ayrışımı (TDA) kullanılmaktadır. Ancak tekil değer ayrışımının maliyetinin büyük olması sebebi ile TDA'ya alternatif yöntemler önerilmektedir [8,9].

Son yıllarda gerçekleştirilen çalışmalarda, matris ayrışımları kullanılarak bilgi çıkarımı [10,11], metin madenciliği [11,12], doküman sınıflandırma [11,13], web madenciliği [11-13], sosyal medya madenciliği [11-14], imge, ses ve video işleme [14,15] gibi alanlarda çalışmalar yapılmıştır.

GAA metin madenciliği, görüntü işleme, veri madenciliği, sinyal işleme, ses analizi gibi birçok alanda kullanılmaktadır. Elvan lisansüstü tezinde destek vektör makineleriyle web sayfalarını sınıflandırmak için özellik çıkarımı amacıyla GAA kullanmışlardır [16]. Benzer bir çalışmada Shima K. ve arkadaşlarıysa sınıflama işleminde daha verimli sonuç almak için GAA indeksleme işleminden önce özellik sıralama metodu uygulamışlar sınıflandırma metodu için ise destek vektör makinesi tekniğini kullanmışladır [17]. Uysal ve Gunal yaptıkları çalışmada metin sınıflandırırken dokümanların daha iyi temsil edilmesi için genetik algoritmayla güçlendirilmiş GAD'dan faydalanmıştır. Bu çalışmada terim doküman matrisindeki dokümanların daha iyi temsil edilmesi, en büyük tekil değerlerin işleme alındığı standart GAD yaklaşımlarının aksine uygun tekil değerlerin bulunmasıyla gerçekleştirilmiştir [18].

Güran ise yaptığı çalışmada GAA temelli ve çıkarıma dayalı bir metin özetleme sistemi gerçekleştirmiştir. Ayrıca, önermiş olduğu ağırlık değerlendirmesinin başarısını görebilmek için bu ön işlem aşamasını dört farklı GAA yönteminde denemiş ve ağırlık değerlendirmesinin tüm yöntemlerde daha başarılı sonuçlar

verdiğini gözlemiştir [19]. Steinberger ve Murray'ın yaptıkları çalışmalarda metin özeti için terim doküman matrisi TDA ile çarpanlarına ayrılmış ve daha sonra da bu çarpanlardan terim ve doküman verilerine dair bağımsız vektörler elde edilmiştir. Böylece metin içerisindeki daha çok ilişki içerisindeki olan terim ve dokümanlar dikkate alınarak metni temsil eden dokümanlardan oluşan yeni bir metin elde edilmiştir. [20,21]. Lee ve arkadaşları ise yeni bir cümle seçim kriteri önerdikleri çalışmada negatif olmayan matris ayrışımını metin özetleme sisteminde kullanarak GAA ile yapılan metin özetlemeleriyle kıyaslayarak geliştirdikleri algoritmanın daha başarılı olduğunu gözlemlemişlerdir [22]. Özsoy geliştirdiği cümle seçim metotları farklı olan iki adet GAA temelli metin özeti algoritmasını, GAA'nın sadece o an üzerinde çalıştığı metin dışında başka bir metin gruplarına ihtiyaç duymadan ya da herhangi bir ön öğrenme edinmeksizin işlem yaptığına dikkat çekerek diğer metin özetleme algoritmalarıyla karşılaştırmıştır. GAA tabanlı metin özetleme algoritmasının küçük boyuttaki metinlerin özetindeki başarısı diğer metin özetleme algoritmalarına göre daha düşük olduğu gözlemlenmiştir. Büyük boyutlardaki metinlerde ise kelime ve doküman sayısının fazla ve çeşitli olmasından dolayı anlamsal yapı yeterli olmuş ve başarısı diğerleriyle benzerlik göstermiştir [23].

Kumar yapmış olduğu çalışmada Medline veri tabanından aldığı gerçek veri seti üzerinde bilgi keşfi amacıyla oluşturduğu vektör uzay modelinde Örgün Kavram analizi ve gizli anlam dizinleme tekniklerini irdelemiş ve benzer sonuçlar almıştır [10]. GAA yönteminin kıyaslandığı diğer bir çalışmada ise GAA, TF*IDF ve Çoklu-Sözcük yöntemlerinin metin sınıflandırma ve bilgi keşfi amacıyla test edilebilir olduğu görülmüş ve GAA'nın yeniden ölçeklendirilmesinin Çince ve İngilizce doküman kümesinde en verimsiz olduğu açıklanmaktadır [12]. Bir diğer çalışmada ise kategori ayrımının kolay olmadığı tereddütlü web sayfalarının sınıflandırılmasında yoğunluk temelli kaba küme modelini kullanarak GAA yöntemiyle karşılaştırmış ve kaba küme modelini yeni bir uygulama alanında denemiştir [13].

Büyük boyutlu yani uzun metin verilerinin yerine az boyutlu metinlerin işlenmesi üzerine durulan çalışmada ise kısa metinlerin kullanım alanları ve etkinliğinin önemi

üzerine durularak bu metinler için tekil değer ayrışımının kullanıldığı GAA ve LDA gibi yaklaşımlarla birlikte mevcut metin sınıflandırma yöntemleri irdelenmiştir [11].

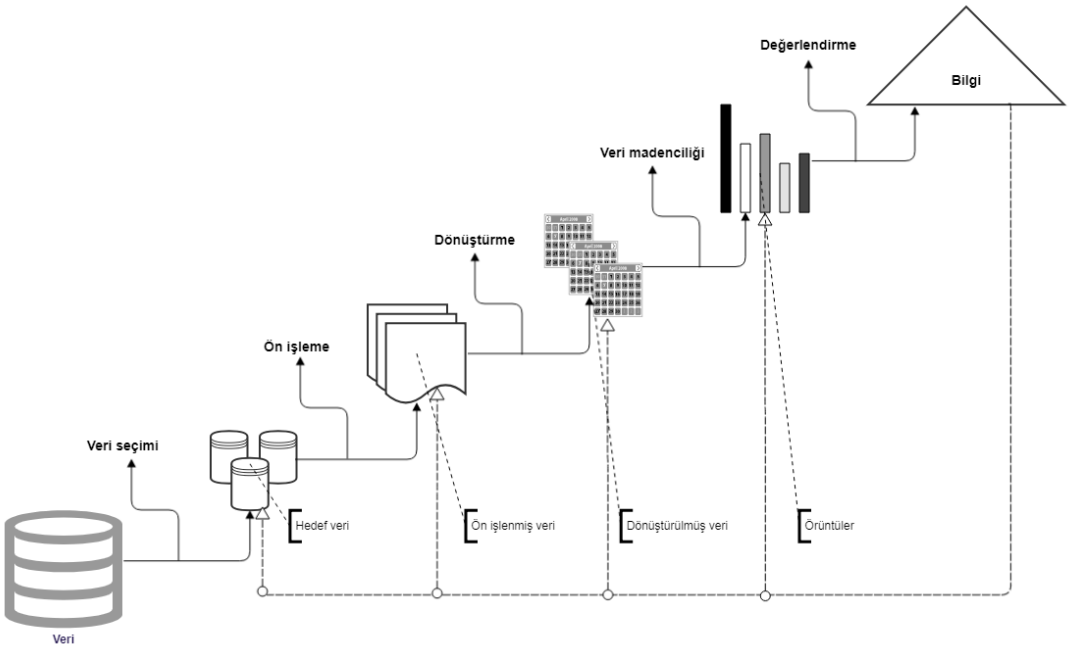
O'brien TDA Güncelleme ismini vermiş olduğu çalışmasında geliştirdiği algoritma ile terim doküman matrisine gelecek yeni doküman ve/veya terim blokları için yapılabilecek yeniden hesaplama işleminin aksine mevcut TDA ayrışım bilgileri güncellenerek yeni TDA matrisleri elde edilmiştir. Önermiş olduğu bu algoritmanın başarısını zaman karmaşıklığı, hafıza kullanımı ve çıkarım performansı açısından diğer benzer algoritmalarla kıyaslamıştır. Yapmış olduğu güncelleme işleminin TDA'nın yeniden hesaplanması işlemine göre daha az maliyetli ve performans çıkarımı göze alındığında iyi bir alternatif olduğu, folding-in yöntemine göre ise performans çıkarımı daha başarılı olmasına rağmen hafıza kullanımının daha çok olduğu gözlemlenmiştir [24]. Varçın yapmış olduğu çalışmada GAA'da matris ayrışımı için kullanılan TDA yerine, TDA'nın yeniden hesaplamasının ve güncelleme maliyetinin büyük olmasına dikkat çekerek kesik ULV algoritmasının daha az maliyetli ve benzer sonuçlar çıkaran bir algoritma olmasını öne sürerek iyi bir alternatif olabileceğini belirtmiştir [25].

Yapılan çalışmalar incelendiğinde, GAA ile yapılan çalışmalarında genellikle TDA'nın kullanıldığı görülmektedir. Ancak TDA'nın hesaplama karmaşıklığı ve yeni veriler geldiğinde vektör uzayının yeniden güncellenmesinin zorluğu nedeni ile TDA yerine alternatif matris yaklaşımları önerilmektedir. Ayrıca, boyut indirgeme tabanlı bir yaklaşım olan GAA'da dokümanlar arası gizli anlamsal yapıyı bulmada terim-doküman matrisinin düşük ranklı yaklaşımı kullanılmaktadır. Bu tez çalışmasında ise çok büyük doküman yığını içerisinden aranan dokümanların ya da benzer dokümanların Kesik ULV (Truncated ULV) ayrışımının kullanıldığı GAA ile dizinlenmesi incelenmiştir. Bu amaçla bilgiye erişim çalışmalarında yaygın olarak kullanılan Amerikan Dokümantasyon Enstitüsü Raporları (ADI), Time dergisinde yayınlanan makale koleksiyonu (TIME) ve Medline makalelerinden oluşan koleksiyon (MED) gibi veri setlerinin yanında arama motorlarında kullanılan bot benzeri bir yazılım geliştirilerek 5 farklı Türkçe haber sitesindeki haber sayfaları veri seti olarak kullanılmıştır. Geliştirilen bu yazılım ile her bir sayfanın ön işlem süreci bir defaya mahsus yapılmaktadır. Geliştirilen yazılım vasıtası ile taranmakta olan

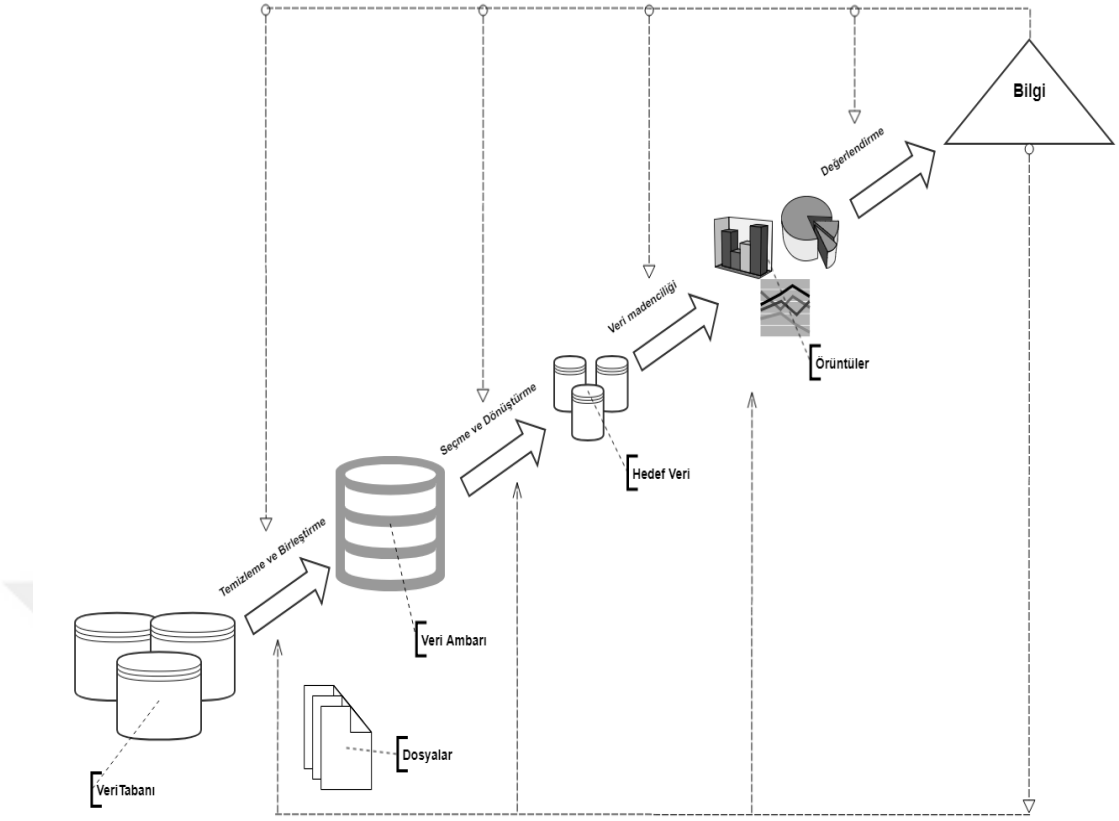
web sayfasındaki kelimeler ve bu kelimelerin bulunduğu dokümanlardaki sıklığı veri tabanına kaydedilmektedir. Her bir web sayfasındaki Hyper text markup language (HTML) kodları içerisindeki yapılandırılmamış metinler web madenciliğindeki ön işlem sürecinden geçirilerek elde edilmiştir. Daha sonra her biri yapılandırılmamış veri olan haber metinleri, metin madenciliği yöntemindeki ön işlem süreçlerinden geçirilmiştir. Geliştirilen yazılım ile web sayfalarındaki veri setlerine dair kelime ve sıklık bilgilerinin veri tabanına kaydedilmesinden sonra haber metinlerinde yer alan her bir kelimenin terim, her bir haber metninin doküman olarak isimlendirildiği terim-doküman matrisi elde edilmektedir. Elde edilen terim doküman matrisine uygulanan GAA yöntemleri uygulanmıştır. Çalışmada performanslarını incelemek amacıyla hem TDA hem de Kesik ULV ayrışımının uygulandığı iki farklı işlem gerçekleştirilmiştir.

2. BİLGİ KEŞFİ SÜRECİ

Veri madenciliği her ne kadar kapsamı geniş bir konu olsa da bilginin keşfi sürecinde bir aşama olarak yer almaktadır. Bilginin keşfinde veri madenciliği işlemi yapılmadan önce, verilerin seçimi, ön işlem, indirgeme adımları gerçekleştirilir. Veri madenciliği aşamasında geniş veri bütünlüğünden ilişkili enformasyonlar elde edilir. Sonrasında ise bu enformasyonları yorumlama ve doğrulama işlemleri ile bilgiye ulaşılmaktadır [5]. Buna göre verileri ayrıştırma, düzenleme, bir sonraki aşamaya hazır hale getirme ve yorumlama gibi işlemler bilgi keşfi sürecinin bir aşaması olarak yer almaktadır. Şekil 2.1 ve Şekil 2.2.'de iki farklı yaklaşıma göre bilgi keşfi süreci aşamalarıyla birlikte verilmektedir.



Şekil 2.1. Fayyad ve arkadaşlarına göre Bilgi Keşfi Sürecinde Veri Madenciliğinin Yeri [5]



Şekil 2.2. Han ve arkadaşlarına göre bilgi keşfi sürecinde veri madenciliğinin yeri [3]

2.1. Ön işleme Süreçleri

Veri madenciliğinde doğru sonuç almak için işlem yapılacak verilerin kaliteli olması en önemli kriterlerden biridir. Bu nedenle seçilen verilerdeki gürültülü, eksik, tutarsız ve hatalı verileri küme içerisinden çıkarmak ya da bu verileri düzenlemek gerekmektedir [26]. Ancak bu aşama bilgi çıkarım sürecinde en çok zaman ve kaynak gerektiren aşama olmaktadır [26,27]. Veri ön işleme sürecinde birden çok teknik kullanılmaktadır [3]. Bu teknikler;

2.1.1. Veri Temizleme

Veriler oluşturulurken ya da seçim işlemlerinden kaynaklanan verilerdeki eksikliklerin düzeltilmesi, çalışma verimliliğini olumsuz etkileyen gürültü adı verdiğimiz verilerin temizlenmesi ve verilerdeki tutarsız olanlarının tespit edilip çıkarılması gibi işlemlerin uygulandığı tekniktir [3,28].

2.1.2. Veri Birleştirme

Veri madenciliği çalışmalarında genel olarak farklı veri tabanlarındaki ya da kaynaktaki verileri bir arada tutmak için veri ambarı oluşturulur. Böylece farklı kaynaklarda bulunan veriler önceden bilinen ilişkilileri referans alınarak bir araya getirilir [3,26].

2.2. Veri Seçme

Büyük veri yığını üzerinde yapılacak analizler için anlamlı sonuç alabilecek değişkenlerin belirlenmesi, seçilmesi ve gereksiz özelliklerden arındırma aşamasıdır. Seçilecek verilerin sayısı da çalışmalın niteliğine ve üretilecek sonucun hassaslığına göre belirlenmelidir [29].

2.3. Veri Dönüştürme

Seçilen verilerin doğrudan veri madenciliği işlemine aktarılması kaynak, zaman ve işlem olarak maliyetli olabildiği gibi problemin çözülmesini de engelleyebilmektedir. Bu nedenle seçilen verilerin içeriğini değiştirmeden biçiminin ya da ifade şeklinin problemin çözümünde kullanılacağı forma dönüştürülmesidir [3].

2.4. Veri İndirgeme

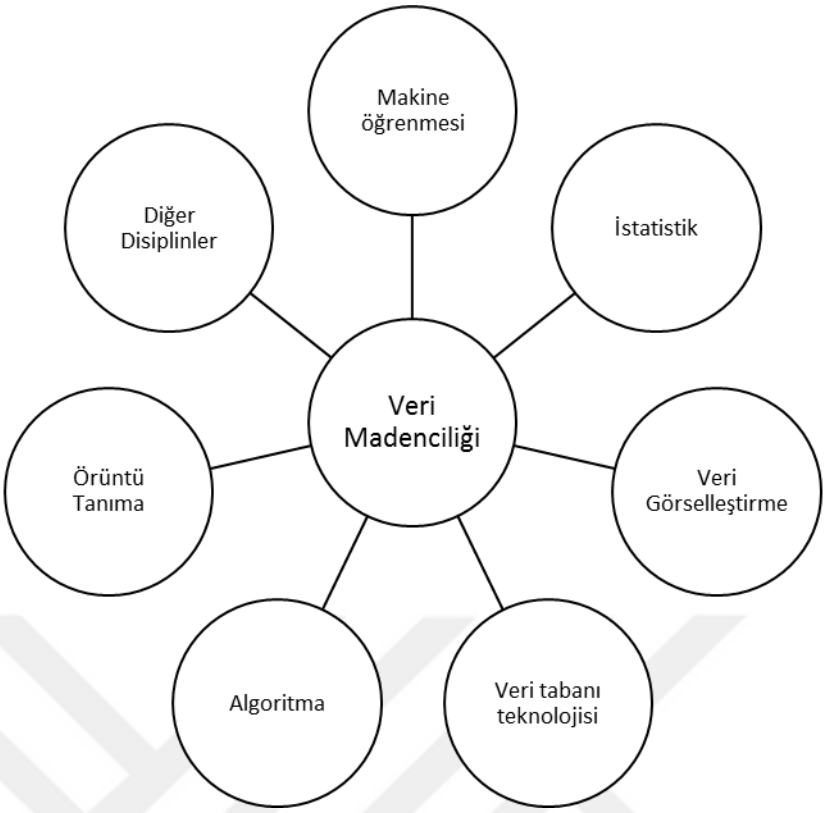
Veriler işleme alınırken bütün özellikleri dikkate alındığında işlem süresi uzun olmaktadır. Bu nedenle, verilerin çıkarıldığında sonucu değiştirmeyen bazı özelliklerinin dâhil edilmediği ve genel olarak tüm verinin yerini tutabilen temsilinin oluşturulmasıdır. İndirgeme işlemlerinde genel olarak veri birleştirme ve küp oluşturma, boyut indirgeme, veri sıkıştırma, sayıca azaltma, ayrıştırma ve hiyerarşi oluşturma gibi yöntemler kullanılmaktadır [3,29].

2.5. Veri Madenciliği

Görüldüğü gibi bilgi keşfi sürecinde bu aşamaya kadar veriler üzerinde anlamsal ve ilişkisel olarak herhangi bir işlem yapılmamaktadır. Veriler temizlenmiş, gerektiği yerde birleştirilmiş ve indirgenmiş olarak işlenmeye hazır hale getirilmiştir. Veri madenciliği ile de bu veriler işlemlere tabi tutularak ilişkisel ve anlamsal değer taşıyan sonuçlar üretilmektedir. Bu işlemlerin yapılması için de veri madenciliği modelleme yöntemleri kullanılmaktadır. Bu yöntemler genel olarak üç ana gruba ayrılmaktadırlar [5,30].

- Sınıflama ve Regresyon
- Kümeleme
- Birliktelik Kuralları ve Ardışık Zamanlı Örüntüler

Aynı zamanda veri madenciliği bir disiplinler arası çalışma alanıdır. Şekil 2.3'te de verildiği üzere veri madenciliğinin ilişkili olduğu ya da olabildiği diğer disiplinler ise makine öğrenmesi, istatistik, yapay zekâ, örüntü tanıma, sinir ağları, bilgi tabanlı sistemler, yüksek performanslı hesaplamalar, veri tabanı yönetim sistemi teknolojileri ve veri görselleştirmedir [3,5].



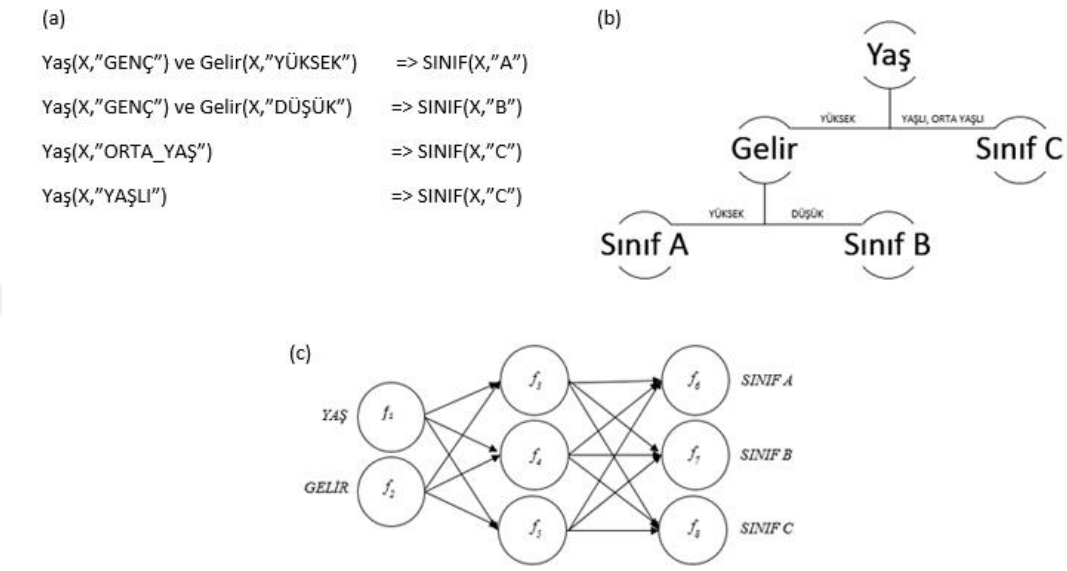
Şekil 2.3. Veri Madenciliğinin İlişkili olduğu Disiplinler [3]

2.5.1. Sınıflama ve Regresyon

Sınıfı belirli olmayan yeni bir nesnenin daha önceden özellikleri belirlenmiş sınıflara istatistik ya da makine öğrenmesi yöntemleri ile atamasının yapılması ya da daha sonra olabilecek eğilimleri tahmin etme işlemleridir. Özelliklere bakarak kategorilere ayırma işlemine sınıflama, süreklilik gösteren değerlerin tahmininde yapılan işleme ise regresyon analizi denmektedir. Sınıfların özellikleri ise sınıflama işlemine geçmeden önce yapılan eğitim sürecinde belirlenmektedir. Bu da sınıflandırma yöntemlerinin danışmanlı öğrenme yöntemleri olduğunu göstermektedir [3,30].

Sınıflama ve regresyon analizi modellemesine göre çeşitli yöntemler geliştirilmiştir. Bunlardan istatistiksel yöntemler içerisinde lineer regresyon analizi, lojistik regresyon analizi, diskriminanz analizi ve bayes sınıflandırma yöntemleri, makine öğrenimi yöntemlerinden ise yapay sinir ağları, destek vektör makineleri, karar

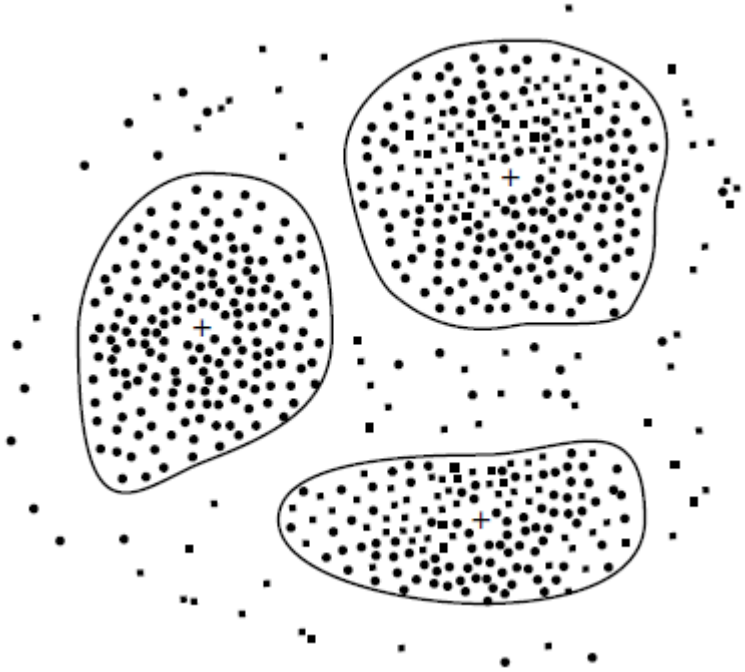
ağaçları, en yakın komşu algoritması gibi yöntemler yer almaktadır. Ayrıca genetik algoritma ve Fuzzy gibi sınıflandırma yöntemleri de kullanılmakta ve yeni modeller geliştirilmektedir [3,30]. Şekil 2.4.'de sınıflandırma modellerine örnekler görülmektedir.



Şekil 2.4. Sınıflandırma modeli biçimleri (a) Kural tabanlı (b) Karar Ağacı, (c)Sinir ağı örnekleri [13]

2.5.2. Kümeleme

Kümeleme, nesnelerin kendilerini temsil eden sayısal değerlerini göze alarak bu değerlerin birbirilerine olan uzaklıklarına ve yakınlıklarına göre Şekil 2.5'de olduğu gibi gruplara ayrılması işlemidir. Birbirine benzeyen yani yakın olan nesneler bir gruba ve bu gruba uzak olan diğer nesneler de farklı gruplarda yer almaktadırlar [3,30].



Şekil 2.5. Kümeleme örneği

Kümeleme işlemi, işlem öncesinde bir kısım verinin tanıtılması ya da eğitilmesi süreci olmaması ve kümelenecek verilerin tamamının özellikleri dikkate alınarak yapıldığı için danışmansız öğrenme grubuna girmektedir. Bu nedenle kümeleme işlemine danışmansız sınıflama da denilebilir [4]. Kümeleme işleminde bir diğer amaç ise büyük veri içerisindeki benzer özellik taşıyanlarının bir arada bulunduğu kümeleri bularak analiz için işlem yapılacak veri aralığını daraltmak, bu kümelere özgü özellikleri belirlemek ve farklı özellikleri bulundurup bu küme içerisinde yer alan istisnai verileri belirlemektir [31].

2.5.3. Birliktelik Kuralları ve Ardışık Zamanlı Örüntüler

Birliktelik kuralları büyük veri içerisinde birlikte hareket eden ya da ardışık zamanlarda gerçekleşen verileri tespit eden analiz yöntemidir. Analiz sonunda birlikte hareket eden bu verilerin paralel ya da ortak en az bir noktada birleştiğini dikkate alarak sonraki işlemlerde referans edinilir. Örneğin bir marketin alışveriş

faturasına bakarak kişilerin genel olarak birlikte aldıkları ürünleri görebilir ve bu kişilere uygun olarak markette satılan ürünlerin yerleştirmesi yapılabilir ya da bir web sayfasında kişilerin ziyaret ettikleri önceki sayfalar dikkate alınarak bu kişiye ilgilenebileceği ürün ya da sayfa önerilebilir.

2.6. Değerlendirme ve Yorumlama

Bilginin keşfi sürecinde veri madenciliği adımından sonra elde edilen anlamlı örüntülerin problemin çözümünde yeterli olup olmadığı, doğru ve tutarlı sonuçlar verip vermediği, farklı modellerde ve tekniklerde uygulanan işlem sonuçlarının kıyaslanarak probleme uygun model ve tekniğin belirlenmesidir. Yapılan işlem sonucunda elde edilen çıktıların kullanıcıya uygun bilgi çıkarımı yapacak şekilde sunumu gerçekleştirilir [3,30].

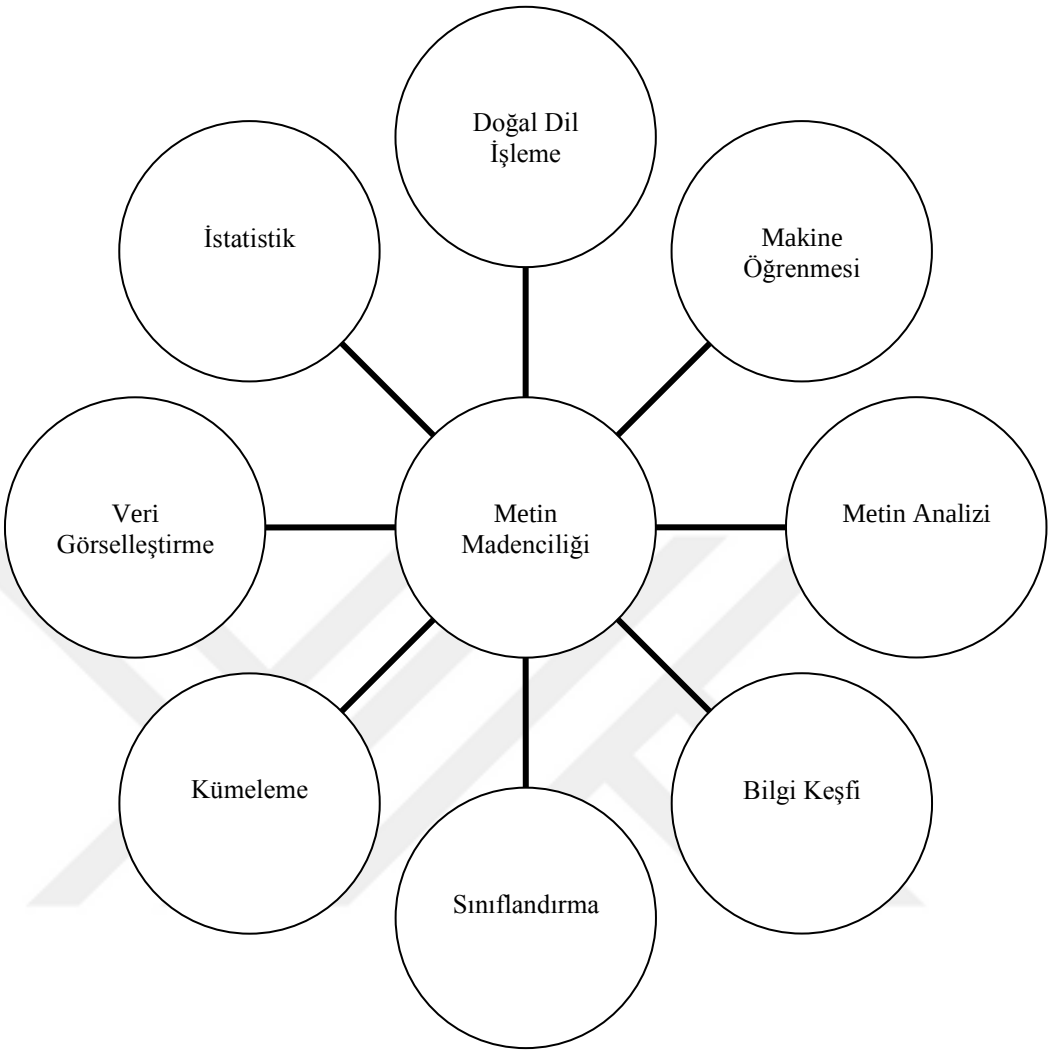
3. METİN MADENCİLİĞİ

Metin madenciliğinde veri kaynağı olarak yapılandırılmamış veri olarak nitelendirilen metin dokümanları işlenmektedir. Yapılandırılmamış veri ise bilgisayar tarafından anlamlandırılmayan, direk işleme tabi olabilecek şekilde veri yapısına sahip olmayan metin ya da sayısal ifadelerin bir arada sunulduğu organize edilmemiş veriler bütünüdür. Yapılandırılmış veri ise bilgisayar tarafından tanınan, işleme direk tabi olabilen, veri yapısı belirlenmiş kategorik ya da sayısal değerler içeren veriler olarak adlandırılmaktadır [32].

Metin madenciliği, veri kaynağı olarak yapılandırılmamış veri olan metinsel dokümanları işleyerek yeni bilgiler keşfedilmesini sağlayan veri madenciliği alanıdır. Örneğin; metinlerin benzerliği, sınıflandırılması, özetlenmesi, temsilci kelimelerinin oluşturulması, metinlerden duygu analizi, metinlerden yazan kişinin tespiti, metin içeriğine bağlı öneri sistemleri, soru-cevap sistemleri gibi birçok çalışma alanı mevcut ve gelişmeleri devam etmektedir [32,33].

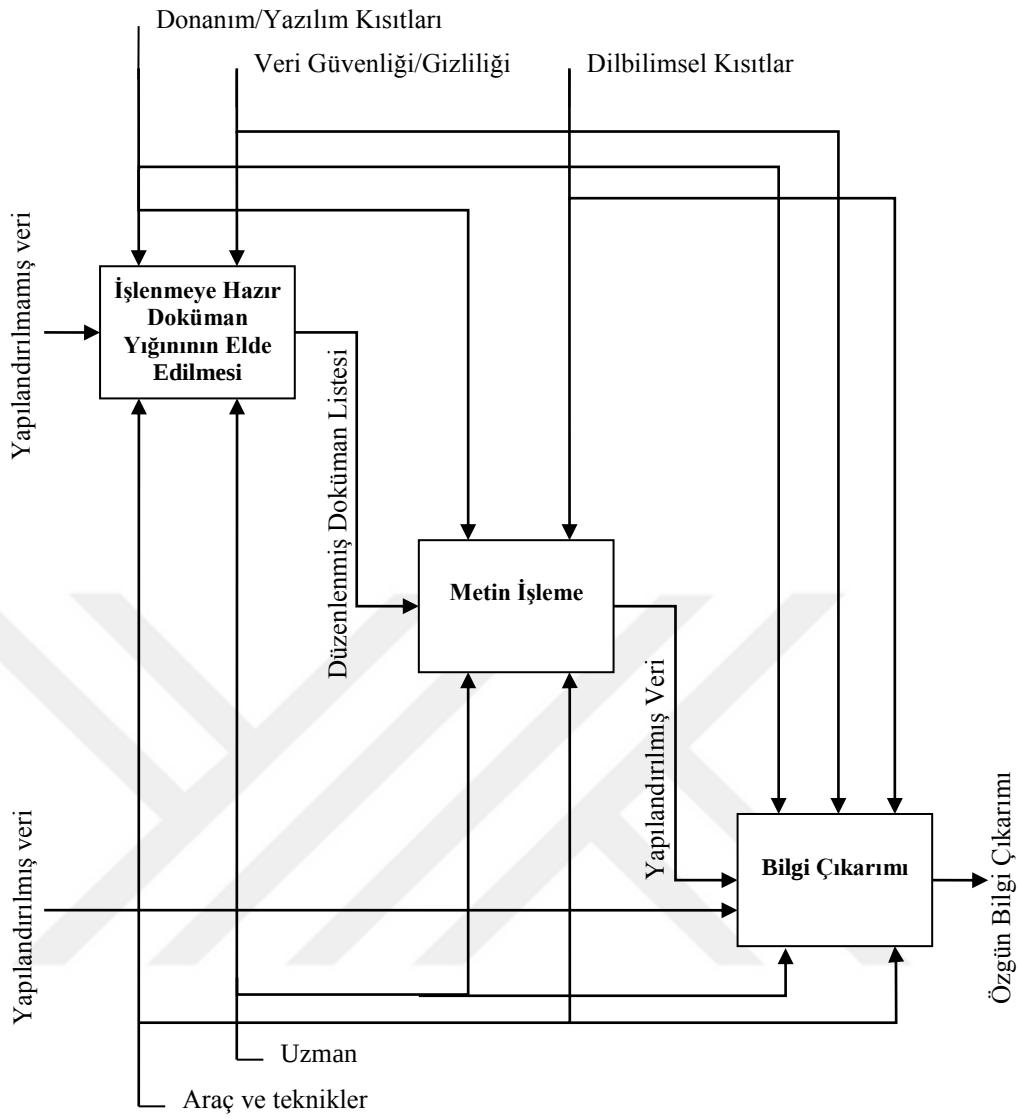
Metin madenciliği metin analizi teknolojisine dayanarak ortaya çıkan işlevler bütünüdür. Çok kapsamlı olmayan bir çalışma da bile otomatik olarak metinlerin önişleme sürecinden geçirilmesi, işlenmesi ve kullanıma hazır hale getirilmesi önemli bir çaba gerektirmektedir [34]. Metin madenciliği veri madenciliğinin bir alt kolu gibi görünse de her ikisini de birbirinden ayıran önemli farklar vardır. Veri madenciliği sayısal ya da kategorik olarak yapılandırılmış verileri inceleyip bilinmeyen ilişkileri çıkarmayı amaçlarken metin madenciliği yapılandırılmamış metin türündeki verileri inceleyerek farkında olmadığımız bilgilere ulaşmayı amaçlamaktadır [32,35,36].

Metin madenciliği makine öğrenmesi, istatistik, bilgi keşfi, doğal dil işleme, metin analizi, sınıflandırma, kümeleme, veri görselleştirme gibi birçok alanla birlikte anılmaktadır (Şekil 3.1.). Ayrıca veri madenciliği yöntemlerinde geliştirilen yeni algoritmaların da bu alanda kullanılması mümkün olabilmektedir[37].



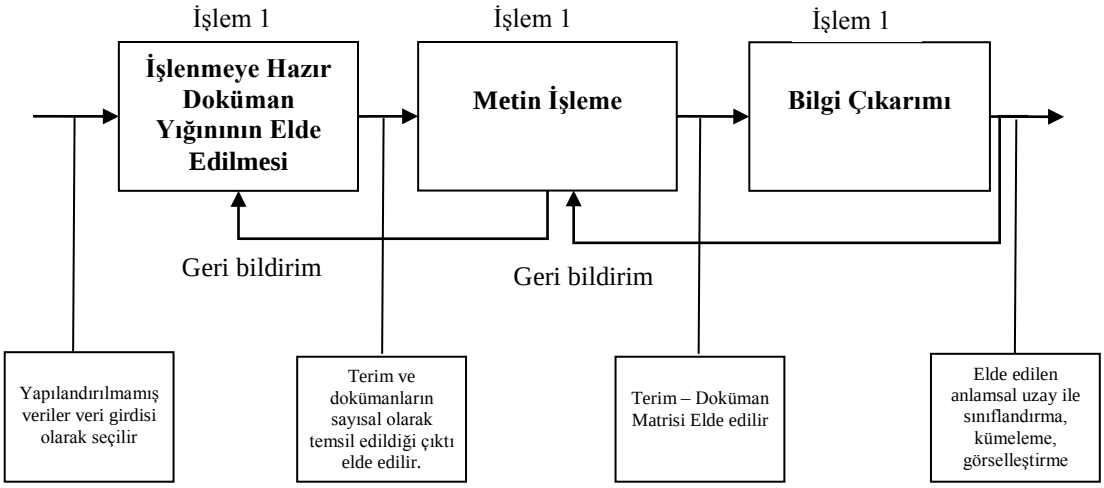
Şekil 3.1 Metin madenciliğin diğer disiplinlerle ilişkisi

Metin madenciliği süreci, işlenmesi gereken verilerin toplandığı ve işleme hazır hale getirildiği aşama, verilerin işlenerek yapılandırılmış verilerin elde edildiği aşama ve bunların ardından gelen bilgi çıkarım aşaması olarak üç aşamada incelenir. Şekil 3.2'de gösterildiği gibi yapılandırılmamış veriden bilginin keşfine doğru uzanan bu sürece tercih edilen yazılım ile ilgili kısıtlar, donanımsal kısıtlar, dilbilimsel kısıtlar da dâhil olmaktadır. Ayrıca bu sürecin öncesi ve sonrasındaki tüm aşamada kişisel verilerin gizliliği ve güvenliğinin sağlanmasına dikkat edilmelidir [38].



Şekil 3.2. Metin Madenciliği ve Paydaşları [38]

Metin madenciliği sürecinde (Şekil 3.3.) ilk olarak çalışılacak doküman yığını ya da korpus olarak adlandırılan veri kümesinin oluşturulması gerekmektedir. Doküman yığınları metin dosyalarının bir arada tutulduğu bir dizin ya da metin türünde verilerin tutulduğu veri tabanı olarak ele alınabilir. Ancak sürekli eklenerek artan web içerikleri dikkate alınırsa doküman yığını olarak verilerin kaynağı olan web sayfaları doküman yığını olarak dikkate alınmaktadır.



Şekil 3.3. Metin madenciliği süreci [39]

Metinsel dokümanlar bilişim sistemleri ile anlamsal olarak işlenebilmesi ve sonuç olarak çıkarımlarda bulunması, metinler üzerinde anahtar kelimelerin sorgulanması ile ya da metne ait kelimelerin bir araya gelerek oluşturduğu anlamsal yapıya dikkat ederek mümkün olmaktadır. Metinde yer alan kelimelerin tamamının metin içindeki öneminin yanı sıra diğer dokümanlarla olan ilişkilerine de dikkat edilmektedir. Metin içindeki her bir terim dilin yapısına göre standart bir şekilde temsil edilmelidir. Bu amaçla dilin özellikleri dikkate alınarak elde edilen doküman yığınındaki her bir terim kök ya da gövdelerine dikkat edilerek ön işleme sürecinden geçirilerek işlenmelidir. Ayrıca metin içerisinde çok geçmekle birlikte tek başına anlamı olmayan kelime grubu olarak tanımlanan durak kelimelerinin işleme dâhil olması engellenmelidir. Doküman yığınları içerisindeki bütün kelimeler ve bu kelimelerin bulunduğu her bir dokümanın temsil edildiği terim doküman matrisi kelimelerin dokümanlardaki ağırlığını dikkate alarak elde edilir. Çizelge 3.1'de örnek bir terim doküman matrisini görebilirsiniz. Metin madenciliğinde genellikle terim sayısı doküman sayısından büyük olduğu görülmektedir [37-39]. Ancak işlenmekte olan doküman sayısının katlanarak arttığı durumlarda terim sayısı doküman sayısından az olmaktadır. Bu duruma arama motorlarındaki terim ve doküman sayıları örnek verilebilir.

Çizelge 3.1. Örnek bir Terim Doküman Matrisi

Dokümanlar / Terimler	Doküman 1	Doküman 1	Doküman 3	…	Doküman n
Terim 1	1,2	0	0,78	…	0
Terim 2	0,05	0	0		0,1
Terim 3	0	1,1	0	…	0
⋮	⋮	⋮	⋮	⋮	⋮
Terim m	0	0	2,1	…	0

Metin madenciliğinde istatistiksel yöntemlerle metin içerisinden anahtar kelimelerin belirlenmesi ya da sadece terim doküman matrisindeki frekans sayıları tek başına yeterli değildir. Terim doküman matrisi vasıtası ile doküman yığını için bir anlamsal uzay elde edilerek metin içerisindeki anlamı etkilemeyen ya da anlamı bozan bileşenlerin göz ardı edildiği analiz işlemleri yerine getirilir. Bu anlamsal uzay sayesinde metin içerisindeki anlamsal kalıplar dikkate alınarak metne ait ilginç ya da belirli amaçlar için kullanılacak yararlı bilgiyi temsil eden verileri sunulur [34,39,41].

4. WEB MADENCİLİĞİ

Dünya üzerinde erişilebilen en büyük veri yığınlarından biri olan ve internet üzerinde dağıtık ve etkileşimli erişimi kolaylaştırmak için ilgili ve benzer diğer dokümanlar arasında bağlantıların yer aldığı World Wide Web (WWW) teknolojileri her geçen gün daha çok hayatımızda yer almaktadır. Öncelerinde her bir birey sadece web üzerindeki bilgileri edinmekte yetinebiliyorken teknolojik gelişmelerle birlikte blog, sosyal medya etkileşimleri, özgün ve anonim internet etkileşimleri, kişisel web sayfaları, arama motoru sorguları, ziyaret edilen web sayfaları vb. gibi birçok alanda veri üreterek bu dağıtık ortamı genişletmektedir. Web üzerinde yer alan verilerin bu şekildeki artışı doğru bilgiye erişimde büyük zorlukları beraberinde getirmiştir. Belirli standartlara göre hazırlanması mümkün olmadığı için gelişi güzel yayınlanmış olan yarı yapılandırılmış ya da yapılandırılmamış veri yığınları bilişim sistemleri tarafından işlenmesinde zorluklar yaşanmaktadır. Web üzerindeki dokümanlarda özgün tasarım ve yazım stili bakımından genel metin dokümanlarından daha çok çeşitlilik bulunması sebebi ile bilginin keşfi süreci daha uğraştırıcı olmaktadır. Bu bilgiler ışığında web madenciliği, web üzerinde yer alan veri yığınlarından otomatik olarak bilgi çıkarmak amacıyla veri madenciliği ve metin madenciliği tekniklerini kullanan bir süreç olarak adlandırılabilir. Web madenciliği dört aşamadan oluşmaktadır [42,43]. Bunlar:

- Kaynakların Bulunması: Veri yığını olarak ele alınan web sayfalarının her birinde yer alan o sayfaya özgü olan içerik metinlerinin HTML kodlarından temizlenerek elde edilmesi sürecidir.

- Bilgilerin Çıkarılması: Bir önceki aşamada elde edilen metinlerdeki terimler kök ya da gövdelerine dikkat edilerek ön işleme sürecinden geçirilir. Ayrıca bu süreçte metinler içerisinde çok geçmesine rağmen tek başına önemli bir anlamı olmayan kelime grubu olarak nitelendirilen durak kelimeleri de terim listesinden temizlenir. Terim ve dokümanlar sayısallaştırılarak terim doküman matrisi elde edilir.

- Genelleştirme: Her bir web sayfasının ayrı ayrı işlemlerde ya da aynı işlemde gerçekleştirilen bilgi keşfi sürecidir.

- Analiz: Yapılan bilgi keşfi sürecinin performansının incelendiği ve sonuçlarının değerlendirildiği süreçtir.

Web madenciliğinde genellikle sunucularda, istemcilerde, internet erişiminde kullanılan vekil sunucularda ve veri tabanı sunucularından elde edilen veriler işlenmektedir. Ancak bu veriler bulundukları konum, verilerin oluşma ve toplanma şekli ve uygulama alanı gibi hususlar dikkate alarak daha detaylı incelendiğinde dört sınıfa ayrılmaktadır [44].

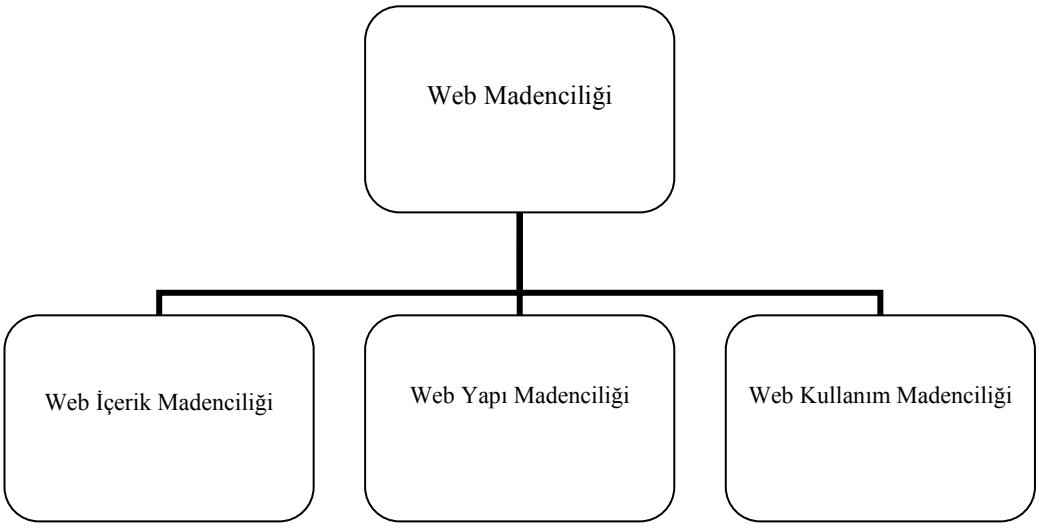
- İçerik verisi: web sayfalarının sunulduğu HTML kodları ve bu kodların içerisinde yer alan metinsel verilerle birlikte sayfa içinde yer alan çoklu ortam verileridir.

- Yapı verisi: web sayfalarının içeriklerini sunduğu bağlantı düzeni ve bağlantılara ait bilgileridir.

- Web kullanım verisi: Kullanıcıların web sayfalarındaki ziyaret süreci içerisindeki gerçekleştirdikleri işlemlere dair verilerdir. Web sayfasının ziyaret süreci, ziyaret eden kullanıcının demografik bilgileri, web sayfasındaki gerçekleştirmiş olduğu ekleme, güncelleme, silme gibi işlemler bütünü, bu tür veriler grubundadır.

- Kullanıcı profili: Web sayfalarını ziyaret eden kişilerin izin verdiği ölçüdeki kişisel bilgilerinin yer aldığı verilerdir. Bu tür verilere kullanıcının arama motorlarındaki arama sorguları, internet özgeçmişi ve sosyal medya hesapları gibi daha detaylı kişisel veriler dâhil edilmiştir.

Şekil 4.1'de gösterildiği gibi web madenciliği uygulandığı alanlara ve verilerin oluşma yöntemlerine göre web içerik madenciliği, web yapı madenciliği, web kullanım madenciliği olarak üç sınıfa ayrılmaktadır.



Şekil 4.1. Web Madenciliği Sınıfları

4.1. Web İçerik Madenciliği

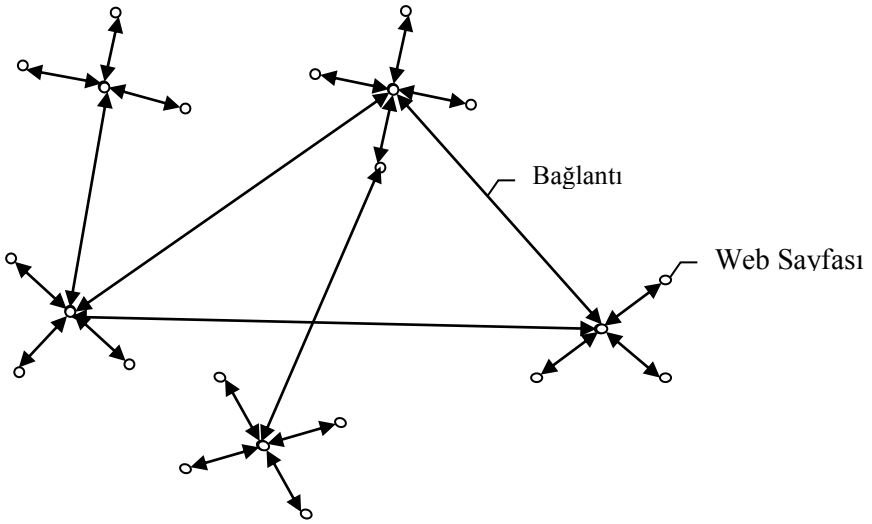
Web içerik madenciliği web sayfalarında yayınlanan içeriklerin işlenmesiyle başlık ve konu tesbiti, web sayfalarının kümelenmesi, web sayfalarının sınıflandırılması ve ilişkili örüntülerin çıkarılması gibi araştırmalarda kullanılmaktadır. Bunun yanı sıra web sayfalarından otomatik olarak gizli ilişki ve yapıların çıkarılması web içerik madenciliğin giderek artan uygulaması haline gelmiştir. Web sayfalarında içerikler genellikle metinsel olarak yayınlanmaktadır. Web içerik madenciliği web sayfalarında, içerisinde metinleri, bağlantıları ve çoklu ortam verilerini barındırabilen HTML kaynak kodlarını temel veri seti olarak ele alır [42,43].

Web madenciliğinde verilerin web sayfalarından elde edilmesi için örümcek ya da bot adı verilen yazılımlar ile gerçekleştirilmektedir. Bu yazılımlar vasıtası ile elde edilen metinsel veriler HTML etiketlerinden, sayfada yer alan özgün içerik dışındaki diğer verilerden temizlenmek amacıyla ön işlem sürecinden geçer. Önişlem sürecinden sonra elde edilen ilgili web sayfasına özgün olan içerik metin madenciliği sürecine dâhil olmaktadır.

Web içerik madenciliğinin en yaygın olanlarından birisi arama motoru ve dizinleme işlemleridir. Bu tür işlemlerde arama sorgusu ya da metni girildiğinde karşılık olarak aranan içerikle ilgili bilgiler sıralanır. Arama motorlarının ilgi duyulan bilgileri listeleyerek sunmaları arka planlarındaki içerik tabanlı dizinleme algoritmalarına dayanmaktadır.

4.2. Web Yapı Madenciliği

Web sayfaları arasındaki bağlantılar vasıtasıyla birbirileri arasındaki ilişkileri dikkate alan web yapı madenciliği bilgiye erişim sistemlerinde kullanılmaktadır. Geleneksel bilgiye erişim sistemlerinde sadece içeriğe odaklanılırken web teknolojisinin sunmuş olduğu bağlantılar bu bilgiye erişim sistemlerini daha verimli kılmaktadır. Bu yönüyle web yapı madenciliği web içerik madenciliğine destek olmak amacıyla kullanılmaktadır. Web sayfalarının bir düğüm olarak ve sayfaların birbirileri ile olan bağlantılarının temsil edildiği Şekil 4.2'deki gibi bir graf yapısı dikkate alınmaktadır.



Şekil 4.2. Web Graf Yapısı

Web sayfaları arasında köprü görevi üslenen bağlantılar iki sayfa arasındaki en kısa yolun oluşmasına olanak sağlarken aynı zamanda bu iki sayfa arasındaki ilişki ve benzerliğin de göstergesi olmaktadır. Bu yönü ile içerik madenciliğinde benzer ve ilişkili dokümanların tespit edilmesi ya da işlenmesi hususunda önemli rol üstlenmektedir.

4.3. Web Kullanım Madenciliği

İnternet kullanıcılarının web üzerinde bırakmış oldukları izler olarak bilinen ve sunucularda kayıt altına alınan erişim kayıt verileri web kullanım verisi olarak adlandırılmaktadır. Vekil sunucularda, web sayfalarının yayınlandığı sunucularda, web tarayıcısı kayıtları gibi erişim ve kullanım bilgilerinin tutulduğu diğer servislerde kayıt altına alınan bu veriler genellikle kullanıcıların erişim sağladığı Internet Protokol (IP) adresleri, erişimin gerçekleştiği web sayfa bilgileri, erişim zamanı, web tarayıcısı ve işletim sistemi gibi bilgilerden oluşmaktadır. Bu veri gruplarına kullanıcı profil verisi olarak adlandırılan kullanıcıların demografik bilgisini içeren veri grubu dahil olduğunda daha detaylı veri grubu elde edilmektedir. Web sayfalarını ziyaret eden kullanıcıların daha önce ziyaret ettiği web sayfaları, cinsiyeti, konumu, geçmişte yapmış olduğu alışverişler gibi verilerin yer aldığı kullanıcı profil verilerinden elde edilen bilgiler ışığında daha verimli bir kullanım madenciliği gerçekleştirilmesi mümkün olabilir.

Web kullanım madenciliğinin en temel veri kaynağı sunucular üzerinde tutulan log dosyalarıdır. Sunucular üzerinde belirli zaman aralıklarında tutulan bu kayıt verileri siteye ziyaret kayıtları, mail kayıtları, web sayfasında gerçekleşen hatalı erişim kayıtları ve dosya transferlerinin tutulduğu File Transfer Protocol (FTP) kayıtlarıdır. Bunların yanı sıra her web sayfasının sisteminde kendilerine özgü erişim ya da işlem kayıtları da yer alabilir. Bu tür kayıtlar bu grupta incelenebilir.

Web kullanım verilerinin işlenmesiyle web sayfalarının hedef kitlelere ulaşması, hedef kitlelerin tercihlerinin belirlenmesi, hedef kitlelerin ihtiyacının gözlemlenmesi gibi işlemler yapılabilir. Web içerik madenciliği, web yapı madenciliği ve web

kullanım madenciliğinin her üçünün dikkate alındığı bir erişim sistemi gerçekleştirildiğinde daha hassas bir çıkarım yapılması mümkün olabilir. Örneğin kullanım verileri dikkate alındığında kullanıcıların web sitesi üzerinde ne kadar zaman harcadığını ya da aktif olduğu zamana dikkat edildiğinde kullanıcının bu siteye olan ilgisi çıkarılabilir. Öte yandan web sayfasındaki linkler vasıtası ile ilişkili web sayfalara ve web sayfasındaki metin türündeki içeriklere dikkat edilerek daha ilgi çekici sayfalar listelenebilir. İyi sonuçlar listeleyen bir doküman dizinleme sistemlerinde ya da arama motorlarında bu üç web madenciliği sınıfının dikkate alınması gereklidir.

5. ALTERNATİF DÜŞÜK RANK MATRİS AYRIŞIMI İLE GİZİL ANLAMSAL DİZİNLEME

Bu kısım, tez boyunca ihtiyaç duyulacak Lineer cebir temel kavramlarına ayrılmıştır.

5.1. Lineer Cebirle İlgili Temel Kavramlar

Tanım 1 (Vektör) $i = 1,2,3,...,n$ ve $x_i \in \mathbb{R}$ olmak üzere n adet reel sayıdan oluşan x'in

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \tag{5.1}$$

şeklinde gösterilen sayı dizisidir.

Tanım 2 (Sıfır Vektörü) Bütün elemanları sıfır olan vektörlere sıfır vektörü denir. Boyutları fark etmeksizin 0 ile temsil edilir.

Tanım 3 (Birim Vektörü) $i-$inci bileşeni 1 ve diğer bileşenleri 0 olan vektöre $i-$inci birim denir ve e_i ile temsil edilir.

Tanım 4 (Matris) m satırlı ve n sütunlu reel sayı dizisinden oluşan $m \times n$ boyutlu reel değerli A matrisi

$$A = [a_{ij}] = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m1} & \cdots & a_{mn} \end{pmatrix}, \; a_{ij} \in \mathbb{R}, \; i = 1,2,3,...,m \text{ ve } j = 1,2,3,...,n \tag{5.2}$$

biçiminde tanımlanır. Matrislerin $i-$inci satır ve $j-$inci sütununda yer alan elemanı a_{ij} ile temsil edilir.

Tanım 5 (Kare Matris) $m \times n$ boyutlu reel A matrisi $m = n$ şartını sağlıyor ise bu matris kare matris olarak isimlendirilir ve A matrisine n-inci dereceden kare matris denir.

Tanım 6 (Sıfır Matrisi) Eğer matrisin tüm elemanları sıfır ise bu matrise sıfır matrisi denir ve boyutuna bakılmaksızın 0 ile temsil edilir.

Tanım 7 (Birim Matris) Bir kare matris olan $I = [l_{ij}]$

$$l_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{5.3}$$

şartını sağlıyor ise bu matrise birim matris denir ve I ile temsil edilir.

Tanım 8 (Matrisin Rankı) $A \in \mathbb{R}^{m \times n}$ matrisinin rankı

$$rank(A) = \dim(range(A)) \tag{5.4}$$

olarak tanımlanır. Eğer $rank(A) = \min(m, n)$ şartı da sağlanıyor ise bu matrise tam dereceli denir. Diğer durumlarda ise eksik dereceli olarak tanımlanmaktadır.

Tanım 9 (Tekil Olmayan Matris) $A \in \mathbb{R}^{m \times n}$ hem kare hem de tam dereceli ise tekil olmayan matris olarak adlandırılır.

Tanım 10 (Matrisin Tersi) Tekil olmayan bir $A \in \mathbb{R}^{m \times n}$ matrisiyle

$$A^{-1}A = AA^{-1} = I \tag{5.5}$$

şartını sağlayacak bir A^{-1} matrisi varsa bu matrise A matrisinin tersi denir.

Tanım 11(Matrisin Transpozu) $A \in \mathbb{R}^{m \times n}$ matrisinin transpozu A^T ile gösterilir ve A'nın sütunlarını sırasıyla satır olarak yazmakla elde edilir.

Tanım 12(Ortogonal Matris) n boyutlu bir kare matris olan Q matrisi

$$QQ^T = Q^TQ = I \tag{5.6}$$

şartını sağlıyorsa Q matrisine ortogonal bir matris denir.

Tanım 13(Üçgensel Matris) $U = \left(u_{ij}\right)$ matrisinde $u_{ij} = 0, i > j$ ise U matrisine üst üçgensel matris adı verilir ve

$$U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ 0 & & & u_{mn} \end{pmatrix} \tag{5.7}$$

şeklinde gösterilir. $L = (l_{ij})$ matrisinde $l_{ij} = 0, i < j$ ise L matrisine alt üçgensel matris adı verilir ve

$$L = \begin{pmatrix} l_{11} & & & 0 \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{m1} & l_{m1} & \cdots & l_{mn} \end{pmatrix} \tag{5.8}$$

şeklinde gösterilir.

5.2. Düşük Rank Matris Ayrışımı

Düşük rank matris ayrışımı veri biliminde kullanılan önemli bir tekniktir. Düşük ranklı matris ayrışımında indirgenmiş şekilde temsil edilen büyük boyuttaki verilerde yer alan gizli örüntülerin keşfi amaçlanır. Bu yönü ile düşük dereceli matris ayrışımları büyük boyutlu matrislerin indirgenmesinde, eksik ya da kayıp verileri olan matris verilerinin tamamlanmasında ve kümeleme çalışmalarında kullanılan yöntemlerden biridir. Yaygın olarak bilinen düşük rank matris yaklaşımları temel

bileşenler analizidir (Principal Component Analysis-TBA). Düşük rank matris yaklaşımı hesaplamalarında ya da GAA'da kullanılan en yaygın matris ayrışımı ise tekil değer ayrışımıdır TDA [45]. Ancak mevcut doküman yığınına yeni dokümanların eklendiği durumda vektör uzayının yeniden oluşturulması ya da güncellenmesi işlemlerinin hesaplama karmaşası yüksek maliyetlidir. Bu nedenle TDA'ya alternatif olarak mevcut vektör uzayının güncellenmesinde TDA'ya göre daha düşük hesaplama maliyeti olan ULV ve URV matris ayrışımları önerilmektedir [46]. Güncelleme yöntemlerinden biri olan Folding-in metotu mevcut anlamsal yapıya dikkat ederek vektör uzayına yeni ögelerin yerleşmesini sağlarken vektör uzayının elde edilmesinde kullanılan matrislerin ortogonalliğini bozma ihtimali de bulunmaktadır [47]. Bu nedenle bu yöntemle yapılan güncelleme işlemleri anlamsal yapının bozulmasına sebep olabilir. Güncelleme işlemlerinin anlamsal yapıyı bozmadan en doğru şekilde yapılabilmesi için hesaplama maliyeti büyük olmasına karşın ortogonal matris yapısını bozmayan yaklaşımdaki işlemlerin yeni eklemeleri de hesaba katarak tekrar yapılması gerekmektedir. Böylece mevcut yapıya eklenen dokümanların etkisi tam anlamı ile anlamı bozmadan gerçekleşmiş olur.

5.2.1. Tekil Değer Ayrışımı

$A \in \mathbb{R}^{m \times n}$, $m \geq n$ olması koşulu ile A matrisinin TDA'sı

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T \tag{5.9}$$

ile gösterildiği gibi üç matrisin çarpımı şeklindedir. Formülde yer alan sırasıyla sol ve sağ tekil vektörü olarak bilinen $U \in \mathbb{R}^{m \times n}$ ve $V \in \mathbb{R}^{n \times n}$ ortogonal matrislerdir. Diğer bir deyişle $U^T U = U U^T = I_m$ ve $V^T V = V V^T = I_n$ şartı sağlanmaktadır. Köşegen olan ve elemanları A'nın tekil değerleri olan $\Sigma = diag(\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_n)$ matrisi ise

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq \sigma_{k+1} = \cdots = \sigma_n = 0 \tag{5.10}$$

şartını sağlamaktadır. Bu durumda A'nın rankı k'dır. Ancak, karşılaşılabilecek sayısal hesaplama işlemlerindeki yuvarlama hatalarından dolayı rank kavramı esnekleştirilerek sayısal rank A matrisine rank değeri atanır. A matrisinin sayısal rankı k olduğunda,

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > \delta \geq \sigma_{k+1} \geq \cdots = \sigma_n \tag{5.11}$$

olur. Formülde yer alan δ eşik değeri olarak bilinir ve beklenti σ_k ile σ_{k+1} değerleri arasındaki farkın anlamlı büyüklükte olmasıdır. Fakat, metin madenciliği uygulamalarında tekil değerler arasındaki farkların değişim oranı büyük ölçüde benzer özellik gösterdiğinden dolayı bu değerin tespit edilmesi zor bir problem olarak göze çarpmaktadır. Literatürde yapılan çalışmalar matrisin boyutu dikkate alınmaksızın k'nın 200 ile 300 arasında bir değer aldığını göstermektedir [48-51].

GAA ile ilgili çalışmalarda A matrisi yerine A matrisinin rank-k yaklaşımı (A_k) kullanılmaktadır. A_k, A'nın en büyük k adet tekil değerleri dışındaki değerleri sıfır kabul edilerek elde edilir. A_k

$$A_k = U_k \sum_k V_k^T \tag{5.12}$$

ile elde edilir. Formülde yer alan $\sum_k = diag(\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_k)$ köşegen matrisi, U_k ve V_k ise sırası ile U ve V matrislerinin ilk k sütunlarını temsil etmektedir.

Gizli Anlamsal Analiz modellemesinde terimleri $U_k \sum_k$ çarpımı, dokümanları ise $\sum_k V_k^T$ çarpımı temsil etmektedir. Burada boyut n'den k'ye düşürülerek hem işlem süreci büyük oranda azalmakta hem de doğru sonuçlar bulmayı engelleyen gürültüler ve etkisiz veriler hesaba katılmamaktadır. Böylece daha verimli performans elde edilmektedir

5.2.2. Kesik ULV Ayrışımı

$A \in \mathbb{R}^{m \times n}$, $m \geq k \geq n$ olması koşulu ile sayısal rankı k olan A matrisinin kesik ULV ayrışımı

$$A = ULV^T + E \tag{5.13}$$

biçiminde gösterilir. Formülde yer alan $U \in \mathbb{R}^{m \times k}$, $V \in \mathbb{R}^{n \times k}$

$$U^T U = I_m \text{ ve } \quad V^T V = I_n \tag{5.14}$$

koşullarını sağlayan sol ortogonal matrisleri, $L \in \mathbb{R}^{k \times k}$ tekil olmayan alt üçgensel matris ve $E \in \mathbb{R}^{m \times n}$ ise hata matrisidir. L matrisinin tekil değerleri A matrisinin tekil değerlerine yaklaşır [8,51].

5.3. Gizil Anlamsal Dizinleme

Metin analizi, büyük boyuttaki metin türündeki veriyi, kendisini temsil eden daha küçük boyutlu veri yapısına çevirerek genel anlamının tespitini amaçlayan çalışmalar bütünüdür. Metin içindeki kelime, cümle gibi dil ögelerini inceleyen ve metin içindeki konumuna göre ve metnin geneline göre sıklığını, anlamını ve etkisini inceleme aşamalarıdır. Metin içindeki en küçük birim kelimeler bir araya gelerek anlamlı cümleleri ve sonuç olarak hepsi birleşerek anlamlı örüntüleri oluşturmaktadır. Metin madenciliği ise bu örüntüler içerisinde farkında olmadığımız işlem yapılmadığında dikkat çekmeyen örüntüleri gün yüzüne çıkarmaktadır. Metin analizinde genel olarak sık geçen kelimeler, anahtar kelimeler, metinde birlikte geçen kelimeler ve bu kelimelerin birbirileriyle ilişkileri göze alınarak işlemler yapılmaktadır [29,52,53].

Büyük boyutlu veriler üzerinde işlem yapmak oldukça maliyetli bir süreç gerektirdiği için özellik çıkarım yöntemleri uygulanır. Özellik çıkarım yöntemleri çok boyutlu ve

büyük veri kümelerinden verimli sonuç almak, işlem ve hafıza maliyetini azaltmak için sadeleştirerek temsilci değerlerini oluşturmaktır [54,55].

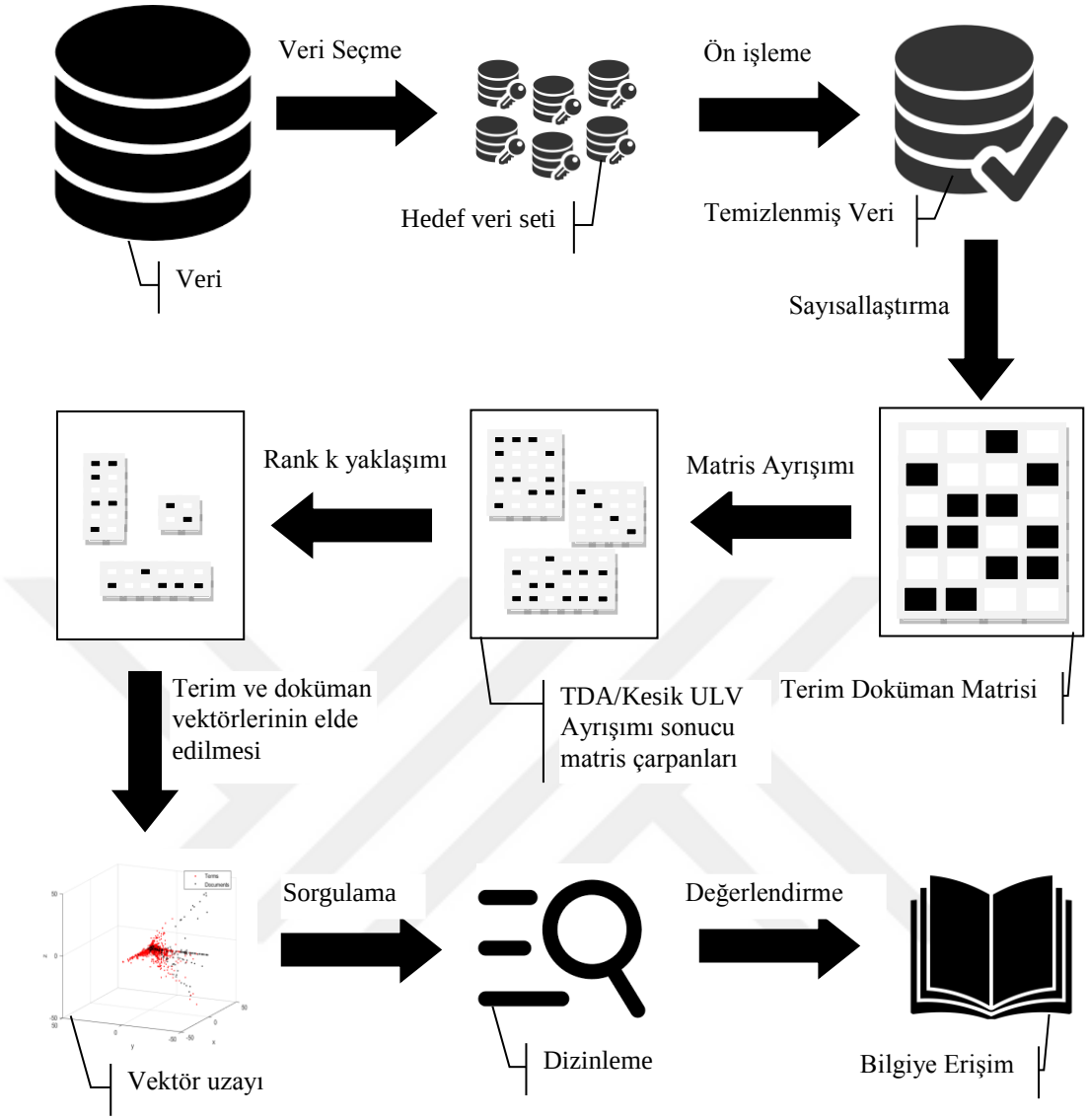
GAA boyut indirgeme tabanlı bir yaklaşım olması sebebiyle veri kümesindeki önemli olan veri gruplarını işleme dâhil etmektedir. Veri yığınında yer alan ve anlama katkı etmeyen ve anlamı olumsuz şekilde etkileyen veriler ise bu işleme dâhil olmamaktadır. Bunun için terim ve dokümanlar arasındaki gizli anlamsal yapıyı bulmada terim-doküman matrisinin düşük ranklı yaklaşımı kullanılmaktadır.

GAA, özellik çıkarım yöntemi olarak matris ayrışım metotlarını kullanarak bir vektör uzayında terimler ve dokümanlar arasındaki gizli yapıyı ortaya çıkarır. Diğer bir deyişle GAA, kelimelerin bir arada geçme sıklıkları ve bir arada bulunmalarını lineer cebir yöntemleriyle temsil ederek dokümanlar içerisindeki gizli anlamı bulma yöntemidir. Kelimeler ve kelimelerin bulunduğu dokümanlar matris üzerinde satır ve sütün olarak tanımlanır, her kelimenin ağırlık değeri belirli metotlara göre hesaplanarak indisine atanır. Elde edilen terim-doküman matrisi ise daha sonra matris ayrışım metotları ile çarpanlarına ayrılarak hem terim hem de dokümanlar için bu ayrışım sonucu elde edilen matrisler kullanılarak yeni matrisler elde edilir. Terim ve doküman matrislerinin değerleri incelenmek için vektörel düzlemde gösterildiğinde birbirileri arasındaki ilişkiler görülür. Gözle görülebilen ya da görülemeyen bu ilişkileri bulmak için terim ve dokümanların kendilerine ait indislere göre bulundukları konumları dikkate alarak yapılabilecek sınıflama, kümeleme ve dizinleme gibi işlemler yapılır [56]. GAA kullanılarak gerçekleştirilen bir sistemde sorgu cümlecikleriyle doküman yığınları içerisinden istenen dokümanların listelenmesi ya da referans olarak verilen dokümana göre diğer dokümanların benzerliği dikkate alınarak listelenmesi işlemine Gizil Anlamsal Dizinleme (GAD) denir.

GAD'da bilgiye erişimin gerçekleştiği süreçte (Şekil 5.1.) en başta hedef verilerin seçimi gerçekleştirilir, bu verilerin ön işlem sürecinden ve temizleme süreçlerinden geçerek durak kelimelerinden, noktalama işaretlerinden arındırılır. Ayrıca, kök ya da gövde şeklinde işlenecek duruma getirilmiş olan kelimelerin bulundukları dokümanlardaki frekanslarına dayalı olan ağırlıkları hesaplanarak terim doküman

matrisi elde edilir. Elde edilen bu matrise her iki algoritmada kullanılmak üzere TDA ya da Kesik ULV ayrışımı uygulanır. Matris ayrışımı sonucunda elde edilen matrislere hem boyut indirgeme amacı ile hem de gürültü ve etkisiz verilerin işleme katılmasını önlemek amacı ile rank-k yaklaşımı uygulanır. Rank-k yaklaşımı sonucunda sırası ile sol ve sağ ortogonal matrislerin tekil değer matrisi ile çarpımı ile terim temsilci matrisi ve doküman temsilci matrisi elde edilir. Elde edilen terim temsilci matrisindeki her bir satır terim doküman matrisinde yer alan aynı indisli terimin yerini tutmaktadır. Aynı şekilde doküman temsilci matrisindeki her bir sütun ise terim doküman matrisinde yer alan aynı indisli dokümanın yerini temsil etmektedir. Böylece terim ve doküman temsilci matrislerinden elde edilen terim ve doküman vektörleri aynı vektör uzayında gösterilir. Vektör uzayının elde edilmesinden sonra ise yapılacak sorgular birer doküman gibi dikkate alınarak terimlerinin ağırlığı terim doküman matrisini oluştururken yapılan ağırlık fonksiyonları dikkate alınarak hesaplanır. Sorgu cümleciğine ait vektörün, vektör uzayındaki konumu folding-in metoduna göre belirlenir. Daha sonra bu konum bilgileri dikkat edilerek her bir dokümanın sorgu cümleciğine olan benzerliği hesaplanır ve bütün dokümanlar en çok benzer olandan en az olanına doğru listelenir.

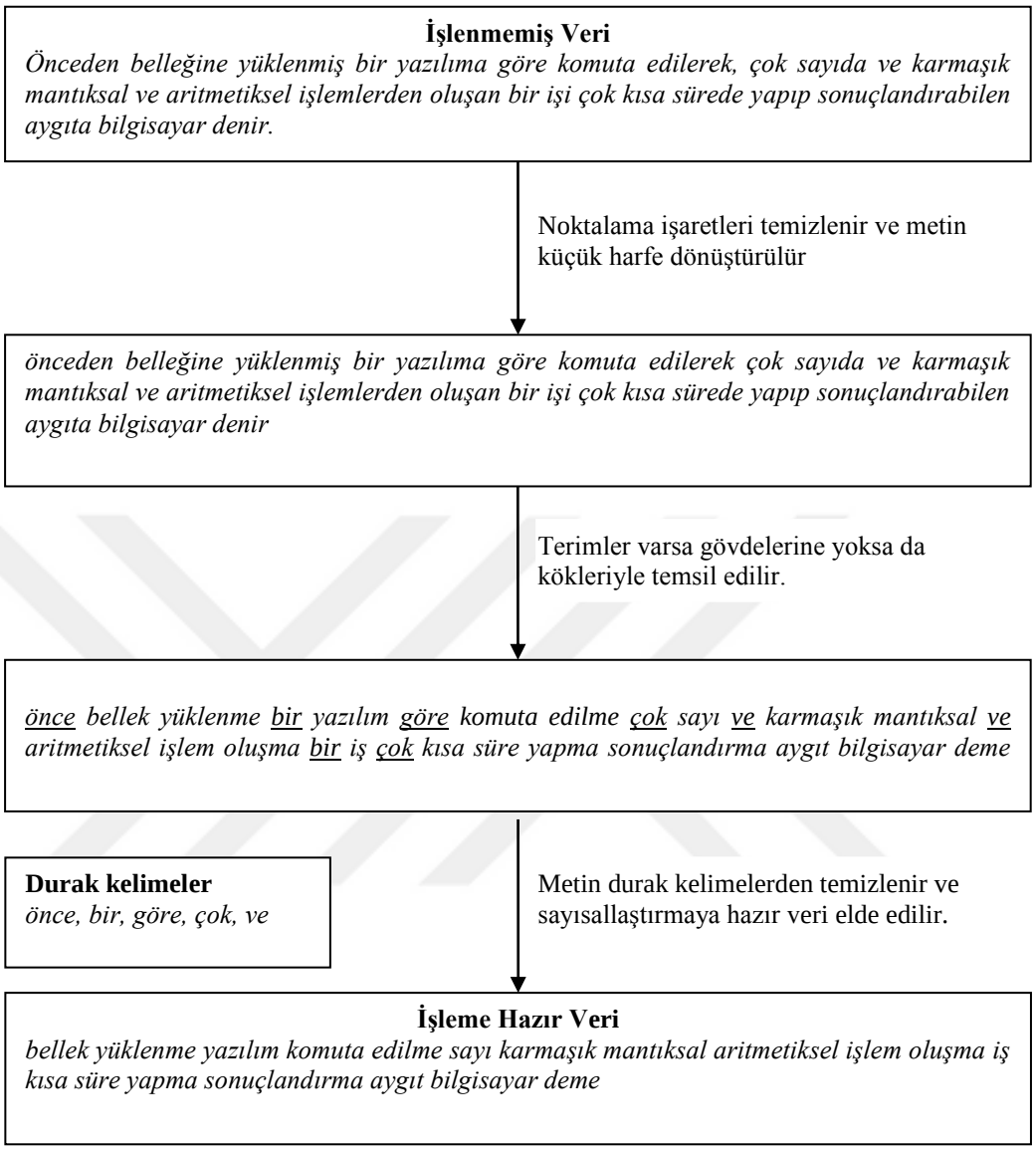
Listeleme işlemlerinin sonucunda doküman içerisinde sorgu ile ilgili olan dokümanlar ve sorgu sonucunda geri dönen dokümanlar dikkate alınarak performans değerlendirmesi yapılır. Bu şekilde sorgu işlemleri sonucunda dizinlenen dokümanların başarım oranı belirlenmiş olur. Yapılan testler ve bu testlerin sonucunda alınan geri dönütler sayesinde en doğru ağırlıklandırma metodu ya da rank$k$ yaklaşımındaki k değeri belirlenir. Böylece büyük bir veri yığınındaki istenen dokümanların başarılı bir şekilde listelenmesi gerçekleştirilir.



Şekil 5.1. GAD Süreci

5.3.1. Veri Seçimi ve Ön İşleme Süreci

GAD uygulanacak verilerin seçimi ve bu verilerin ön işleme süreci bilgiye erişim süreçlerinin tamamında olduğu en çok kaynak ve zaman alan aşamalardan birisidir. Belirlenen amaç çerçevesinde verilerin seçiminde işlenecek verilerin kaynakları ve türlerinin de belirlendiği bu aşamada etkisiz ya da gürültü niteliği taşıdığı önceden bilinen veri gruplarının olmamasına dikkat edilmelidir.



Şekil 5.2. Ön işleme süreci

GAD işleminin uygulanacağı doküman yığınlarında yer alan yazım biçimlerinden dolayı yaşanabilecek yanlış sonuçların önüne geçilmesi amaçlanır. Bu nedenle dokümanların içerisinde yer alan bütün terimler sadece büyük ya da küçük harf şekline dönüştürülür. Böylece hem işlem sürecinde aynı anlamı taşıyan verilerin farklı anlaşılmasının önüne geçilir hem de işlem sonrasında yapılacak sorgulamalardaki yazım şekillerinin sonucu etkilemesine imkân verilmemiş olur.

İşlem öncesi (Şekil 5.2.) yapılan temizleme aşamasında ilk olarak veri yığınında sadece terim ve doküman şeklinde verilerin işlenebilmesi amacı ile noktalama işaretlerinin temizlenmesi sağlanır. Daha sonra da hedef olarak seçilen verilerin anlamı bozabilen gürültü niteliği taşıyan anlama olumlu ya da olumsuz katkısı olmayıp işlem süresini uzatan durak kelime olarak adlandırılan bağlaç, edat zamir gibi tek başına anlamı olmayan kelimelerin çıkarılması gerekmektedir. Böylece dokümanların içerisinde sayıca daha çok bulunan ancak anlamı etkilemeyen bu kelimelerin çıkarılmasıyla yapılacak işlem sürecinin zaman ve kaynak yönünden maliyeti azalır. Bunun yanında sadece anlamı daha çok etkileyen kelimelerin işleme dâhil olmasını sağlayarak bilgiye erişim performansının da artmasına neden olur.

Çizelge 5.1. Örnek bir terimin kök ya da gövdesine ayrıştırılması

Kelime	**Kök**	**Gövde**	**İşlenen kısım**
kitapçıdan	kitap	kitapçı	kitapçı
kitaplık	kitap	kitaplık	kitaplık
kitabeden	kitap	kitabe	kitabe
kitapçılık	kitap	kitapçılık	kitapçılık
kitaplaştırma	kitap	kitaplaştırma	kitaplaştırma
kitabı	kitap	-	kitap
kitaptan	kitap	-	kitap
kitap	kitap	-	kitap

Dokümanlarda yer alan kelimelerin kullanılan dilin yapısına göre türediği gövde varsa gövdeleriyle temsil edildiği, yoksa kelime kökü ile temsil edilmesi gerekmektedir. Böylece terimler taşıdıkları anlamlara göre tek bir formatta standardize edilerek işleme hazır edilir. Aksi durumda kelimeler aldığı her bir çekim eki yüzünden yeni bir kelime olarak dikkate alınır ve anlamı aynı olan bu kelimeler farklı kelime olarak hesaplanarak farklı anlam taşıyor gibi işlenir. Bu şekilde işlenen

terimler veri grubunda yer alan örüntülerin daha duyarsız ve hatalı sonuçların çıkmasına sebep olur.

Bu çalışmada kelimelerin köklerini ve gövdelerinin tespiti için geliştirilen yazılım eklentisi ile Türkçe kök ve gövde biçiminde yer alan terimler listesi dikkate alınarak gerçekleştirilmiştir. Çizelge 5.1'de örnek bir terimin kök ve gövdesine kadar ayrıştırıldığı örneği görebilirsiniz.

5.3.2. Terim-Doküman Matrisinin Elde Edilmesi

Veri yığınında yer alan ön işlem sürecinin tamamlanmasının ardından standart bir biçimde işleme hazır duruma gelen terimler ve dokümanlar GAA'da vektör uzayında sembolize edilmek ve anlamsal örüntülerin ortaya çıkarılması için ilk önce terim doküman matrisiyle temsil edilmesi gerekmektedir. Bu amaçla GAA işleminde girdi olarak kullanılmak üzere $A \in \mathbb{R}^{m \times n}$ terim doküman matrisi elde edilir [45]. Bu amaçla dokümanlar ve dokümanlar içerisinde yer alan bütün terimler sayısal bir anahtar değeri alarak sayısallaştırılır. Örneğin 1 numaralı olarak atanan dokümanın numarası başka hiçbir doküman için kullanılmaz ve 1 numaralı terim için atanan terim numarası diğer dokümanlarda var ise yine 1 numaralı terim değerini alır ve yeniden aynı terim için numaralandırma işlemi gerçekleştirilmez. Bütün dokümanlar ve terimler sayısallaştırma işleminden geçtikten sonra her bir dokümanda bulunan terimlerin ağırlığı belirli formüllerle hesaplanarak terim doküman matrisini oluşturulur. Terim-doküman matrisinin satırları terimleri, sütunları ise dokümanları temsil eder. m tane terim ve n tane doküman bulunan bir veri grubu $m \times n$ boyutlu bir matris ile temsil edilir ve bu matris

$$A = [a_{ij}] = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m1} & \cdots & a_{mn} \end{pmatrix} \tag{5.15}$$

şeklinde gösterilir. Burada a_{ij} sırası ile i-inci terimin j-inci dokümandaki ağırlık değerini taşır. Çoğu zaman terim sayısı doküman sayısından fazla olduğu için $m \geq n$ olarak dikkat edilse de arama motoru ya da çok büyük veri grubunun olduğu çalışmalarda terim sayısı artan doküman sayısına oranla belirli bir noktadan sonra artmayacağı için $n \geq m$ olabilir [57]. Bu çalışma da $m \geq n$ şartının sağlandığı görülmüştür. Terim-doküman matrisi her bir terim ve her bir doküman için ayrı ayrı incelendiğinde terimler için terim vektörlerinden, dokümanlar için ise doküman vektörlerinden oluşmaktadır. Her bir doküman için bütün terimlerin kendisindeki ağırlığının listelendiği doküman vektörleri ve her bir terimin bütün dokümanlardaki ağırlığının listelendiği terim vektörleri yer alır. i-inci terim için terim vektörü T_i ile gösterilirken j-inci doküman için doküman vektörü D_j ile gösterilir.

$$T_i = \begin{pmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \end{pmatrix} \tag{5.16}$$

$$D_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix} \tag{5.17}$$

$$A = [a_{ij}] = \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_m \end{pmatrix} = \begin{pmatrix} D_1 & D_2 & \cdots & D_n \end{pmatrix} \tag{5.18}$$

Terim-doküman matrisini oluşturan her bir terimin bulundukları her bir doküman üzerindeki anlamsal etkisini belirten ve ağırlık değeri olarak adlandırılan a_{ij}, işlem sürecinin başarısını en çok etkileyen etkenlerden biridir. a_{ij}'nin belirlenmesinde yapılan ağırlıklandırma işlemi, ilgili terimin bulunduğu dokümandaki ve bütün doküman yığınındaki ayırt ediciliğine dikkat edilerek yapılmalıdır. Örneğin bir doküman içerisinde çok sayıda tekrar eden bir terim bu doküman için ayırt edici olurken, aynı terim dokuman yığını içindeki diğer dokümanlarda da geçme sıklığı çok ise bu terimin doküman yığını içerisinde ayırt ediciliği olmayacaktır. Bunun

aksine herhangi bir terimin bulunduğu dokümanda geçme sıklığı çok ve bütün doküman yığını içerisindeki geçme sıklığı az olduğunda da ayırt ediciliği yüksek olmaktadır. Bu nedenle bu terimin ağırlık değeri bulunduğu sıklık sayısıyla doğru orantılı iken bütün doküman yığını içerisindeki sıklık değeri ile ters orantılı olarak hesaplanır. O halde i-inci terimin j-inci dokümandaki ağırlık değeri olarak a_{ij}

$$a_{ij} = L(i, j) \times G(i) \tag{5.19}$$

formülüne göre hesaplanır. Formüldeki $L(i, j)$ i-inci terimin j-inci dokümandaki yerel ağırlığını temsil ederken $G(i)$ ise i-inci terimin bütün doküman yığını için genel ağırlığını temsil etmektedir.

Terim-doküman matrisinin en önemli ögesi olan a_{ij} değerinin belirlenmesinde hem yerel ağırlıklandırma hem de genel ağırlıklandırma işlemlerinde için farklı yöntem ve teknikler kullanılmaktadır.

5.3.2.1. Yerel Ağırlıklandırma Teknikleri

Terim-doküman matrisi oluşturulurken hesaplanan ağırlıklandırma formülünde yer alan $L(i, j)$ için i-inci terimin j-inci dokümandaki ağırlığını bulmak adına terimin bulunduğu dokümandaki bulunma durumu ve geçme sıklığına dayalı hesaplanan ölçüm işlemleridir. Yapılan çalışmalar incelendiğinde yerel ağırlıklandırma yöntemleri;

- İkili Ağırlıklandırma
- Terim Frekansı (TF) ile Ağırlıklandırma
- Logaritma Ağırlıklandırma ile sıralanabilir.

- **İkili Ağırlıklandırma**

Bu yönteme göre terimlerin yerel ağırlığı bulundukları dokümanlardaki varlığına ya da yokluğuna dayalıdır. Maliyet açısından çok kazanç sağlanacak yöntem gibi görünse de anma ve hassasiyet açısından performansı iyi olmadığı için çok tercih edilmeyen yöntemdir. Çok küçük veri gruplarının dışında kullanılması önerilmez. $L(i, j)$;

$$L(i, j) = \begin{cases} 1 & tf_{ij} > 0 \\ 0 & tf_{ij} = 0 \end{cases} \tag{5.20}$$

formülüne göre hesaplanır. Buna göre $L(i, j)$ i-inci terimin j-inci dokümandaki geçme sıklığı olan tf_{ij}'nin sıfırdan büyük olduğu durumlarda bir değerini, tf_{ij}'nin sıfır yani terimin dokümanda bulunmadığı durumlarda da sıfır değerini alır.

- **Terim Frekansına Göre Ağırlıklandırma**

TF'e göre ağırlıklandırma yönteminde terimlerin ağırlığı bulundukları dokümanlardaki geçme sıklığıyla ağırlıklandırılır.

$$L(i, j) = tf_{ij} \tag{5.21}$$

formülünde olduğu gibi $L(i, j)$, i-inci terimin j-inci dokümandaki ağırlığı, tf_{ij} değerini alır. Maliyeti düşük olmayan bu metot hassasiyet açısından irdelendiğinde ortalama değerin çok üstünde ya da altında olan ağırlık değerlerinin bulunduğu durumlarda yanlış sonuçlar üretebilmektedir.

- **Logaritma ile Ağırlıklandırma**

Logaritma ile göre ağırlıklandırma yönteminde terimlerin ağırlığı bulunduğu dökümanlardaki geçme sıklığı normalize edilerek hesaplanmaktadır. Böylece terim frekansında karşılaşılan ortalama değerin çok üstünde ya da altında yer alan terim frekansından dolayı performans kaybının önüne geçilmiş olmaktadır. Bu yöntemde

$$L(i,j) = \begin{cases} \log(tf_{ij} + 1) & tf_{ij} > 0 \\ 0 & tf_{ij} = 0 \end{cases} \tag{5.22}$$

formülüne göre hesaplanan $L(i,j)$ i-inci terimin j-inci dokümandaki geçme sıklığı logaritma işlemi ile normalize edilerek gerçek değerinden daha az bir değer aralığında değer alır. Böylece bütün terimlerin geçme sıklığı baz alındığında ortalamanın çok üstünde ya da altında değer alan dolayısı ile işlem sonunda yanlış çıktılar almaya sebep olan ağırlıklandırmanın önüne geçilmiş olur.

5.3.2.2. Genel Ağırlıklandırma Teknikleri

Terim-doküman matrisi oluşturulurken hesaplanan ağırlıklandırma formülünde yer alan $G(i)$ için i-inci terimin bütün doküman yığınındaki temsil değerini bulmak adına yapılan hesaplama teknikleridir. Yapılan çalışmalar incelendiğinde genellikle terimlerin her bir dokümandaki toplam geçme sıklığı, terimin bulunduğu doküman sayısı ya da terimin bulunduğu dokümandaki geçme sıklığı gibi faktörlere göre hesaplanmaktadır. Bilinen örnekleri ise:

- Normal Ağırlıklandırma
- Ters Doküman Frekansına (TDF) Ağırlıklandırma
- GFIDF Ağırlıklandırma
- Entropi Ağırlıklandırma olarak sıralanabilir.

- **Normal Ağırlıklandırma**

Doküman yığınındaki her bir terimin, bütün dokümanlardaki geçme sayılarının kareleri toplamının tersi alındıktan sora karekökü alınarak hesaplanan genel ağırlığıdır. Formül olarak;

$$G(i) = \sqrt{\frac{1}{\sum_j tf_{ij}^2}} \tag{5.23}$$

ile gösterilir.

- **Ters Doküman Frekansı (TDF) Ağırlıklandırma**

Bu doküman frekansı en çok kullanılan ağırlıklandırma yöntemlerinden biridir. Bu yöntemde her bir terimin genel ağırlığı, doküman yığınındaki doküman sayısının bu terimin geçtiği doküman sayısına olan oranının logaritması alınarak hesaplanır. Formül ile ifade edilirse,

$$G(i) = \log\left(\frac{N}{df_i}\right) \tag{5.24}$$

ile gösterilir. Formülde yer alan N toplam doküman sayısını ifade ederken df_i ise i inci terimin geçtiği toplam doküman sayısını ifade etmektedir.

- **GFIDF Ağırlıklandırma**

Bu yöntemde her bir terimin genel ağırlığı doküman yığınındaki toplam geçme sayısının bulunduğu doküman sayısına oranı ile hesaplanır ve

$$G(i) = \frac{gf_i}{df_i} \tag{5.25}$$

ile ifade edilir. Burada yer alan gf_i, i. terimin tüm dokümanlardaki toplam geçme sayısını ve df_i de i. terimin geçtiği toplam doküman sayısını ifade eder.

- **Entropi Ağırlıklandırma**

Terimlerin bulundukları dokümandaki olasılığını da dikkate alan bu yöntemde, terimin yer aldığı dokümandaki frekansının doküman yığınındaki tüm dokümanlardaki frekanslarının toplamına oranı ile terimin bulunduğu dokümandaki yer alma olasılığı elde edilir. Terimin bütün dokümanlarda yer alma olasılıklarının toplamı ve kendisinin logaritması ile çarpılır sonuç toplam doküman sayısının logaritmasına bölünür. Elde edilen sonuca bir eklenmesi ile terimin genel ağırlığı elde edilmiş olur. Formül olarak açıklamak istersek ile terimin yer aldığı dokümandaki frekansının (tf_{ij}) tüm dokümanlardaki frekanslarının toplamına (gf_{ij}) oranı ile

$$p_{ij} = \frac{tf_{ij}}{gf_i} \tag{5.26}$$

elde edilir ve elde edilen i. terimin bulunduğu dokümandaki yer alma olasılığı (p_{ij}) aşağıdaki formülde kullanılmak üzere

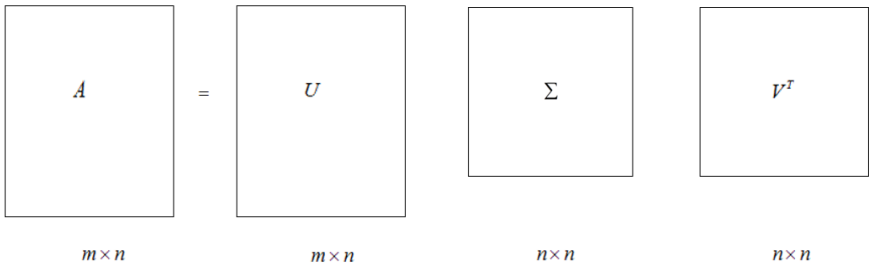
$$G(i) = 1 + \sum_{j}^{N} \frac{p_{ij} \log_2 (p_{ij})}{\log_2 (N)} \tag{5.27}$$

biçimiyle gösterilir.

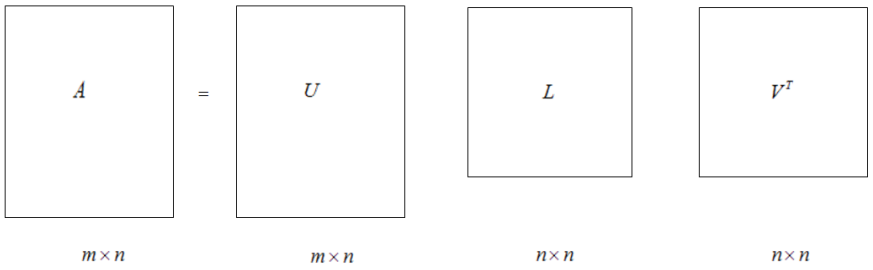
Bu çalışmada kullanılan veri setlerinin tamamında verdiği sonuçların daha iyi olması sebebi ile yerel ağırlıklandırma yöntemi olarak tf , genel ağırlıklandırma yöntemi olarak ise tdf yöntemi kullanılarak terim doküman matrisi elde edilmiştir.

5.3.3. Terim-Doküman Matrisine Matrisi Ayrışımının Uygulanması

GAD'da, Terim doküman matrisinin, $m \times n$ boyutlu A matrisi, elde edilmesinden sonraki aşama bu matrise matris ayrışımının uygulanmasıdır. Bu çalışmada önerilen matris ayrışımı yöntemi olarak kesik ULV ayrışımının yanında performans kıyası yapabilmek için TDA ayrışımı ile de modelleme gerçekleştirilmiştir. Tekil değer ayrışımı için bölüm 5.2.1. de de anlatıldığı gibi $U \in \mathbb{R}^{m \times n}$ ve $V \in \mathbb{R}^{n \times n}$ ortogonal matrislerdir. $\sum \in \mathbb{R}^{n \times n}$ ise A'nın tekil değerlerini köşegen elemanlarında bulunduran bir köşegen matristir. Kesik ULV ayrışımı ile yapılan modelde ise bölüm 5.2.2. de anlatıldığı gibi $U \in \mathbb{R}^{m \times n}$ ve $V \in \mathbb{R}^{n \times n}$ matrisleri sol ortogonal matrisler, $L \in \mathbb{R}^{n \times n}$ ise tekil olmayan alt üçgensel matristir. Şekil 5.3 ve Şekil 5.4 Tekil Değer Ayrışımı ve Kesik ULV ayrışımı sonucu elde edilen matrislerin sembolik gösterimlerini görebilirsiniz.



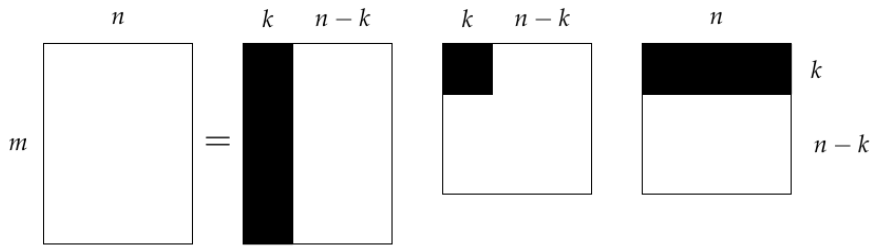
Şekil 5.3. Tekil değer ayrışımının gösterimi



Şekil 5.4. Kesik UlV Ayrışımının gösterimi

5.3.4. Rank k yaklaşımı ve Vektör Uzayının Elde Edilmesi

Veri setinden elde edilen terim doküman matrisi seyrek bir matristir. Bunun nedeni her bir terimin yer aldığı doküman sayısı toplam doküman sayısına göre çok az olmasıdır. Dokümanlardaki tüm terimlerin seçilmesi ya da seçilen terimlerin anlamının dikkate alınmadığı durumlarda bu seyreklik çok daha fazla olmaktadır. Seyrekliğin bu denli fazla olmasının yanında terim sayısının da çok fazla olması işlem maliyetini artırmaktadır. Ayrıca anlamı etkilemeyen ya da gürültü olarak nitelendirilen anlamı olumsuz etkileyen kelimelerin terim doküman matrisinde temsil edilmesi de yanlış çıkarımların yapılmasına sebep olmaktadır. Bu amaçla $A \in \mathbb{R}^{m \times n}$ matrisinde $m > n$ (Şekil 5.5'te olduğu gibi) olarak düşünüldüğünde (genellikle böyledir ancak arama motorları gibi çok fazla doküman yığınının olduğu durumlarda terim sayısının artışı bir noktada duracağı için $m < n$ olur) A matrisi çarpanlarına ayrıldıktan sonra elde edilen çarpanlarına $m > k > n$ şartını sağlayan k değerine göre rank k yaklaşımı adı verilen boyut indirgeme işlemi yapılır. Burada boyut n'den k'ye indirgenerek işlem maliyeti büyük oranda azalmıştır. Aynı zamanda performansı olumsuz etkileyebilecek gürültü ve etkisiz veriler hesaba dâhil edilmemektedir. Böylece daha verimli performans elde edilir. Fakat rank-k yaklaşımındaki k değerini belirlemek, genellikle tekil değerler arasındaki farkın önemli derecede değişmediği için kolay bir işlem olarak değerlendirilmemektedir [58].



Şekil 5.5. Rank k yaklaşımı

Bu çalışmada TDA ve Kesik ULV yöntemine göre iki farklı vektör uzayı elde edilmektedir. TDA yöntemine göre vektör uzayının elde edilmesi, rank k yaklaşımının uygulanması sonucunda elde edilen U_k, $\sum_k$ ve V_k^T matrislerinin

$$A_k = U_k \sum_k V_k^T \tag{5.28}$$

denklemi ile k boyutlu vektör uzayı elde edilir. Terimlerin ve dokümanların vektör uzayındaki temsilcileri ise sırası ile T_k ve D_k olarak

$$T_k = U_k \sum_k \tag{5.29}$$

$$D_k = \sum_k V_k^T \tag{5.30}$$

belirlenir. Elde edilen T_k matrisinin $i.$ satırı $i.$ terimi simgeleyen vektör ve D_k matrisinin $j.$ sütunu $j.$ dokümanı simgeleyen vektördür.

Kesik ULV yöntemine göre vektör uzayının elde edilmesi ise

$$A = ULV^T + E \tag{5.31}$$

denkleminde yer alan U_k, ve V_k^T matrislerine rank k yaklaşımı uygulanır ve U_k, ve V_k^T matrisleri elde edilir. Ancak L matrisinin tekil olmayan alt üçgensel matris olması sebebiyle rank k yaklaşımı yapıldıktan sonra Tekil değerlerini elde etmek için

$$L_k = X_k S_k Y_k \tag{5.32}$$

denklemindeki gibi tekil değer ayrışımı uygulanır. Bu aşamada $k \ll \min(m,n)$ olduğu için tekil değer ayrışımının maliyeti çok küçüktür. Elde edilen denklem, (5.33)'deki yerine rank k yaklaşımına göre yazılırsa

$$A_k = U_k X_k S_k Y_k V_k^T \tag{5.34}$$

k boyutlu vektör uzayını elde ederiz. Elde edilen matrislerden S_k matrisi tekil değer matrisi olmak üzere $\bar{U}_k = U_k X_k$ ve $\bar{V}_k^T = Y_k V_k^T$ olarak ele alınırsa

$$A_k = \bar{U}_k S_k \bar{V}_k^T \tag{5.35}$$

denklemi elde edilir. Terimlerin ve dokümanların vektör uzayındaki temsilcileri ise sırası ile T_k ve D_k olarak

$$T_k = \bar{U}_k S_k \tag{5.36}$$

$$D_k = S_k \bar{V}_k^T \tag{5.37}$$

belirlenir. Elde edilen T_k matrisinin $i.$ satırı $i.$ terimi simgeleyen vektör ve D_k matrisinin $j.$ sütunu $j.$ dokümanı simgeleyen vektördür.

5.3.5. Sorgulama

GAD'da vektör uzayının elde edilmesinden sonra doküman yığını içerisindeki verilecek her hangi bir sorgu cümlesi ya da sorgu metni ile ilgili olan doküman listesine erişimin gerçekleştiği süreçte ilk olarak sorguya ait vektör ve bu vektörün vektör uzayındaki konumu belirlenir. Sorgu metnine ait $m \times 1$ boyutunda olan ve indisleri terim doküman matrisi oluştururken kullanılan ağırlıklandırma yöntemine göre hesaplanarak elde edilen q vektörü, doküman yığının TDA yöntemine göre elde edilen vektör uzayı için

$$Q_{SVD} = q^T U_k \sum_k^{-1} \tag{5.38}$$

formülündeki gibi , q vektörü doküman yığının Kesik ULV yöntemine göre elde edilen vektör uzayında ise

$$Q_{TULVD} = q^T \bar{U_k} S_k^{-1} \tag{5.39}$$

formülü kullanılarak Q ile temsil edilir.

Sorgu metnine ait konumun belirlenmesinden sonra ise bu konuma yakın ya da benzer olan dokümanlar listelenir. Bu aşamada sorguya ait vektör ve diğer dokümanların arasındaki benzerlik hesaplamaları genellikle kosinüs teoremine dayanan kosinüs benzerliği yöntemiyle gerçekleştirilmektedir. Kosinüs benzerliği yöntemi aynı vektör uzayındaki iki vektörün birbirileriyle kesiştikleri açılarının kosinüs değerini dikkate alır. İki vektör arasındaki yakınlığı ya da diğer bir deyişle benzerliği

$$\text{Cos\_}Similarity(X,Y) = \frac{X \cdot Y}{\|X\|\|Y\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5.40}$$

formülü ile hesaplanır. Formülde yer alan X ve Y $m \times 1$ boyutlu olan vektörleri temsil etmektedir. Sorgu vektörü ile doküman vektörü arasındaki benzerliklerin hesaplandığı ve -1 ve 1 arasında değer üreten bu formül sonucunda kosinüs benzerlik değeri ne kadar 1'e yakınlaşırsa sorgu vektörüne benzerliği o kadar çok olmaktadır. Kosinüs benzerlik değeri sorgu vektörüne en yakın olandan en az olana doğru sıralanan doküman vektörlerine ait dokümanlar sorgu metnine karşılık gelen en doğru doküman listesinin bulunmasını sağlar. Kosinüs benzerliği belirli bir eşiğin altında olanları bu listeye dâhil edilmemesi de sorgu sonrasında listelenen doküman listesindeki sorguyla ilişkili doküman oranının artmasını sağlamaktadır.

5.3.6. Performans Değerlendirme

Yapılan bilgi çıkarımı işlemlerinin değerlendirilmesi amacıyla sorgulama sonucunda sistemin sorguyla ilişkili dokümanları geri döndürüp döndürmediği irdelenir. Bunun için ise hassasiyet (precision) ve anma(recall) ölçümleri kullanılmaktadır. Burada kesinlik kavramı p ile gösterilir ve sorgu sonucunda elde edilen ve sorgu ile ilişkili dokümanların sayısının listelenen tüm dokümanların sayısına oranıyla hesaplanır. r ile gösterilen anma (recall) değeri ise sorgu sonucu elde edilen ve sorgu ile ilişkili olan doküman sayısının tüm dokümanlar içerisinde yer alan ve sorgu ile ilişkili olan doküman sayısına oranıyla elde edilmektedir.

$$p = \frac{\text{listelenen ilişkili doküman sayısı}}{\text{listelenen toplam doküman sayısı}} \tag{5.41}$$

$$r = \frac{\text{listelenen ilişkili doküman sayısı}}{\text{toplam ilişkili doküman sayısı}} \tag{5.42}$$

Performansı en çok etkileyen etmenlerden birisi de benzerlik eşik değeridir. Benzerlik eşik değeri arttıkça listelenen doküman listesi azalmakta ve eşik değerinin yüksek olmasından dolayı sorgu metniyle daha ilişkili doküman listesi dönmektedir. Benzerlik eşik değeri azaldıkça hem sorgu sonucunda listelenen doküman listesi artarken hem de listeleme süreci zaman almaktadır. Kosinüs eşik değerinin

belirlenmesi ise *rank k* yaklaşımındaki k değerine göre belirlenmektedir. k değeri ise Terim doküman matrisi olarak adlandırdığımız A matrisinin tekil değerleri olan aşağıdaki

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > \delta \geq \sigma_{k+1} \geq \cdots = \sigma_n \tag{5.43}$$

$k.$ ve $(k+1).$ tekil değerlerini gösteren σ_k ile σ_{k+1} değerleri arasındaki farkın önemli bir şekilde değişmesine bağlıdır. Ancak bu tekil değerler arasındaki bu fark değişiminin çok az farkla gerçekleşmesi sonucu bu değişim noktasının tespiti zor bir işlemdir. k değeri arttıkça kosinüs eşik değeri azalırken k değeri azaldıkça eşik değeri artmaktadır. Ayrıca k değerinin n değerine yaklaşması bir noktaya kadar performansı olumlu etkilerken bu noktadan sonra gürültü ve etkisiz değerlerin işleme dâhil olması sonucu değişimin azalması ya da sabit kalması beklenir. k değerinin en optimum olarak belirlenmesi için σ_k ile σ_{k+1} arasındaki farkın önemli şekilde değişmesi ve sorgu sonrasında dönen doküman listesinin hassasiyet ve anma değerlerinin yüksek olması gerekmektedir.

5.4. Vektör Uzayının Güncellenmesi

Doküman yığınına yeni dokümanların dâhil olması sonucunda vektör uzayının da güncellenmesi gerekmektedir. $X \in \mathbb{R}^{n \times p}$ matrisi doküman yığınına eklenen yeni doküman listesinin önceden kullanılan ağırlıklandırma yöntemine göre elde edilen matrisi olmak üzere A matrisinin eklenmesi sonucu güncelleceği yeni matris olan $\overline{A} \in \mathbb{R}^{m \times p}$ matrisi

$$\overline{A} = \begin{pmatrix} A \\ X^T \end{pmatrix} \tag{5.44}$$

ile gösterilir. Bu durumda vektör uzayının güncellenmesi için elde edilen bu matrisin ULV ayrışımındaki çarpanlarının da güncellenmesi gerekmektedir. Bu durumda matris ayrışım hesaplamalarının yeniden yapılması yerine mevcut yapının

güncellenmesi önerilir. Bu amaçla bloklar halinde eklenen dokümanların kesik ULV ayrışımının blok güncellenmesi gerçekleştirilmiştir. Blok güncelleme algoritmasında kullanılan **local_QR** fonksiyonu bir matrisin ortogonal çarpanlarına ayrışımı sağlayan QR ayrışımını gerçekleştirir. Girdi olarak $p \le n \le m$ şartını sağlayan $\overline{A} \in \mathbb{R}^{m \times p}$ matrisini alan **local_QR** fonksiyonu ile $\overline{Q} \in \mathbb{R}^{m \times p}$ soldan ortogonal matrisi ve $\overline{R} \in \mathbb{R}^{p \times p}$ üst üçgensel matrisi elde edilir. $\overline{Q}$ ve $\overline{R}$ matrisleri aşağıdaki

$$\left\| I_p - \overline{Q}^T \overline{Q} \right\| \le \varepsilon_M \Omega(m, p) \ll 1 \tag{5.45}$$

$$\overline{A} + \Delta \overline{A} = \overline{Q}\,\overline{R} \ , \ \left\| \Delta \overline{X} \right\| \le \varepsilon_M \Omega(m, p) \left\| \overline{X} \right\| \tag{5.46}$$

denklemlerini sağlamaktadır. Denklemde yer alan $\Omega(m, p)$ çok küçük değerli bir ε_M makine sayısıdır. local_qr fonksiyonu Householder QR dönüşümü ya da Givens QR dönüşümü yöntemlerinden herhangi biri ile modellenebilir. Householder QR ayrışımının hata analizi [59]'de belirtildiği gibi $\Omega(m, p) = cmp^{3/2}$ sonucuna göre c bir sabitdir.

QR ayrışımında kullanılan temel algoritmalardan biri Gram-Schmidt algoritmasıdır. Bu çalışmada iki aşamalı BCGS [60] kullanılmıştır. Bu algoritmada daha önce tanımladığımız **local_QR** fonksiyonu ile girdi matrisinin QR ayrışımı hesaplanmaktadır.

Algoritma 1 BSGS Algoritması

```
function [Q, R, S] = BCGS(U, X)
% Girdi:
    % U sol ortogonal matris
    % X yeni eklenen dikdrtgen matris
% Çıktı:
    % Q sol ortogonal matris
    % R üst üçgensel matris
S = U^T B;
Ā = B − US;
[Q, R] = local_QR(Ā);
end BCGS
```

BCGS algoritmasında A matrisinin Kesik ULV sonrasında elde edilen U matrisiyle birlikte yeni eklenen veri yığınının temsilcisi olan X matrisi girdi değerleridir. Çıktı değerleri olarak ise $Q \in \mathbb{R}^{m \times p}$ yaklaşık sol ortogonal matrisi, $R \in \mathbb{R}^{p \times p}$ üst üçgensel matrisi ve bir dikdörtgen matris olan $S \in \mathbb{R}^{q \times p}$ matrisi elde edilir. Ayrıca Q ve X matrisleri (5.47)'de verilen şartları sağlarlar. İki aşamalı BCGS (BCGS2) algoritmasında girdi matrisi birinci BCGS yönteminin çıktıları daha sonra yine bir BCGS algoritmasının girdisi olarak işlenmekte ve ikinci çıktılar QR ayrışımını vermektedir. BCGS algoritması ve BCGS2 algoritması sırası ile Algoritma 1 ve Algoritma 2'de görülebilir.

Algoritma 2 İki aşamalı BSGS Algoritması

```
function [Q_B, R_B, S_B] = BCGS2(U, B)
% Girdi:
    % U sol ortogonal matris
    % B yeni eklenen dikdrtgen matris
% Çıktı:
    % Q_B sol ortogonal matris
    % R_B üst üçgensel matris
[Q_1, R_1, S_1] = BCGS(U, B);
[Q_B, R_2, S_2] = BCGS(U, Q_1);
S_B = S_1 + S_2 R_1;
R_B = R_2 R_1
end BCGS2
```

5.4.1. Kesik ULV Blok Güncelleme Algoritması

Terim-Doküman matrisi olan ve Kesik ULV uygulanarak vektör uzayının elde edildiği $A \in \mathbb{R}^{m \times n}$ matrisine eklenecek $X \in \mathbb{R}^{n \times p}$ matrisi sonrasında elde edilen $\overline{A} \in \mathbb{R}^{m \times p}$ matrisi, $k \ll n$, $m \gg n$ ve $p < n$ şartları altında

$$\begin{aligned} \overline{A} &= \begin{pmatrix} A \\ X^T \end{pmatrix} \\ \overline{A} &= \begin{pmatrix} U_1 L_1 V^T + E \\ X^T \end{pmatrix} \\ \overline{A} &= \begin{pmatrix} U_1 L_1 V^T \\ X^T \end{pmatrix} + \begin{pmatrix} E \\ 0 \end{pmatrix} \end{aligned} \tag{5.48}$$

biçiminde ifade edilir.

X matrisinin ve Kesik ULV sonucunda elde edilen V_1 matrisinin girdi olarak işlendiği BCGS2 algoritmasının sonucunda V_{new}, L_{new} ve S_{new} matrisleri

$$\left[V_{new}, L_{new}, S_{new}\right] = BCGS2(V_1, X) \tag{5.49}$$

biçiminde elde edilir ve

$$X^T = S_{new}^T V_1^T + L_{new} V_{new}^T \tag{5.50}$$

sonucuna ulaşılır. Ayrıca

$$\overline{A} = \begin{pmatrix} U_1 & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} L & 0 \\ S_{new}^T & L_{new} \end{pmatrix} \begin{pmatrix} V_1^T \\ V_{new}^T \end{pmatrix} \begin{pmatrix} E \\ 0 \end{pmatrix} \tag{5.51}$$

sonucu ve

$$\overline{U}_1 = \begin{pmatrix} U_1 & 0 \\ 0 & I_p \end{pmatrix}, \overline{L} = \begin{pmatrix} L & 0 \\ S_{new}^T & L_{new} \end{pmatrix}, \overline{V}_1 = \left(V_1, V_{new}\right) \text{ ve } \overline{E} \begin{pmatrix} E \\ 0 \end{pmatrix} \tag{5.52}$$

eşitlikleri elde edilir. Elde edilen bu eşitliklerle de güncellenmiş vektör uzayını temsil eden

$$\overline{A} = \overline{U}_1 \overline{L} \overline{V}_1^T + \overline{E} \tag{5.53}$$

matrisi elde edilir. Yeni eklenen matris bloğundan dolayı elde edilen yeni $\overline{L}$ matrisine, oluşabilecek hataların önüne geçilmesi için refinement algoritması uygulanır [61]. Kesik ULV blok güncelleme algoritması Algoritma 3'te görülebilir.

Algoritma 3 Kesik ULV Blok Güncelleme Algoritması

```
function [Ū1, L̄, V̄1] = truncated_ULV_block_update(A, U1, L, V1, X)
% Input:
    % A veri matrisi
    % U1 yaklaşık sol ortomonal matris
    % V1 yaklaşık sol ortomonal matris
    % L alt uçgensel matris
    % X eklenecek yeni veri matrisi
% Output:
    % Ū1 yaklaşık sol ortomonal matris
    % V̄1 yaklaşık sol ortomonal matris
    % L̄ alt uçgensel matris
Ā = [A; X^T];
[V_new, L_new^T, S_new] = T_BCGS(V1, X);
Ũ1 = [U1 0; 0 I]; L̃ = [L 0; S_new^T L_new]; Ṽ1 = [V1 V_new]; norm_Ẽ = norm_E; % Ẽ = [E; 0]
[Ū1, L̄, V̄1] = refinement(Ā, Ũ1, L̃, Ṽ1, norm_Ẽ);
end truncated_ULV_block_update
```

5.4.2. Kesik ULV Blok Güncelleme Örnekleri

Önceden Kesik ULV işlemine tabi tutulmuş bir matrise eklenecek yeni bir blok matris olduğunda yapılan blok güncelleme algoritmalarında matrisin ilk durumu yeni elde edilecek matrisin şekillenmesinde çok etkilidir. Blok güncelleme algoritmalarında bu nedenle bu eski matrisin elde edilecek yeni matrise olan etkisini azaltmak için eski matris 0 ve 1 arasındaki bir değer alan unutma faktörü ile çarpılarak işleme alınır. Yapılan bu işlem üstel pencereleme yöntemi olarak adlandırılır [62]. Böylece $A \in \mathbb{R}^{m \times n}$ matrisi α ile simgelenen ve 0 ve 1 arasında değer alan unutma faktörü ile çarpılır ve elde edilen bu matrise eklenen $X \in \mathbb{R}^{n \times p}$ matrisiyle t zamanlı bir

$$A(t+1) = \begin{pmatrix} aA(t) \\ X^T(t) \end{pmatrix} \tag{5.54}$$

matrisi elde edilir. Veri matrisi olarak ele aldığımız X_{data} matrisi, $m \times n$ boyutlu ve elemanları (0-1) aralığında tekdüze bir dağılımdan rastgele seçilmiş bir matristir.

Başlangıç aşaması $t = 0$'da X_{data} matrisinin ilk m satırını içeren $X(0) \in \mathbb{R}^{m \times n}$ veri matrisinin ayrışımı [30] deki UTV yazılımı içindeki lulv isimli MATLAB fonksiyonu kullanılarak elde edilir.

Daha sonra $t > 0$ olduğu aşamada $p = 2, u < n$ boyutlu veri bloğuna

$$\eta < \varphi < \omega \text{ şartını sağlayan } M = 2^{\omega}, m = 2^{\varphi}, n = 2^{\eta} \tag{5.55}$$

$$A(t) = X_{data}^{T}(m + p*(t-1) : m + p*t, :) \tag{5.56}$$

üstel pencereleme işlemi uygulanır. Üstel pencereleme işlemi boyunca uygulanan adım sayısı v olmak üzere

$$v = 2^{\varphi-\mu}(2^{\omega-\varphi} - 1) \tag{5.57}$$

biçimindedir. Uygulanan her bir v adımında $U_1(t)$ ve $V_1(t)$ matrislerinin sol ortogonalliği sırasıyla

$$\left\| I - U_1^T(t) U_1(t) \right\|_F \tag{5.58}$$

ve

$$\left\| I - V_1^T(t) V_1(t) \right\|_F \tag{5.59}$$

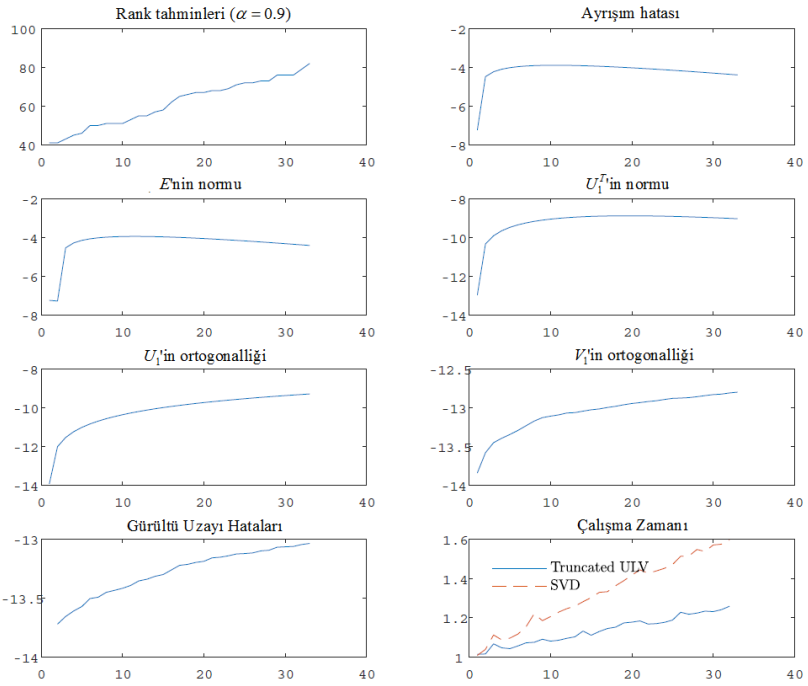
denklemleri kullanılarak kontrol edilir. Ayrışım hatası

$$\left\| E(t) \right\|_F = \left\| X(t) - U_1(t) L(t) V_1^T(t) \right\|_F \tag{5.60}$$

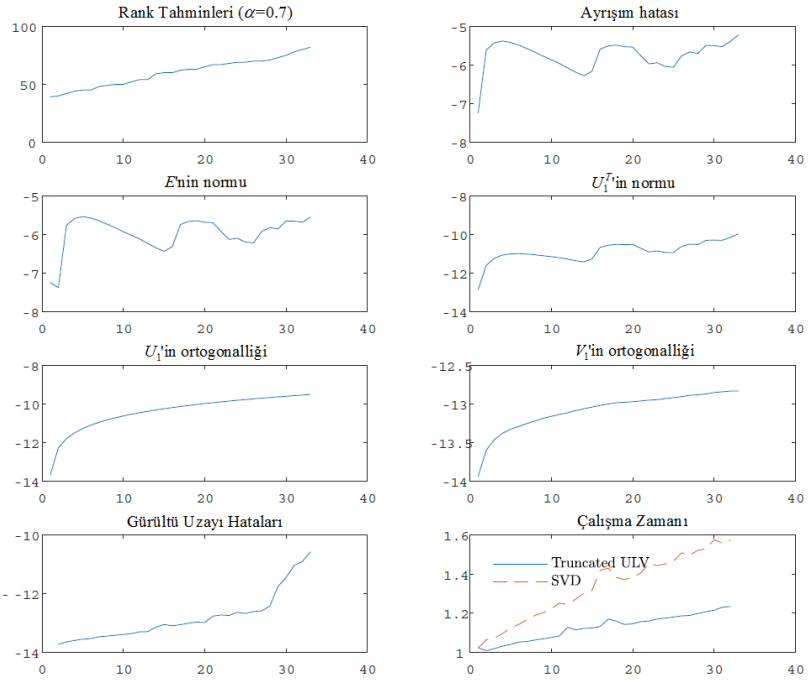
denklemi ile ölçülmüş ve sonuçları $log10$ tabanına alınarak gösterilmiştir. Ayrıca her bir t adımında $X(t)$ matrisinin sayısal rankı $k(t)$ incelenmiştir.
Diğer yandan her t adımındaki rank doğruluğu $X(t)$ matrisinin TDA ayrışımındaki sonuçları referans alınarak kontrol edilmiştir.

Bunlara ek olarak her bir t blok adımında, [63]'de verilen TDA blok güncelleme algoritması ile Kesik ULV blok güncelleme algoritması hız açısından karşılaştırılmış ve sonuçları $log10$ tabanında alınarak irdelenmiştir.

Örnek 1: X_{data} matrisi için w=14,vq=13, n=9 ve u=8, xdata'nın rastgele seçilen satır oranı r=[%95M], n=10-9 ile çarpılsın. Rank toleransı e=10-8 ve unutma faktörü a=0.9.



Şekil 5.6. Örnek 1'e göre sayısal sonuçlar



Şekil 5.7. Örnek 2'ye göre sayısal sonuçlar

6. ARAŞTIRMA BULGULARI

Bu bölümde GAD sürecinde kullanılacak alternatif matris ayrışımı olarak önerdiğimiz Kesik ULV modelini kıyaslamak ve incelemek amacıyla TDA modeli de oluşturulmuştur. Elde edilen her iki model içerisinde hem İngilizce hem de Türkçe veri setleri için dizinleme işlemleri gerçekleştirilmiştir. Veri seti olarak bilgiye erişim çalışmalarında yaygın olarak kullanılan Amerikan Dokümantasyon Enstitüsü Raporları (ADI), Time dergisinde yayınlanan makale koleksiyonu (TIME) ve Medline makalelerinden oluşan koleksiyon (MED) gibi veri setlerinin yanında Türkçe veri seti olarak Türkçe haber sayfalarına ait veri setleri (TRNEWS) kullanılmıştır. Her bir veri seti için terim doküman matrisinin oluşturulmasından önce dokümanlarda yer alan durak kelimelerin temizlenmesi ve kullanılacak her bir kelime için gövdeleme (stemming) işlemi gerçekleştirilmiştir. Bu işlemlerin gerçekleşmesinin ardından kullanılan veri setlerine dair detaylı bilgiler Çizelge 6.1'de sunulmaktadır.

Çizelge 6.1. Veri setleri

Veri Seti	**Doküman Sayısı**	**Terim Sayısı**	**Sorgu Sayısı**
ADI	82	986	35
TIME	424	14774	83
MED	1033	9477	30
TRNEWS	7500	11675	6

Çizelge 6.1'de verilen veri setlerinin her birinde sorgu cümleciklerine karşılık gelen ilişkili doküman listeleri de yer almaktadır. Böylece geliştirilen metodun başarım oranı bu listeler vasıtası ile gerçekleştirilmiştir.

TRNEWS veri seti ise 5 adet Türkçe haber sitesindeki içeriklerinin, çalışma sürecinde geliştirilen ve arama motorlarında kullanılan bot benzeri bir yazılımla elde

edilmiştir. Her bir web sayfasındaki HTML kodları içerisindeki yapılandırılmamış metinler web madenciliğindeki ön işlem sürecinden geçirilerek elde edilmiştir. Daha sonra her biri yapılandırılmamış veri olan haber metinleri, metin madenciliği yöntemindeki ön işlem süreçlerinden geçirilmiştir. Elde edilen haber metinlerindeki terim ve bu terimlerin ilgili dokümandaki sıklık bilgisi veri tabanına kaydedilmektedir. Yapılan bu işlem süreci, yani bir sayfadaki hedef metnin elde edilmesindeki ön işlem süreci ve elde edilen metine dair terim ve sıklık bilgilerinin veri tabanına kaydedilme işlemi bir defaya mahsus yapılmaktadır.

Diğer bir şekilde açıklanırsa, web sayfasındaki içeriğin tamamının veri tabanına kaydedilmesi yerine sadece sayısallaştırılmış değerleri yani terim doküman matrisinde kullanılacak frekans bilgileri, terim bilgisi (terimin ilgili dokümanda yer alıyor olması bilgisi) kaydedilmektedir. Bunun yanında sayfadaki az yer kaplayan meta bilgiler de yardımcı bilgiler olarak kaydedilmektedir. Ancak geliştirilen algoritmanın başarısını doğru teyit etmek için örneklerdeki doküman listeleme sürecinde meta bilgilerden faydalanılmamıştır. Böylece hem veri tabanında yoğunluk olmazken hem de kullanılan veri büyüklüğünün az olması performansı olumlu yansımaktadır.

Web sayfalarındaki metinlerin ve bu metinlerde yer alan terimlerin sıklık bilgileri veri tabanına kaydedilme süreci geliştirilen bot yazılımının tasarımı gereği sürekli devam eden ve web sayfasında yayınlanan yeni içerikler oldukça veri tabanına bu web sayfalarını da ekleyen bir süreçtir. Web sayfasındaki içeriklerin veri tabanına kaydedilmesi sürecinde bu sayfada geçen diğer web sayfa linkleri de birer potansiyel veri kümesi olarak ele alınmaktadır. Dolayısıyla sürekli artan bir terim ve doküman listesi ile karşı karşıya kalınmaktadır. Geliştirilen çalışmada terim doküman matrisinin oluşması sürecinde, ilk andaki son terim ve son doküman bilgileri dikkate alınmaktadır. Sürecin ilk anından sonra veri tabanına kaydedilen terim ve dokümanlar terim doküman matrisine etkisi olmamaktadır. Böylece veri tabanından alınan her bir terimin ilgili dokümandaki geçme sıklığı ve tüm doküman yığınındaki bulunma sayısı hesaplanarak TF-TDF ağırlıklandırma metoduna göre Terim-Doküman matrisi oluşturulmaktadır. Daha sonra elde edilen bu matris ile hem TDA hemde Kesik ULV modeline göre iki ayrı vektör uzayı elde edilmektedir.

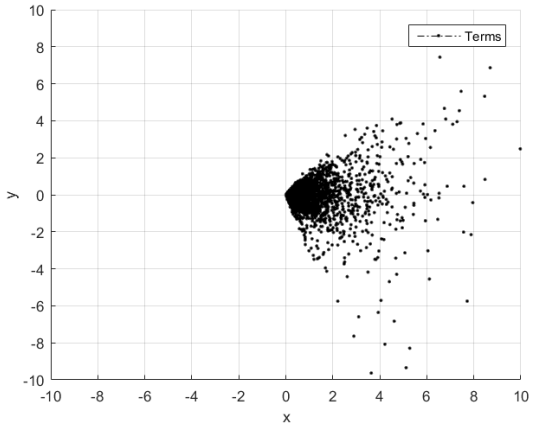
Her bir web sayfasındaki Hyper text markup language (HTML) kodları içerisindeki yapılandırılmamış metinler web madenciliğindeki ön işlem sürecinden geçirilerek elde edilmiştir. Daha sonra her biri yapılandırılmamış veri olan haber metinleri, metin madenciliği yöntemindeki ön işlem süreçlerinden geçirilmiştir. Geliştirilen yazılım ile web sayfalarındaki veri setlerine dair kelime ve sıklık bilgilerinin veri tabanına kaydedilmesinden sonra haber metinlerinde yer alan her bir kelimenin terim, her bir haber metninin doküman olarak isimlendirildiği terim-doküman matrisi elde edilmektedir. Yapılan her iki yöntemin başarısını test etmek amacıyla Türkçe haber yığını için sorgu cümlecikleriyle ilişkili doküman listesi oluşturulmuştur. Bu liste oluşturulurken sorguyla yakın ya da uzak ilişkide olabilecek bütün haber metinleri sorguyla ilişkili doküman listesine eklenmiştir. Çizelge 6.2'de TRNEWS veri setinde yer alan sorgular ve bu sorgularla ilişkili doküman listesini görebilirsiniz.

Çizelge 6.2. TRNEWS veri seti için sorgular ve bu sorgularla ilişkili doküman sayısı

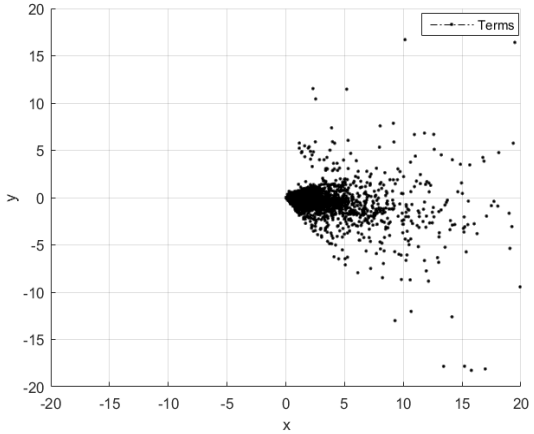
Sorgu	**İlişkili Doküman Sayısı**
Kültür sanat haberleri	157
Hava tahmini ve meteroloji haberleri	54
Sağlıklı beslenme ve diyet haberleri	35
Galatasaray, fenerbahçe, besiktas ve trabzonspor haberleri	135
Facebook, twitter ve instagram gibi sosyal medya sitelerinin haberleri	219
Avrupa birliği (ab) haberleri	40

MED veri setindeki terim ve dokümanlar için her iki yöntemde oluşturulan vektör uzayları k değeri 2 olarak ele alınarak irdelendiğinde; Şekil 6.1'de TDA kullanılarak oluşturulan vektör uzayındaki terim kümesinin dağılımı ve Şekil 6.2'de Kesik ULV

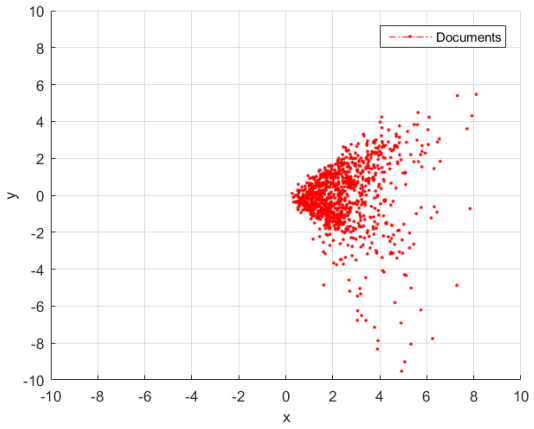
kullanılarak oluşturulan vektör uzayındaki terimlerin dağılımı gösterilmektedir. Aynı şekilde Şekil 6.3'te TDA kullanılarak oluşturulan vektör uzayında dokümanların dağılımını ve Şekil 6.4'te Kesik ULV Kullanılarak oluşturulan vektör uzayındaki dokümanların dağılımı gösterilmektedir. Her iki algoritma için terimlerin ve dokümanların dağılımları incelendiğinde Kesik ULV ayrışımı ile gerçekleşenlerin daha geniş alana dağıldığı görülmektedir. Ancak açısal olarak irdelendiğinde her iki algoritma için verilen dağılımlar birebir aynı olmamakla birlikte benzer dağılım gösterdiği gözlemlenmektedir.



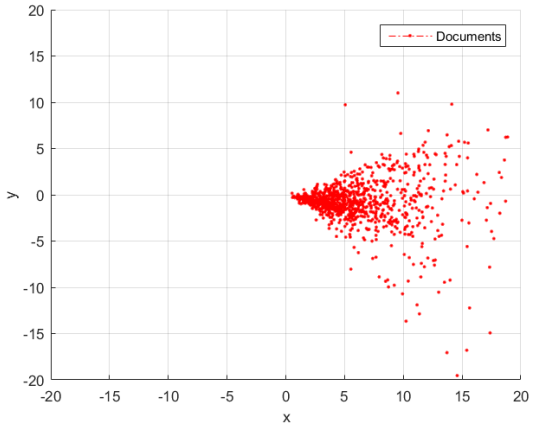
Şekil 6.1 TDA ile elde edilen vektör uzayındaki terimlerin dağılımı (MEDLINE Veri seti için)



Şekil 6.2 Kesik ULV ile elde edilen vektör uzayındaki terimlerin dağılımı (MEDLINE Veri seti için)



Şekil 6.3 TDA ile elde edilen vektör uzayındaki dokümanların dağılımı (MEDLINE Veri seti için)



Şekil 6.4 Kesik ULV ile elde edilen vektör uzayındaki dokümanların dağılımı (MEDLINE Veri seti için)

Çizelge 6.3, Çizelge 6.4, Çizelge 6.5 ve Çizelge 6.6 sırası ile ADI, MED, TIME ve TRNEWS verilerine uygulanan TDA ve Kesik ULV ayrışımı yöntemlerinin farklı k değerlerine göre sonuçlarını göstermektedir. Bu tablolarda yer alan %10 ve %50 değerleri tüm sorguların sonucunda geri dönen doküman listesinin ele alınan yüzdelik dilimlerini belirtmektedir. Precision ise bu dilimlerdeki dizinlenen dokümanların bütün sorgular için ortalama başarısını göstermektedir. Ayrıca Min.

Benzerlik Değeri ile de sorgu sonucunda listelenen dokümanların ortalama minimum benzerlik değerini gösterilmektedir.

Çizelge 6.3. TDA ve Kesik ULV modellerine göre doküman dizinleme başarısı (ADI veri seti için)

k	***TDA***			***Kesik ULV***		
	Min. Benzerlik Değeri	Hassasiyet (10%)	Hassasiyet (50%)	Min. Benzerlik Değeri	Hassasiyet (10%)	Hassasiyet (50%)
5	0.162	2.3	12.6	0.0884	21.3	14.1
10	0.119	31.1	18.6	0.0225	31.5	17.3
20	0.105	39.4	23.8	0.0101	34.7	19.9
30	0.103	55.3	29.0	0.0086	46.6	20.2
40	0.102	58.9	32.8	0.0071	69.3	32.1
50	0.101	73.5	33.1	0.0034	70.2	37.6
60	0.101	71.2	39.7	0.0008	65.3	37.9
70	0.100	69.2	39.3	0.0007	53.8	36.2
82	0.100	50.0	41.2	0.0003	49.3	29.1

Çizelge 6.4. TDA ve Kesik ULV modellerine göre doküman dizinleme başarısı (MED veri seti için)

k	***TDA***			***Kesik ULV***		
	Min. Benzerlik Değeri	Hassasiyet (10%)	Hassasiyet (50%)	Min. Benzerlik Değeri	Hassasiyet (10%)	Hassasiyet (50%)
5	0.618	22.2	12.6	0.786	29.8	11.7
10	0.289	34.2	15.2	0.547	53.0	17.2
20	0.069	73.1	19.3	0.148	66.5	18.8
50	0.035	77.9	19.4	0.091	77.5	24.2
100	0.025	85.2	34.4	0.035	83.2	45.6
150	0.022	85.3	27.1	0.021	85.8	46.0
300	0.015	66.4	18.1	0.010	77.2	38.3
600	0.007	56.2	16.5	0.006	52.0	16.0
900	0.005	51.6	15.3	0.003	47.7	15.1
1033	0.003	49.6	15.5	0.001	46.9	15.2

Çizelge 6.5. TDA ve Kesik ULV modellerine göre doküman dizinleme başarısı (TIME veri seti için)

k	***TDA***			***Kesik ULV***		
	Min. Benzerlik Değeri	Hassasiyet (10%)	Hassasiyet (50%)	Min. Benzerlik Değeri	Hassasiyet (10%)	Hassasiyet (50%)
5	0.6290	30.4	13.2	0.70	31.3	12.4
10	0.4840	49.1	17.0	0.56	44.4	15.5
20	0.2140	58.9	17.4	0.51	49.8	16.1
50	0.1001	59.7	18.5	0.17	59.6	34.7
100	0.1013	70.8	22.8	0.09	65.8	35.2
150	0.1009	72.6	25.1	0.06	73.4	38.3
200	0.1004	75.6	33.1	0.03	74.1	40.8
300	0.1026	73.5	40.9	0.02	63.1	44.8
400	0.0823	72.8	47.1	0.02	63.1	46.1
424	0.0812	68.8	45.1	0.01	56.7	32.4

Çizelge 6.6. TDA ve Kesik ULV modellerine göre doküman dizinleme başarısı (TRNEWS veri seti için)

k	***TDA***			***Kesik ULV***		
	Min. Benzerlik Değeri	Hassasiyet (10%)	Hassasiyet (50%)	Min. Benzerlik Değeri	Hassasiyet (10%)	Hassasiyet (50%)
5	0.790	15.4	6.6	0.981	10.4	4.1
10	0.653	24.9	8.8	0.939	15.5	9.4
25	0.393	38.7	14.7	0.591	18.3	8.5
50	0.171	64.6	26.0	0.402	37.9	13.0
100	0.130	76.7	39.7	0.196	65.1	27.4
150	0.112	78.5	45.7	0.139	75.5	42.2
200	0.099	83.5	49.4	0.124	78.6	44.2
250	0.090	84.1	50.7	0.100	80.2	44.9
300	0.0831	82.8	52.3	0.099	83.2	45.4
400	0.0797	80.1	54.7	0.086	84.6	50.5
500	0.0754	79.3	57.2	0.079	86.1	55.5
750	0.0676	79.4	60.5	0.067	89.4	55.2
1000	0.0642	80.4	64.1	0.061	90.3	57.2
2000	0.0538	86.4	66.5	0.042	90.1	63.2

Çizelge 6.7, Çizelge 6.8, Çizelge 6.9 ve Çizelge 6.10 sırası ile ADI, MED, TIME ve TRNEWS verilerine uygulanan TDA ve Kesik ULV yöntemlerinin benzerlik eşik değerine göre tüm sorgular için ortalama sonuçlarını göstermektedir. Tablolarda yer alan anma, Hassasiyet ve Listelenen Doküman sayısı değerleri geri dönen dokümanların %100'ü dikkate alınarak hesaplanmıştır.

Çizelge 6.7. Benzerlik eşiğine göre başarı (ADI veri seti)

Benzerlik Eşik Değeri	TDA			Kesik ULV		
	Anma	Hassasiyet	Listelenen Döküman Sayısı	Anma	Hassasiyet	Listelenen Döküman Sayısı
0.1	82.1	11.4	34.4	76.7	18.1	35.3
0.2	71.2	17.2	19.7	62.2	23.4	21.3
0.3	60.5	24.7	11.5	49.8	35.4	11.3
0.4	46.7	33.9	6.3	44.9	51.9	6.5
0.5	36.3	47.4	3.6	31.1	60.5	3.7
0.6	32.8	59.6	2.3	18.6	58.7	2.1
0.7	25.8	71.0	2.0	13.0	72.2	1.5
0.8	15.2	73.5	1.3	0.0	0.0	0.0
0.9	0.0	0.0	0.0	0.0	0.0	0.0

Çizelge 6.8. Benzerlik eşiğine göre başarı (MED veri seti)

Benzerlik Eşik Değeri	TDA			Kesik ULV		
	Anma	Hassasiyet	Listelenen Döküman Sayısı	Anma	Hassasiyet	Listelenen Döküman Sayısı
0.1	97.4	21.3	135.7	97.9	17.2	165.8
0.2	94.1	43.1	64.1	94.5	38.0	73.4
0.3	89.7	59.4	41.1	87.1	53.4	44.7
0.4	83.3	70.4	30.2	81.9	66.3	32.4
0.5	66.3	76.8	21.7	68.4	75.1	23.2
0.6	51.7	86.8	15.1	51.8	84.2	15.2
0.7	30.0	91.7	18.4	33.0	90.2	9.3
0.8	18.6	97.1	4.5	20.9	90.7	5.5
0.9	7.9	100.0	1.5	6.1	100	1.3

Çizelge 6.9. Benzerlik eşiğine göre başarı (TIME veri seti)

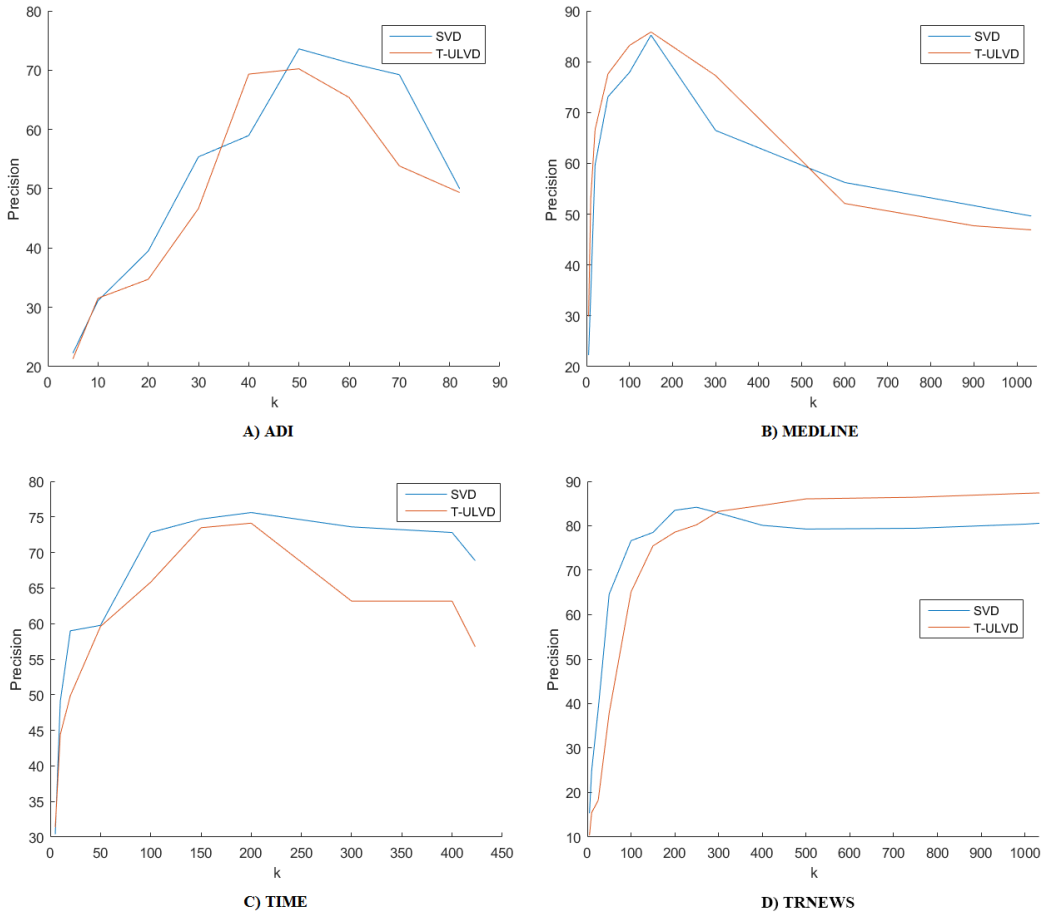
Benzerlik Eşik Değeri	TDA			Kesik ULV		
	Anma	Hassasiyet	Listelenen Döküman Sayısı	Anma	Hassasiyet	Listelenen Döküman Sayısı
0.1	88.6	12.4	58.4	82.5	22.5	37.5
0.2	82.4	17.2	46.6	71.8	34.4	20.2
0.3	73.5	26.9	25.6	59.1	40.1	12.8
0.4	68.3	37.7	16.3	49.9	51.7	8.1
0.5	63.2	49.8	10.8	40.0	56.8	5.5
0.6	54.1	63.2	7.1	32.8	66.3	4.8
0.7	43.8	72.3	4.8	32.7	67.5	3.0
0.8	40.6	76.6	4.0	27.6	71.6	2.5
0.9	0.0	0.0	0.0	0.0	0.0	0.0

Çizelge 6.10. Benzerlik eşiğine göre başarı (TRNEWS veri seti)

Benzerlik Eşik Değeri	TDA			Kesik ULV		
	Anma	Hassasiyet	Listelenen Döküman Sayısı	Anma	Hassasiyet	Listelenen Döküman Sayısı
0.1	84.9	15,3	545.5	83.9	15.0	576.4
0.2	78.1	31.0	232.0	74.7	30.7	305.4
0.3	67.7	45.4	138.8	61.4	48.4	169.0
0.4	58.8	60.5	89.2	49.9	61.1	115.4
0.5	42.9	73.2	53.3	42.3	72.8	77.2
0.6	29.4	86.4	31.2	30.8	83.0	49.6
0.7	23.7	88.5	24.3	24.9	85.6	28.0
0.8	0	0	0	0	0	0
0.9	0	0	0	0	0	0

Şekil 6.5'de sırası ile ADI, MED, TIME ve TRNEWS veri setleri için TDA ve Kesik ULV kullanılarak oluşturulan vektör uzayındaki dokümanların dizinleme başarıları gösterilmektedir. Her bir sorgu için dönen bütün dokümanlar en benzer olanından başlamak üzere benzemeyene doğru listelenmektedir. Bu dokümanların şekilde belirtildiği gibi yüzdelik dilimlerdeki hassasiyetleri (precision) hesaplanmıştır. Yüzdelik dilim oranı arttıkça hassasiyet azalmaktadır ancak ilişkili dokümanlara

erişim artmaktadır. Diğer sonuçlarda da olduğu gibi bu grafiklere de her bir veri setindeki sorguların tamamı için listelenen dokümanların performans ölçümlerinin ortalaması yansıtılmıştır.

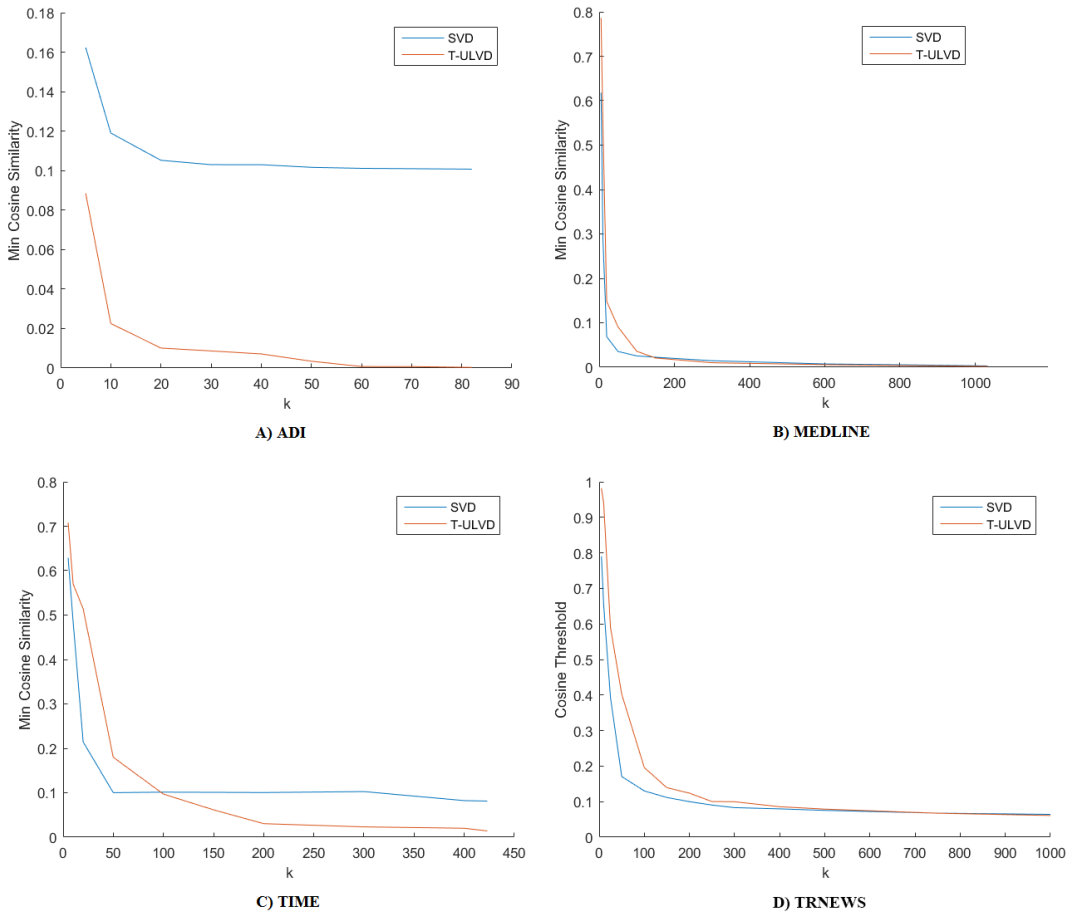


Şekil 6.5 Farklı k değerine göre TDA ve Kesik ULV'ye göre hassasiyet sonuçları

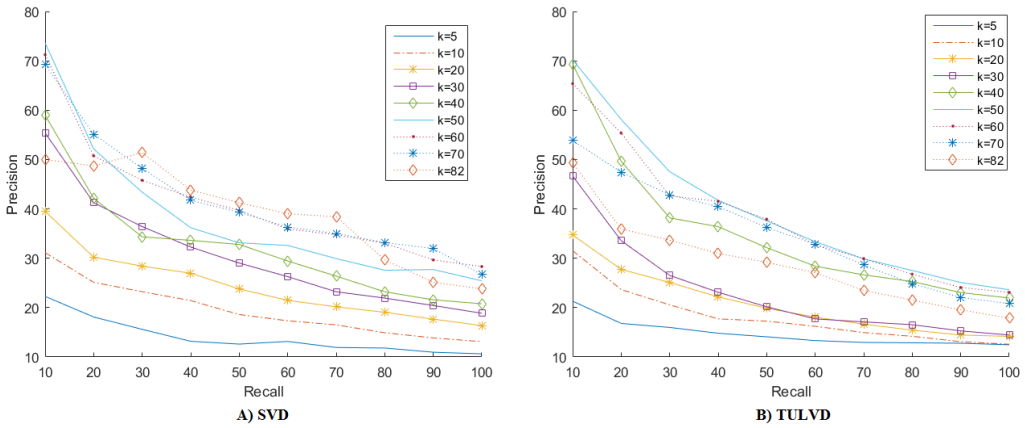
Şekil 6.6 test sürecinde kullanılan üç veri seti için TDA ve Kesik ULV yöntemlerini dizinleme işlemi sonucundaki listelenen dokümanların ortalama minimum benzerlik değerlerini farklı rank değerlerine göre karşılaştırmaktadır. MED ve TIME veri setleri için sonuçlar daha benzer olduğu görülmektedir. ADI veri setinde ise artan rank değerlerine göre farklı değerler almasına karşın benzerlik değişim oranının neredeyse aynı olduğu görülmektedir. Buradaki farklılığın sebebi doküman sayısı en

çok olan MED veri setindeki benzerliğin TIME veri setinden daha iyi olması da dikkate alınarak ADI veri setinin doküman sayısının az olmasına bağlanabilir.

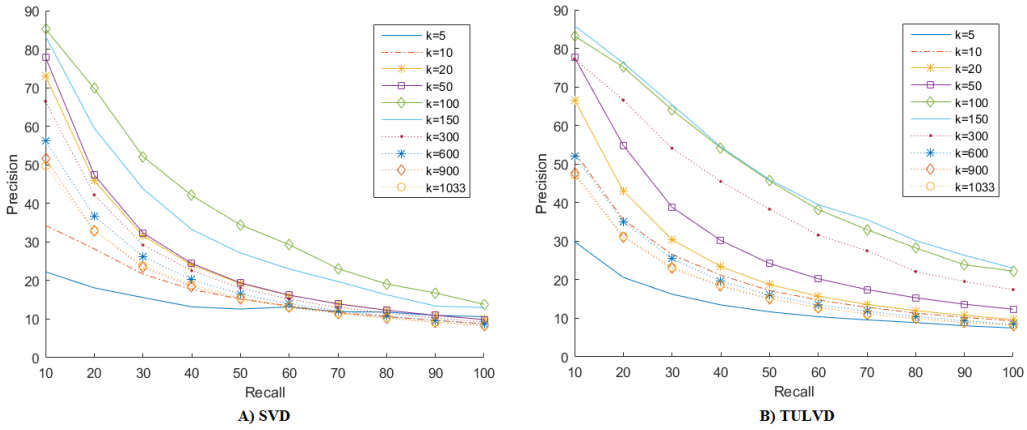
Diğer bir yandan her bir veri seti için farklı k değerine göre ortalama dizinleme başarısı sorgulama sürecinden sonra listelenen dokümanların yüzdelik oranlarına göre ADI, MED, TIME ve TRNEWS veri seti için sırasıyla Şekil 6.7, Şekil 6.8, Şekil 6.9 ve Şekil 6.10'da gösterilmektedir.



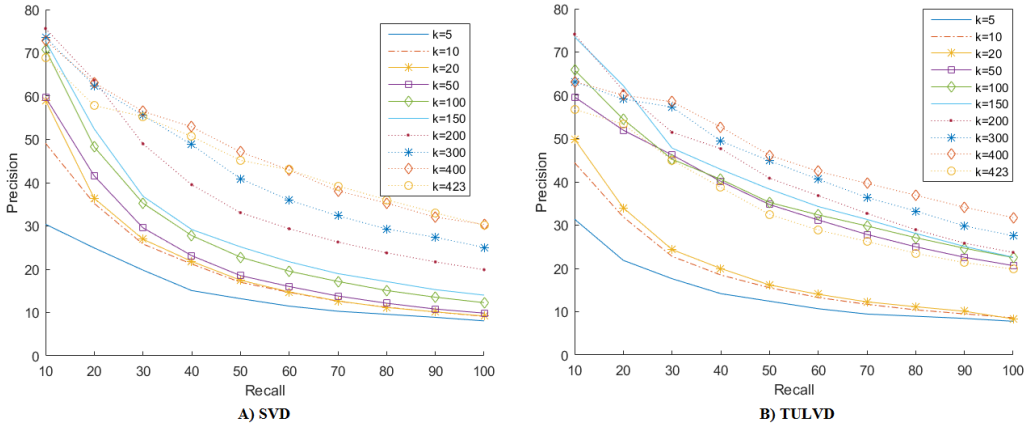
Şekil 6.6 Farklı k değerine göre Minimum Benzerlik Değerinin Değişimi



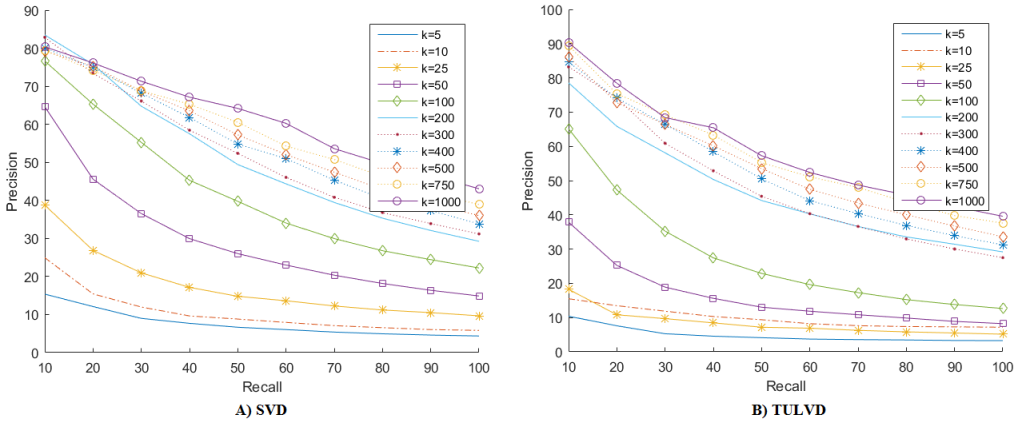
Şekil 6.7 Farklı k değerine göre Hassasiyet ve Anma Sonuçları (ADI veri seti için)



Şekil 6.8 Farklı k değerine göre Hassasiyet ve Anma Sonuçları (MED veri seti için)



Şekil 6.9 Farklı k değerine göre Hassasiyet ve Anma Sonuçları (TIME veri seti için)



Şekil 6.10 Farklı k değerine göre Hassasiyet ve Anma Sonuçları (TRNEWS veri seti için)

Şekil 6.6 test sürecinde kullanılan üç veri seti için TDA ve Kesik ULV yöntemlerini dizinleme işlemi sonucundaki listelenen dokümanların ortalama minimum benzerlik değerlerini farklı rank değerlerine göre karşılaştırmaktadır. MED, TIME ve TRNEWS veri setleri için sonuçlar daha benzer olduğu görülmektedir. ADI veri setinde ise artan rank değerlerine göre farklı değerler almasına karşın benzerlik değişim oranının neredeyse aynı olduğu görülmektedir. Buradaki farklılığın sebebi doküman sayısı en çok olan MED ve TRNEWS veri setindeki benzerliğin TIME veri setinden daha iyi olması da dikkate alınarak ADI veri setinin doküman sayısının az olmasına bağlanabilir. Diğer bir deyişle doküman sayısı arttıkça benzer özellik gösterdikleri düşünülebilir.

ADI veri setinin rank k değerine göre performansının gösterildiği Çizelge 6.3'de, her iki yöntemin performansı $k < 40$ olduğunda kötüdür, ancak $50 < k < 60$ şartı sağlandığında dizinleme doğruluğu artış göstermektedir. Bununla birlikte k değeri 70'den büyük olduğunda, indeksleme başarısı tekrar düşmeye başlamaktadır. Benzer şekilde MED veri setinin k değerine göre performansını gösteren Çizelge 6.4'te performansın, $k < 10$ olduğunda kötü olduğu, $20 < k < 150$ iken artış gösterdiği ve $k > 300$ şartının olduğu durumlarda azaldığı görülmektedir. TIME veri setinin

performansının ise $k < 50$ olduğunda kötü olduğu, $100 < k < 200$ şartında arttığı ve $k > 200$ olduğunda azaldığı görülmektedir. TRNEWS veri seti için ise TDA yöntemine göre $k < 100$ olduğu durumlarda iyi performans göstermediği, $100 < k < 250$ arasında başarının artış gösterdiği ve $k > 250$ şartlarının sağladığı durumlarda başarının çok olmasa da azaldığı görülmektedir. TRNEWS veri seti için Kesik ULV yöntemine göre ise $k > 100$ olduğu tüm şartlarda başarının arttığı gözlemlenmiştir.

Çizelge 6.3, Çizelge 6.4, Çizelge 6.5 ve Çizelge 6.6'da yer alan sırasıyla ADI, MED, ADI VE TRNEWS veri setleri için farklı rank k değerlerine göre başarı durumları ve her bir veri seti için rank k değerine göre hesaplanan ortalama minimum benzerlik değerlerinin gösterildiği Şekil 6.6 birlikte incelenirse; her bir veri seti için de k sayısı arttıkça minimum benzerlik değeri değişim oranının azaldığı görülmektedir. Bu doküman dizinleme sürecindeki belirtilmesi gereken benzerlik eşik değerinin belirlenmesini zorlaştırmakta ve dolayısı ile başarılı dizinleme yapılmasına engel olmaktadır. Bu nedenle hem süreçten kazanç sağlamak hem de doğru verilere ulaşmak için k değeri değişim oranının yüksek olduğu ve doküman dizinleme başarısının iyi olduğu değerleri alması önerilmektedir. Bu durum göze alındığında, hem TDA hem de Kesik ULV yöntemindeki rank k değişkeninin, ADI veri seti için 50, MED veri seti için 150, TIME veri seti için 150 ve TRNEWS için 250 değerini alması en verimli sonuçlar için önerilir.

7. TARTIŞMA VE SONUÇ

Kullanım alanı sürekli genişleyen bilgisayarlar tarafından dijital ortamda depolanan verilerin boyutları günden güne büyümektedir. Ancak bu veriler işlenmediği ya da analiz edilmediği sürece sadece bir arşivden ibarettir. Bu nedenle, istatistikçiler, ekonomistler, iş planlayıcıları, reklam analistleri ve iletişim mühendisleri gibi birçok sektör çalışanları bu depolanan verilerden anlamlı bilgiler elde etmek amacıyla sürekli araştırma ve geliştirme yapmaktadırlar. Doğal dil ile yazılan metinlerin depolanma ve erişim sürecini en etkin bir şekilde gerçekleştirmeyi amaçlayan metin tabanlı bilgiye erişim sistemleri de öne çıkan yöntemlerden birisidir. GAD ise devasa doküman yığını içerisinde kullanıcının istediği doküman ya da doküman kümesine en doğru şekilde ulaşmasını amaçlayan istatistiksel/matematiksel bir yöntemdir. Bu yöntem ile dokümanların temsil edildiği vektör uzayı elde edilir ve kullanıcı tarafından girdi olarak alınan bir sorgu cümleciğinin de bu vektör uzayındaki konumuna karşılık en yakın ya da benzer doküman listesine ulaşılır. Sorgu sonrasında çıktı olarak listelenen doküman listesi sorgu cümleciği ile ilişkili olup olmadığına göre incelenerek yöntemin başarısı irdelenir. Bilgiye erişim sistemlerinde sorgu işleminden sonra listelenen dokümanların ilgili sorgu cümleciği için anma ve hassasiyet ölçütlerine göre başarıları irdelenir. Devasa doküman yığını içerisinden önceden ilişkili olduğu doküman listesi bilinen sorgu cümleciklerinin tamamının ortalama anma ve hassasiyet değer ölçütlerinin bulunması sonucunda geliştirilen yöntemin başarısı hakkında karar verilebilir. Literatürde yaygın şekilde kullanılan ADI, MED, TIME gibi veri setlerinin yanında Türkçe veri seti olarak elde edilen Türkçe haber metinlerinin de test edildiği bu çalışmada farklı büyüklüklerde doküman yığınlarının başarısını görmek de mümkün olmuştur.

GAD sürecinin performansı ölçülürken bakılması gereken bir diğer ölçüt ise sorgulama işlemi sonrasında listelenen dokümanların minimum benzerlik değeridir. Rank k değerinin ile ters orantılı olarak değişen benzerlik değerlerinin irdelendiği Şekil 6.6'da k değerinin giderek artması sonucunda en az fark gösterdiği noktalardan ziyade en çok kırılmanın olduğu ve en az fark göstermeye başladığı nokta, sistemin başarısını belirleyen Rank k değerinin belirlenmesinde

kullanılmaktadır. Bu değerin belirlenmesinde en önemli etken ise sorgu sonrasında listelenen doküman listesinin hassasiyet değerinin en yüksek olduğu değere yaklaştığı ve k değeri arttıkça hassasiyet değerindeki değişimin en az olduğu nokta olarak irdelenmektedir. Ancak k değeri arttıkça maliyet çok artacağı için hassasiyet değerinin tatmin edici olduğu noktada alınması sistemin verimliliği açısından ayrı bir önem taşımaktadır. Sorgu sonrasındaki listelenen ilk dokümanların ilk sıradakilerinin önem arz etmesinden dolayı Çizelge 6.3, Çizelge 6.4, Çizelge 6.5 ve Çizelge 6.6 sonuçlarda görüldüğü üzere listelenen dokümanların %10'luk kısmındaki başarının %50'lik kısımdan daha iyi olduğu görülmektedir. Bu nedenle listelenen dokümanların tamamından ziyade belirli bir eşik değerinden sonrasındakilerinin listelenmesi verimliliği artırmaktadır. Rank k değerinin belirlenmesinde sorgu işleminin ardından ölçülen başarı ölçütlerinin yanında benzerlik eşik değerinin belirleneceği en uygun nokta da bu anlamda önem arz etmektedir.

GAD'nın temelinde yatan matris ayrışımı genellikle TDA'dır. Ancak TDA'nın zaman karmaşıklığı, blok güncelleme zorluğu gibi nedenler bu anlamda önemli derecede dezavantaj olarak görülmektedir. Bu çalışmada GAA sürecinde kullanılan matris ayrışımı olarak zaman karmaşıklığı daha az maliyetli ve blok güncelleme süreci daha kolay gerçekleşen K-ULVA önerilmektedir. Böylece TDA'nın dizinleme sürecinde ve sonrasında gerçekleşebilecek olası güncelleme durumlarındaki dezavantajların azalması sağlanmıştır. Elde edinilen deneyimler sonucunda, özellikle sinyal işleme çalışmalarında olmak üzere veri sıkıştırma, eksik veri tamamlama, görüntü işleme, ses işleme, gürültülü veri temizleme gibi birçok alanda karşılaşılan TDA'nın uygulama alanlarında alternatif yöntem olarak K-ULVA yönteminin kullanılabileceği düşünülmektedir. Ayrıca bu çalışma metin özetleme, metin benzerliği, anahtar kelime çıkarma, yazar tespiti, metin sınıflandırma gibi birçok alanda çalışma konusu olarak ele alınabilir.

İlerleyen çalışmalarda doküman yığının daha çok olduğu ve birden çok bilgi erişim sisteminin paralel çalıştığı bir büyük veri ve doküman dizinleme sistemi üzerine çalışılması düşünülmektedir. İşlenecek doküman yığınının parçalanarak daha fazla bilgisayar vasıtasıyla işlenmesi sonucunda veri yığını arttıkça sorun haline gelen hız sorununun ve donanımsal kısıtların önüne geçilmesi planlanmaktadır.

KAYNAKLAR

[1] Stonebraker M, Agrawal R, Dayal U, Neuhold EJ, Reuter A. DBMS research at a crossroads: The vienna update. InVLDB 1993 Aug 24 (Vol. 93, pp. 688-692).

[2] Han J, Pei J, Kamber M. Data mining: concepts and techniques. Elsevier; 2011 Jun 9.

[3] Akpınar H. Veri tabanlarında bilgi keşfi ve veri madenciliği. İÜ İşletme Fakültesi Dergisi. 2000;29(1):1-22.

[4] Chen MS, Han J, Yu PS. Data mining: an overview from a database perspective. IEEE Transactions on Knowledge and data Engineering. 1996 Dec;8(6):866-83.

[5] Fayyad U, Piatetsky-Shapiro G, Smyth P. From data mining to knowledge discovery in databases. AI magazine. 1996 Mar 15;17(3):37.

[6] Sever H, Buket OĞ. Veri Tabanlarında Bilgi Keşfine Formel Bir Yaklaşım Kısım I: Eşleştirme Sorguları ve Algoritmalar. Bilgi Dünyası. 2003 Oct 10;3(2):173-204.

[7] Witten IH, Frank E, Hall MA, Pal CJ. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann; 2016 Oct 1.

[8] Erbay H, Varcin F, Horasan F. Alternate Low Rank Approximation In Latent Semantic Analysis.

[9] Golub GH, Van Loan CF. Matrix computations. JHU Press; 2012 Dec 27.

[10] Kumar CA, Radvansky M, Annapurna J. Analysis of a vector space model, latent semantic indexing and formal concept analysis for information retrieval. Cybernetics and Information Technologies. 2012 Mar 1;12(1):34-48.

[11] Song G, Ye Y, Du X, Huang X, Bie S. Short text classification: A survey. Journal of Multimedia. 2014 May 1;9(5):635.

[12] Zhang W, Yoshida T, Tang X. TFIDF, LSI and multi-word in information retrieval and text categorization. InSystems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on 2008 Oct 12 (pp. 108-113). IEEE.

[13] Wang J, Peng J, Liu O. A classification approach for less popular webpages based on latent semantic analysis and rough set model. Expert Systems with Applications. 2015 Jan 1;42(1):642-8.

[14] Alkouz A, De Luca EW, Albayrak S. Latent Semantic Social Graph Model for Expert Discovery in Facebook. InIICS 2011 (pp. 128-138).

[15] Nasir H, Stanković V, Marshall S. Singular value decomposition based fusion for super-resolution image reconstruction. Signal Processing: Image Communication. 2012 Feb 1;27(2):180-91.

[16] Duman E., Erbay H., Web Sayfalarının Gizli Anlam Analizi Yaklaşımıyla Otomatik Olarak Sınıflandırılması, Yüksek Lisans Tezi, 2013.

[17] Shima K, Todoriki M, Suzuki A. SVM-based feature selection of latent semantic features. Pattern Recognition Letters. 2004 Jul 2;25(9):1051-7.

[18] Uysal AK, Gunal S. Text classification using genetic algorithm oriented latent semantic features. Expert Systems with Applications. 2014 Oct 1;41(13):5938-47.

[19] Güran A., Otomatik Metin Özetleme Sistemi, Doktora Tezi, 2013.

[20] Steinberger J. Text summarization within the LSA framework. PhD diss. 2007 Jan 26.

[21] Murray G, Renals S, Carletta J. Extractive summarization of meeting recordings, In Proceedings of the 9th European Conference on Speech Communication and Technology, September 2005, Lisbon, Portugal..

[22] Lee JH, Park S, Ahn CM, Kim D. Automatic generic document summarization based on non-negative matrix factorization. Information Processing & Management. 2009 Jan 1;45(1):20-34.

[23] Ozsoy MG, Cicekli I, Alpaslan FN. Text summarization of turkish texts using latent semantic analysis. InProceedings of the 23rd international conference on computational linguistics 2010 Aug 23 (pp. 869-876). Association for Computational Linguistics.

[24] O'Brien GW. Information management tools for updating an SVD-encoded indexing scheme (Master's thesis, University of Tennessee, Knoxville).

[25] Varçın F., Kesik Ulv Ayrışımı İle Gizli Anlamsal Dizinleme, Yüksek Lisans Tezi, Kırıkkale Üniversitesi, Fen Bilimleri Enstitüsü, 2016.

[26] Oğuzlar A. Veri ön işleme. Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi. 2003(21).

[27] Piramuthu S. Evaluating feature selection methods for learning in data mining applications. European journal of operational research. 2004 Jul 16;156(2):483-94.

[28] Brachman RJ, Anand T. The process of knowledge discovery in databases. InAdvances in knowledge discovery and data mining 1996 Feb 1 (pp. 37-57). American Association for Artificial Intelligence.

[29] Bayer H., Veri Madenciliğinde Bir Metin Madenciliği Uygulaması, Yüksek Lisans Tezi, İstanbul,2011.

[30] Akpınar H., DATA Veri Madenciliği, Veri Analizi, Papatya Yayıncılık, 1. Basım, Eylül 2014.

[31] Karahan Adalı G., Veri Madenciliğinde Birliktelik Yöntemleri ve Müşteri İlişkileri Yönetimine İlişkin Bir Uygulama, Doktora Tezi, İstanbul,2011.

[32] Fan W, Wallace L, Rich S, Zhang Z. Tapping the power of text mining. Communications of the ACM. 2006 Sep 1;49(9):76-82.

[33] Visa A. Technology of text mining. InInternational Workshop on Machine Learning and Data Mining in Pattern Recognition 2001 Jul 25 (pp. 1-11). Springer, Berlin, Heidelberg.

[34] Yang HC, Lee CH. A text mining approach for automatic construction of hypertexts. Expert Systems with Applications. 2005 Nov 1;29(4):723-34.

[35] Ergün K., Metin Madenciliği Yöntemleri İle Ürün Yorumlarının Otomatik Değerlendirilmesi, Doktora Tezi, 2012.

[36] Hearst MA. Untangling text data mining. InProceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics 1999 Jun 20 (pp. 3-10). Association for Computational Linguistics.

[37] Kantardzic M. Data mining: concepts, models, methods, and algorithms. John Wiley & Sons; 2011 Jan 5.

[38] Delen D, Crossland MD. Seeding the survey and analysis of research literature with text mining. Expert Systems with Applications. 2008 Apr 1;34(3):1707-20.

[39] Turban E, Sharda R, Delen D, Efraim T. Decision support and business intelligence systems. Pearson; 2014.

[40] Saraçoğlu R, Tütüncü K, Allahverdi N. A new approach on search for similar documents with multiple categories using fuzzy clustering. Expert Systems with Applications. 2008 May 1;34(4):2545-54.

[41] Adeva JJ, Calvo R. Mining text with Pimiento. IEEE internet computing. 2006 Jul 1;10(4):27-35.

[42] Kunc M. Web Mining Overview. InProceedings of the 13th Conference Student EEICT 2007 Volume 2007 (pp. 391-395). Brno University of Technology.

[43] Cooley R, Mobasher B, Srivastava J. Web mining: Information and pattern discovery on the world wide web. InTools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on 1997 Nov 3 (pp. 558-567). IEEE.

[44] Srivastava J, Cooley R, Deshpande M, Tan PN. Web usage mining: Discovery and applications of usage patterns from web data. Acm Sigkdd Explorations Newsletter. 2000 Jan 1;1(2):12-23.

[45] Martin DI, Berry MW. Mathematical foundations behind latent semantic analysis. Handbook of latent semantic analysis. 2007 Feb 15:35-56.

[46] Berry MW, Fierro RD. Low-rank Orthogonal Decompositions for Information Retrieval Applications. Numerical linear algebra with applications. 1996 Jul;3(4):301-27.

[47] Varçın F, Erbay H, Horasan F. Latent semantic analysis via truncated ULV decomposition. InSignal Processing and Communication Application Conference (SIU), 2016 24th 2016 May 16 (pp. 1333-1336). IEEE.

[48] Dumais ST. Improving the retrieval of information from external sources. Behavior Research Methods, Instruments, & Computers. 1991 Jun 1;23(2):229-36.

[49] Berry MW, Dumais ST, O'Brien GW. Using linear algebra for intelligent information retrieval. SIAM review. 1995 Dec;37(4):573-95.

[50] Lochbaum KE, Streeter LA. Comparing and combining the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval. Information Processing & Management. 1989 Jan 1;25(6):665-76.

[51] Barlow JL, Erbay H. Modifiable low-rank approximation to a matrix. Numerical Linear Algebra with Applications. 2009 Oct;16(10):833-60.

[52] Metin S. K.,Türkçede Hesaplamalı Metin Analizi, Doktora Tezi, 2011.

[53] Oktay M.,A Text Processing and Analysis Tool For Turkish, M. S. Thesis, 2007.

[54] Kurt, Z., Temel Bileşen Analiziyle Öznitelik Seçimi Ve Görsel Nesne Sınıflandırma, Yüksek Lisans Tezi, 2013.

[55] Bilgin,G., Hiperspektral Görüntülerin Eğiticisiz Bölütlenmesi, Doktora Tezi, 2009.

[56] Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. Journal of the American society for information science. 1990 Sep;41(6):391-407.

[57] Berry MW, Drmac Z, Jessup ER. Matrices, vector spaces, and information retrieval. SIAM review. 1999;41(2):335-62.

[58] Jessup ER, Martin JH. Taking a new look at the latent semantic analysis approach to information retrieval. Computational information retrieval. 2001 Jan 1;2001:121-44.

[59] Higham NJ. Accuracy and Stability of Numerical Analysis. Cambridge University Press, 2002.

[60] Barlow JL, Smoktunowicz A. Reorthogonalized block classical Gram–Schmidt. Numerische Mathematik. 2013 Mar 1;123(3):395-423.

[61] Barlow JL, Erbay H, Slapnicar I. An alternative algorithm for the refinement of ULV decompositions. SIAM Journal on Matrix Analysis and Applications. 2005;27(1):198-211.

[62] Erbay H. Modifying rank-revealing decompositions. The Pennsylvania State University; 2000 Jan 1.

[63] Brand M. Fast low-rank modifications of the thin singular value decomposition. Linear algebra and its applications. 2006 May 1;415(1):20-30.

ÖZGEÇMİŞ

Adı Soyadı : Fahrettin HORASAN
Doğum Tarihi : 15.07.1988
Yabancı Dil : İngilizce

Eğitim Durumu :
Lisans : Selçuk Üniversitesi, Bilgisayar Sist. Öğr.. 2006-2011
Yüksek Lisans : Sakarya Üniversitesi Bilgisayar Müh. 2012-2014

Çalıştığı Kurum/Kurumlar ve Yıl/Yıllar:
Sakarya Üniversitesi, Teknoloji Fakültesi, Bilgisayar Müh. 2012-2014
Kırıkkale Üniversitesi, Mühendislik Fakültesi, Bilgisayar Müh.2014 –

Yayınları (SCI-E) :
Erbay H, Varçın F, Horasan F, Biçer C. Block Classical Gram-Schmidt Based Block Updating in Low-Rank Matrix Approximation. Turkish Journal of Mathematics. 2018

Yayınları (Diğer) :
Varçın F, Erbay H, Horasan F. Latent semantic analysis via truncated ULV decomposition. InSignal Processing and Communication Application Conference (SIU), 2016 24th 2016 May 16 (pp. 1333-1336). IEEE.

Bildirileri

1. Erbay H, Varçın F. and Horasan F., Alternate Low Rank Approximation In Latent Semantic Analysis, MMA2016, June 1–4, 2016.

2. Erbay H, Varçın F. and Horasan F., Gram–Schmidt Based Truncated ULV Block Update, The International Conference on Engineering and Natural Sciences, 2016.

3. Erbay H., Horasan,F. Varçın F. and Deniz E., Alternate Matrix Approximation in Latent Semantic Analysis, International Conference On Mathematics And Engineering , Istanbul, 2017

4. Deniz E., Erbay H., Horasan,F. and Varçın F. Text Classification with Latent Semantic Analysis, International Conference On Mathematics And Engineering , Istanbul, 2017

5. Horasan F., Erbay H, Varçın F. and Deniz E., Search engine optimization with latent semantic analysis, International scientific and vocational studies congress , 2017

6. Varçın F., Erbay E., Horasan F. and Deniz E., Gizli Anlamsal Dizinleme İle Metin Sınıflandırmada Farklı Benzerlik Metotlarının Performanslarının Karşılaştırılması, UMTEB,2018

7. Varçın F., Erbay E., Horasan F. and Deniz E., Farklı benzerlik metotlarının Kesik ULV Tabanlı Gizli Anlamsal Dizinleme Performansına Etkisi, UMTEB,2018

Proje :

Bu tez Kırıkkale Üniversitesi "2016-150" kodlu BAP projesi tarafından desteklenmiştir.

Araştırma Alanları :

Büyük Veri, Makine Öğrenmesi, Veri Madenciliği, Arama Motorları, Doğal Dil İşleme, Dizinleme, Matris Ayrışımları ve Optimizasyon Algoritmaları

Ödüller:

Selçuk Üniversitesi, Teknik Eğitim Fakültesi, Bilgisayar Sist. Öğretmenliği Lisans Program Birincisi, 2011