METİN MADENCİLİĞİ KULLANILARAK
TALEP TANIMA VE YÖNLENDİRME SİSTEMİ

Yasin SANCAR

Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Doç. Dr. Tevhit KARACALI
2016

ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

METİN MADENCİLİĞİ KULLANILARAK TALEP TANIMA VE YÖNLENDİRME SİSTEMİ

Yasin SANCAR

BİLGİSAYAR MÜHENDİLİĞİ ANABİLİM DALI

ERZURUM
2016

T.C.
ATATÜRK ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü Müdürlüğü

TEZ ONAY FORMU

KURUMSAL YAZIŞMALARDA METİN MADENCİLİĞİ KULLANILARAK
TALEP TANIMA ve YÖNLENDİRME

Doç.Dr. Tevhit KARACALI danışmanlığında, **Yasin SANCAR** tarafından hazırlanan bu çalışma, 23/06/2016 tarihinde aşağıdaki jüri tarafından **Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği** Bilim Dalı'nda yüksek lisans tezi olarak **oybirliği** / ~~**oy çokluğu**~~ **(.../...)** ile kabul edilmiştir.

Başkan: Doç. Dr. Tevhit KARACALI İmza :

Üye : Yrd. Doç. Dr. Barış ÖZYER İmza :

Üye : Yrd. Doç. Dr. Ahmet DUMLU İmza :

Yukarıdaki sonuç;

Enstitü Yönetim Kurulu'nun 11.08/2016 tarih ve .32..../...22.... nolu kararı ile onaylanmıştır.

Prof. Dr. Ertan YILDIRIM
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildiriş, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ÖZET

Yüksek Lisans Tezi

METİN MADENCİLİĞİ KULLANILARAK TALEP TANIMA VE YÖNLENDİRME SİSTEMİ

Yasin SANCAR

Atatürk Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Tevhit KARACALI

Günümüzde büyük ölçekli işletmelerde birçok birim bulunmakta ve bu birimlere gönderilen dilekçeler, bir görevli tarafından ait oldukları birimlere teslim edilmektedir. Bu yol ile teslim edilen dilekçeler, büyük ölçekli kurumlarda zaman ve hız açısından eksiklik teşkil etmektedir. Dilekçenin gönderildiği anda teslim edilemeyip ilgili yerlere teslim edilmesi için kategorize işlemlerinin yapılması ve buna bağlı olarak verilen cevapların da aynı şekilde karşı tarafa iletilmesi, zaman ve hız kaybına sebep olmakta ve bu durum işlerin aksamasına sebebiyet vermektedir. Ayrıca yöneticilerin onay beklemekte olan dilekçeleri takip edememesi, dilekçe tanıma konusunda yeni bir çalışma ihtiyacı doğurmuştur.

Bu tez çalışmasında dilekçelerin dağıtım problemlerine çözüm olarak kurumun herhangi bir birimden gönderilen elektronik dilekçe, dilekçe tanıma sisteminden geçirilerek işlenmektedir. Dilekçe içerisinde bulunan cümleler OCR (Optik Karakter Tanıma) ile karakter taraması yapılarak ve eksik ya da yanlış tanınmış karakterler Levenshtein algoritması ile düzeltilerek bilgisayar ortamına aktarılmaktadır. Daha sonra metin madenciliği yöntemi olan Multinominal Naive Bayes yöntemi ile dilekçeler kategorize edilerek ait oldukları birimlere otomatik olarak yönlendirilmektedir.

2016, 47 sayfa

Anahtar Kelimeler: Dilekçe, OCR, levenshtein algoritması, metin madenciliği, Multinominal Naive Bayes

ABSTRACT

MS Thesis

PETITION DETECTION AND FORWARD SYSTEM USING TEXT MINING

Yasin SANCAR

Atatürk University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Tevhit KARACALI

Nowadays, there are many departments in large scale institutions and the petitions which are sent to these departments are delivered tothe departments which they belong to by an employee. The petitions delivered through this way represent the lack of time and speed in the large scale institutions. Making categorization for submisson to the appropriate departments and correspondingly transmitting to the other party in the same manner of given answers causes a loss of time and speed. This case gave rises to the interruption of this work. Fort he automatic submisson of the petition, administrastors can not track petitions pending approval, so that a new study has led to need to categorize the petitions into departments.

In this thesis, as a solution to distribution problems of the petitions the electronic petition which sent from anyone emplooye in the department. The sentences include in the text of petition is forwarded to computer area with OCR and Levenshtein algorithm is used to correct the incorrect words. And then a text mining technique namely Multinominal Naive Bayes is used to classify the petitions and they are forwarded automatically to departments which they belong to.

2016, 47 pages

Keywords: Petition, OCR, levenshtein algorithm, text mining, Multinominal Naive Bayes

TEŞEKKÜR

Bu tezin araştırma konusunun yürütülmesinde danışmanlık yaparak bana yol gösteren danışman hocam Sayın Doç. Dr. Tevhit KARACALI başta olmak üzere çalışmalarım sırasında manevi desteklerini esirgemeyen annem, babam ve kardeşlerime şükranlarımı sunuyorum.

Yasin SANCAR

Mayıs, 2016

İÇİNDEKİLER

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

C	Sınıf etiketi
N	Veri setinin boyutu
$\in$	Eleman
t_i	i. Test dokümanı
f_{ni}	Test dokümanındaki kelime sayısı

Kısaltmalar

OCR	Optik Karakter Tanıma
NB	Naive Bayes
MNB	Multinominal Naive Bayes
DVM	Destek Vektör Makineleri
GYA	Geri Yayılım Algoritması
kNN	K-En Yakın Komşu
ROC	Alıcı İşlem Karakteristikleri Eğrisi
MODI	Microsoft Office Document Imaging
DPI	Inç başına düşen nokta sayısı

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Dilekçe, bir talepte bulunmak, bilgi almak ya da bilgi vermek üzere kişilerden ilgili birimlere yazılan bir mektuptur. Küçük çaplı kurumlarda dilekçe yönetim işlemi kolay ve hızlı olmasına rağmen büyük çaplı kurum ve kuruluşlarda dilekçelerin ilgili birimlere zamanında teslim edilmesi, çalışan sayısı ve birim sayısı arttıkça zorlaşmakta ve bu durum yapılacak işlerin aksamasına neden olmaktadır.

Teknolojinin hızla ilerlediği günümüzde kurum içi yazılan dilekçelerin elden teslim edilmesi yerine elektronik ortam üzerinden hızlı bir şekilde gönderilmesi ve aynı şekilde cevap alınması, dilekçe teslimi konusunda meydana gelen problemleri en aza indirgemektedir. Kurumda çalışan kişi tarafından yazılan dilekçelerin otomatik olarak kategorize edilip hangi birime gönderileceği bilgisini tespit etmek için öncelikle elektronik ortamda yazılan metnin tanınması gerekmektedir. Metnin doğru bir şekilde tanımlanması ve daha sonra hangi birim ile ilgili olduğunun tespiti, "metin madenciliği" yöntemi ile yapılmaktadır.

Bir veri madenciliği çalışması olan metin madenciliği, metinleri veri kaynağı olarak kullanarak bu verilerden anlamlı bilgi ve ilişkileri çıkarmada kullanılan yöntemler olarak tanımlanmaktadır. Veri madenciliğinden farklı olarak metin madenciliği yapısal olmayan veriler üzerinde işlem yapmaktadır. Yapısal veriden kasıt, bir yapı içerisinde işlenebilen bilgi anlamına gelmektedir. Hastanelerde, öğrenci bilgi sistemleri gibi kayıt tabanlı sistemlerde genellikle yapısal veriler bulunmaktadır. SQL, Oracle, Access gibi veritabanlarında saklanan, satır ya da id bazlı bilginin erişimine olanak veren yapılar üzerinde sorgulama yapma ya da o bilgiyi doğrudan kullanma işlemi kolay bir şekilde yapılmaktadır. Fakat yapısal olmayan verilerde verilerin ayrık olarak elde edilmesi doğrudan mümkün olmamaktadır. Resim dosyaları, pdf, word ve text dosyaları, internet üzerinde tutulan kayıt dosyaları ve elektronik postalar yapısal olmayan verilere örnektir (Dolgun vd 2009).

Bu tez çalışmasında da kurum içi ya da kurumlar arası dilekçelerin tanınmasında metin madenciliği kullanılacak ve dilekçeler resim ve pdf dosyası olarak saklanacaktır. Resim dosyaları üzerinde metinlerin okunması işlemi yapılırken pdf dosyaları, dilekçeye cevap yazacak olan kişi için kolaylık olması açısından kayıt altına alınmaktadır. Gerçekleştirilen internet tabanlı bu sistem ASP.NET ortamında C# programlama dili kullanılarak tasarlanmış olup veritabanı yönetim sistemi olarak Microsoft MSSQL kullanılmıştır. Veritabanına resim dosyaları olarak kaydedilen dilekçelerde ye alan cümleler, Optik Karakter Tanıma (OCR) yöntemi ile sisteme tanıtılmış, karakter taraması sonucunda okunamayan veya yanlış okunan veriler "levenshtein algoritması" aracılığı ile tamamlanmış veya düzeltilmiştir. Bu işlemlerden sonra metin madenciliği ile dilekçelerin hangi birime gönderileceği hususunda kategorizasyon işlemi yapılmış ve otomatik olarak ilgili birime yönlendirilmiştir.

Çalışmanın ikinci bölümünde dilekçe tanıma sistemi için kullanılan yöntemlerden OCR, levenshtein algoritması ve metin madenciliği yöntemi açıklanarak bu alanda yapılan literatür çalışmalarından bahsedilmiştir. Üçüncü bölümde tasarlanan sistemin arayüzü ve çalışma mantığı anlatılarak uygulamanın testi için 225 adet dilekçe üzerinde tanıma işlemi yapılmıştır. Dördüncü bölümde test sonucunda elde edilen bulgular üzerinde durulmuştur.

Sonuç ve öneriler bölümünde ise .NET ortamında gerçekleştirilen dilekçe tanıma sisteminin başarı oranından, bu başarı oranını takiben sistemin kurumlarda kullanılması sonucunda daha hızlı bir dilekçe takip ortamının bulunabileceğinden ve sistemin daha doğru, daha etkili ve daha hızlı çalışması için kullanılacak yeni yöntemlerden bahsedilmiştir.

2. KURAMSAL TEMELLER

2.1. Dilekçe Tanımı

Dilekçeler, özel şahısların taleplerini yazılı olarak iletmesi için resmi bir kurum ya da kuruluşa sunulan yazılardır. Bu yazılar belirli bir formatta iletilmesi gereken kuruma hitap edilerek yazılmalıdır. Genel olarak 6 temel öğeden oluşan dilekçede tarih, makam adı, konu metni, imza, dilekçeyi yazan kişinin adı ve dilekçeyi yazan kişinin adres-telefon bilgileri bulunmalıdır.

2.2. Dilekçe Hakkının Kullanılması

Anayasının 3071 sayılı kanununda Türk vatandaşların ve Türkiye'de ikamet eden yabancı uyruklu insanların kendileri ile ya da kamu ile ilgili istek ve şikâyetlerini yetkili makamlara yazılı bir şekilde sunabilme haklarının olduğundan bahsedilmektedir. 3071 sayılı kanunun 7. maddesinde, yetkili makamlara iletilen dilekçe sonucunun dilekçe sahibine en geç otuz gün içerisinde gerekçeli olarak cevap verilmesi gerektiği hükmüne yer verilmiştir.

Anayasaya göre dilekçe hakkının kullanımı çerçevesinde yetkili makama yapılan başvurular aşağıda açıklanan usüllere uygun olarak cevaplandırılacaktır (Anonim 2016a):

- Özel şahısların başvuru dilekçelerini alan ilgili makamlar, dilekçelerin alındığı tarih, kayıt numarası ve konusunu belirten alındı belgesi düzenlemeli ve bu belgeleri başvuru sahiplerine bir ücret talep etmeden teslim edeceklerdir.
- İsim, soyisim, adres bilgisi ve imza bulundurmayan, belirli bir konuyu içermeyen ya da yargı mercilerinin görevine giren konularla ilgili dilekçeler cevaplandırılmayacaktır.
- Başka bir idari makamın görev alanında bulunan başvurular, ilgili makama yönlendirilecek ve dilekçe sahibine bu konuda bilgi verilecektir.

- Özel şahısların usülüne uygun dilekçe ile yaptıkları başvuruların gecikmeksizin en kısa sürede cevaplandırılması yasal bir zorunluluktur. Yetkili idari makam tarafından dilekçe sahibine en geç otuz gün içerisinde cevap verilecek, eğer işlem devam ediyorsa başvurunun sonucu hakkında dilekçe sahibine bilgi verilecektir.

Bu bağlamda kurum ve kuruluşlarda dilekçelerin hızlı bir şekilde takibi ve gerekli birimlere ulaştırılması hem kanuni açıdan hem de kurumların gelişimi açısından büyük önem arz etmektedir.

2.3. Optik Karakter Tanıma (Optic Charachter Recognition - OCR)

Bir bilgisayara veri girmenin geleneksel yolu klavye üzerinden geçmektedir. Fakat bu her zaman en iyi ve en etkili çözüm değildir. Birçok durumda otomatik tanıma alternatif olarak düşünülmektedir. Otomatik tanıma için çeşitli teknolojiler bulunur ve bu teknolojiler uygulamaların farklı alanlarını kapsar. Optik karakter tanıma (OCR), otomatik tanıma gerçekleştiren yöntemlere aittir. Otomatik tanıma uygulamalarına konuşma tanıma, radyo frekans (arabaların tanınması), görü sistemleri, manyetik şerit ve optik işaret okuma örnek olarak verilmektedir (Eikvil 1993).

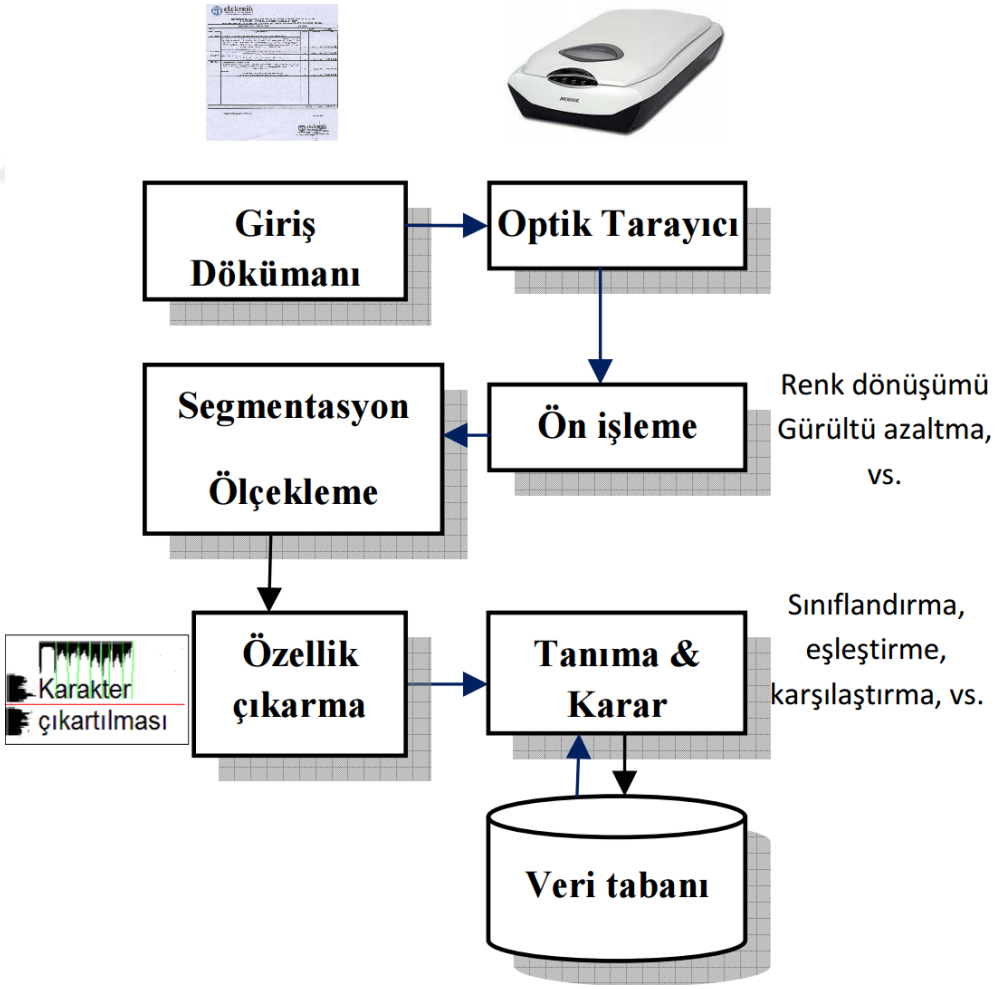
Yazılı bilgilerin klavye aracılığı ile bilgisayara aktarılması ya da elektronik ortamda bir belgenin tanımlanması işleminde pratiklik ve maliyet geri planda kalmaktadır. Bu maliyeti en aza indirgemek ve otomatik bir sistem aracılığıyla hız kazanmak için elle ya da elektronik olarak daha önceden yazılmış belgeler üzerinde görüntü alınarak bu görüntüler üzerindeki karakterler ya da metin bilgileri okunmaktadır (Karasu and Baştan 2015). Okunan karakterler ASCII koduna dönüştürülerek bilgisayar ortamında anlam kazanan sayısal verilere çevrilmektedir (Kır vd 2011).

2.3.1. OCR Sisteminin genel yapısı

Resimlerdeki metin verileri, otomatik bilgi notu, indeksleme ve resimlerin yapılandırılması için gerekli bilgiler içermektedir. Bu bilgilerin çıkarımı, metni tespit

etme, takip etme ve tanımayı sağlamaktadır. Karakter tanıma sistemleri da farklı veri alanlarından metnin içeriğini çıkarmak için kullanılan elektronik dosyalarda metin karakterlerini tanır (Nathiya and Pradeepa 2013). Şekil 2.1'de bir OCR sisteminin genel şeması gösterilmiştir. Öncelikle optik bir sistem aracılığı ile taranan dokümanlar veya el yazıları sayısal görüntüye dönüştürülerek bilgisayar ortamına aktarılır. Karakterlerin yazım şekillerinin karmaşıklığı, oluşan gürültüler, kelime haznesinin büyüklüğü gibi metrikler karakter tanımanın hızını ve doğruluk oranını etkilemektedir.

Elektronik dilekçelerin veya el yazısı verilerinin optik tarayıcıdan geçirilmesinden sonra ön işleme, segmentasyon, özellik çıkartma ve karakter tanıma işlemlerinin gerçekleştiği OCR sisteminin genel yapısı Şekil 2.1'de gösterilmektedir.



Şekil 2.1. Optik Karakter Tanıma (OCR) sisteminin genel yapısı (Kır vd 2011)

RGB renkli resim şeklinde kayıt edilen dosyalar daha fazla gürültüye sahiplerdir. Bu yüzden RGB değerlerinin yol açtığı karmaşıklığı önlemek için renkli resimden gri seviyeye dönüştürme işlemi yapılmaktadır. Bu aşamaya ön işleme aşaması denilmektedir. Daha sonra yazının arka planı ile metnin özünü ayrıştırmak için eşikleme işlemi yapılmaktadır. Eşikleme aşamasından sonra gürültü azaltıcı algoritmalar kullanılarak resim üzerinde satır satır tarama işlemi için yatay projeksiyon kullanır. Satır taramadan sonra dikey projeksiyon ile karakterler ayrıştırılır (Kır vd 2011). Farklı boyutlarda elde edilen karakterleri standartlaştırmak için ölçeklendirme işlemi yapılır ve bu aşamadan sonra gelen özellik çıkarımı, OCR sisteminin en kritik aşamasıdır. Çünkü karakterleri doğru bir şekilde sınıflandırmak için, belirlenen özelliklerin ayırt edici olması gerekmektedir.

2.4. Levenshtein Algoritması

Özellikle arama motorlarında kullanılabilen bir model sunan Levenshtein algoritması, eksik ya da yanlış kelimeler ile arama yapan kullanıcılar için kelimelerin düzeltilmiş halini bulup öneri sunan bir algoritmadır. Google arama motoru düşünüldüğünde sözlükte yer almayan kelimelerin veya yanlış yazılan kelimelerin düzeltilip kullanıcıya öneri olarak sunulduğu görülmektedir. Bu algoritma, karakter tanımada eksik ya da hatalı karakterlerin tespit edilip doğru karakterlerin bulunması açısından çok önemlidir. Bu çalışma kapsamında da Levenshtein algoritması, OCR sisteminden geçirilen karakterlerin yanlış ya da eksik olması durumda tamamlayıcı görev üstlenir.

Benzerlik kıyaslama algoritması olarak da bilinen bu algoritma iki kelime arasındaki benzerlik oranından yararlanır. İki kelimeden birinin diğerine dönüştürülmesi için gereken işlem sayısı ya da maliyet olarak değerlendirilir ve bu sayısal değer, algoritmanın sonucudur ve Levenshtein mesafesi olarak adlandırılır. Elde edilen mesafe değeri ne kadar düşükse ise maliyet o kadar azdır ve bu durum istenen bir durumdur. Bu kelimelerin birbirlerine yakınlığı derecelendirildiğinde daha az yapılan değişiklik, iki kelimenin birbirine daha benzer olduğunu gösterir.

Levenshtein mesafeyi hesaplamak için yaygın olarak kullanılan aşağıdan yukarıya dinamik programlama algoritması (n+1) x (m+1) matrisinin kullanımını içerir. Buradaki n ve m iki kelimenin uzunluklarıdır. Bu algoritma, düzenleme mesafesi için Wagner-Fischer algoritmasına dayanır. Aşağıda, m kelimesinin uzunluğu s ve n kelimesinin uzunluğu t hesaplanarak iki kelime arasındaki Levenshtein mesafesini bulmayı sağlayan sözde kod verilmiştir (Su *et al.* 2008):

```
int LevenshteinDistance (char s[1…m], char t[1…n])
        // d, m+1 satır ve n+1 sütun sayısından oluşan bir tablo
        declare int d[0…m, 0…n]
        for i= 0'dan m'e kadar
          d[i, 0] := i
        for j =0'dan n'e kadar
          d[0, j] := j
        for i=1'den m'e kadar
          for j=1'dan n'e kadar
            if s[i] = t[j] then cost := 0
                else cost := 1
            d[I, j] := minimum(
                d[i-1, j] +1, // silme
            d[i, j-1] +1, //ekleme
                d[i-1, j-1] +cost // yerine koyma)

return d[m, n]
```

Hesaplanan Levenshtein mesafe değerinin sıfır olması, karşılaştırılan iki kelimenin aynı olduğunu göstermektedir. Levenstein mesafesi aracılığı ile optik tarayıcıdan geçirilen dilekçeler resim olarak saklandıktan sonra, bu yazıların OCR ile tanınması sırasında tam olarak bulunamayan kelimeler düzeltilmek üzere benzer kelimelerin çıkarımı yapılabilmektedir.

OCR sisteminde taranan dilekçe metinleri üzerinden düşünüldüğünde elde edilen kelimenin veritabanında bulunan kelimeler ile karşılaştırılması yapılmaktadır. Bu karşılaştırma işlemi için elde edilen kelime ve veritabanındaki kelime sırasıyla matrisin ilk sütununa ve ilk satırına yerleştirilmektedir. Sıra ile karşılaştırılan karakterler

uyuşuyorsa ve karakter sayısı eşit ise sayı sabit kalır, eğer tam tersi durum ise sayı bir artırılır. Bu işlemler tüm karakterler için yapıldığında matrisin sağ alt köşesinde hesaplanan sayı, Levenshtein mesafeyi vermektedir.

Dilekçe örneklerinden alınan bazı kelimelerin yanlış tespitinde kullanılan bu yöntemin örnekleri Şekil 2.2'de gösterilmektedir.

		s	ı	n	a	v
	0	1	2	3	4	5
s	1	0	1	2	3	4
ı	2	1	0	1	2	3
n	3	2	1	0	1	2
a	4	3	2	1	0	1
v	5	4	3	2	1	0
ı	6	5	3	3	2	1

		d	e	k	a	n	l	ı	ğ	ı
	0	1	2	3	4	5	6	7	8	9
d	1	0	2	3	4	5	6	7	8	9
k	2	1	0	1	2	3	4	5	6	7
a	3	2	1	1	1	2	3	4	5	6
n	4	3	2	2	2	1	2	3	4	5
l	5	4	3	3	3	2	1	2	3	4
ı	6	5	4	4	4	3	2	1	2	2
k	7	6	5	4	5	4	3	2	3	3

		t	a	l	e	p
	0	1	2	3	4	5
k	1	1	2	3	4	5
a	2	2	1	2	3	4
y	3	3	2	2	3	4
ı	4	4	3	3	3	4
p	5	5	4	4	4	3

		t	e	ş	e	k	k	ü	r
	0	1	2	3	4	5	6	7	8
m	1	1	2	3	4	5	6	7	8
ü	2	2	2	3	4	5	6	6	7
t	3	2	3	3	4	5	6	7	7
e	4	3	2	3	3	4	5	6	7
ş	5	4	3	2	3	4	5	6	7
e	6	5	3	3	2	3	4	5	6
k	7	6	4	4	3	2	2	3	4
k	8	7	5	5	4	2	2	3	4
i	9	8	6	6	5	3	3	3	4
r	10	9	7	7	6	4	4	4	3

Şekil 2.2. Levenshtein matrisleri

2.5. Metin Madenciliği

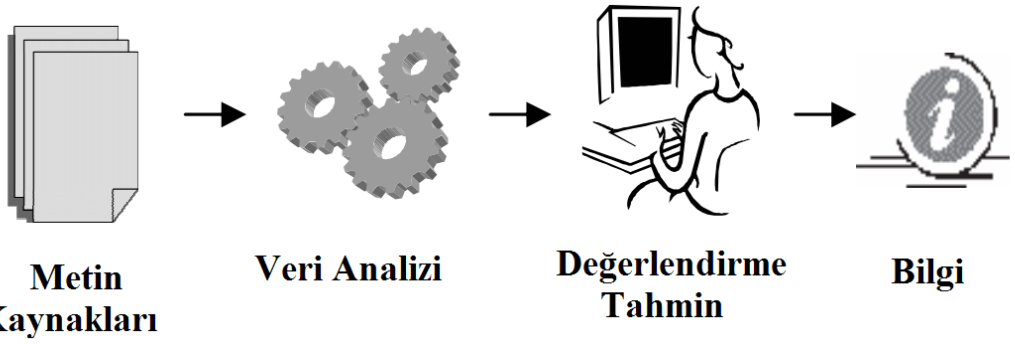
Metin veri madenciliği ya da metin tabanlı veri tabanlarından bilgi keşfi olarak bilinen metin madenciliği, metin içerikli dokümanlardan bilgi çıkarımı ya da gerekli desenlerin ve ilginç verilerin çıkarım sürecini yansıtmaktadır (Tan 1999). Dokümanlardan elde

edilen bilgi çıkarımı ile dokümanlar arası benzerlik ve sınıflandırma işlemlerinin yapılmasına olanak sunmaktadır.

Veri madenciliğinden farklı olarak metin madenciliğinde yapısal olmayan kaynaklar üzerinden işlemler yapılmaktadır. Yapısal veriler, veritabanlarında satır ve sütun olarak kayıt edilebilen, tanımlanabilen ve üzerinde işlem yapılabilen verilerdir, fakat metin madenciliğinde metin dökümanları, resim dosyaları, elektronik postalar gibi yapısal olmayan formatta veriler üzerinden işlem yapılmaktadır. Dokümanlar üzerinden metin madenciliği yapılırken sınıflandırma ya da kümeleme için metinlerin temizlenmesi gerekmektedir. Yani uygulanacak sınıflandırma ya da kümeleme algoritmasının metinlere uygulanması için belirli bir formata dönüştürülmesi gerekmektedir. Dokümanlar kategorilere ayrılmalı ve mümkün olduğunda hatalı yazılmış dokümanlar ayrıştırılmalıdır. Daha sonra dokümanların kategorilerini belirleyen anahtar kelimeler çıkarılmalı ve bir veritabanında saklanmalıdır. Bu kelimelere Joker adı verilmektedir. Kategoriler için kullanılacak kelimeler tespit edildikten sonra doküman içerisinde sık olarak geçip anlamı olmayan kelimeler, veritabanından çıkarılmaktadır. Çünkü bu kelimeler işlem hızını düşürmekle birlikte kategorilerin ayrıştırılmasında da etkili değildir. Gereksiz kelimelerin de çıkarılmasından sonra doküman içerisinde sık geçen fakat aynı kökten gelen kelimelerin kökü bulunur ve sadece kelime kökü veritabanında tutulur. Bütün bu işlemlerden sonra dokümanların vektörel olarak ifade edilmesi işlemi gerçekleşir. Yapısal olmayan verilerin yapısal formata dönüştürülmesi işlemi burada yapılmaktadır. Kelimelerin sayısal formata dönüştürülmesinde dokümanda geçen kelimenin sayısı kullanılmaktadır.

Yapısal olmayan veri analizinde önemli bir yeri olan metin madenciliğinin genel süreci Şekil 2.3'de verilmektedir. Metin kaynaklarının toplanmasından sonra bir metin madenciliği aracı veri analizi için kullanılır. Bu analiz sırasında desen tanıma, ayrıştırma, sözdizimsel ve semantik analizler, kümeleme, şifrelendirme ve diğer çeşitli uygulamaları gibi bir çok alt süreç uygulanır. Veri analizini takiben sonuçlar değerlendirilir ve önceden bilinmeyen bir bilgi ortaya çıkabilir. Erişilen metin bilgisi

veritabanı popülasyonu ve uzlaşma gibi çeşitli yollarda kullanılabilir (Stavrianou *et al.* 2007).



Şekil 2.3. Metin madenciliği süreci

2.6. Literatür Araştırmaları

Metin madenciliği kullanılarak yapılan döküman sınıflandırma çalışmalarında spesifik olarak kurumlarda yer alan birimlere gönderilmek üzere yazılan dilekçelerin kategorizasyon işlemi literatür kapsamında bulunmamaktadır fakat genel olarak doküman sınıflandırma üzerine bir çok çalışma yapılmıştır.

Metin dokümanlarının kategorizasyonu üzerine yapılan bir çalışmada (Antonie and Zaiane 2002) veri madenciliği alanında ilişkisel kural madenciliği kullanarak market sepet analizi yöntemlerinden ödünç alan otomatik metin kategorizasyonu için yeni bir yaklaşım sunulmuştur. Üretm ve budama ile metin tabanlı bir veritabanında en iyi terim ilişki kuralını bulma ve kurallar kullanarak bir metin sınıflandırıcı oluşturma olmak üzere iki major probleme odaklanılmıştır. Geliştirilen metin sınıflandırma yöntemi verimli ve etkili olup iyi bilinen koleksiyonlar üzerinde yapılan deneyler, sınıflandırma performansının iyi olduğunu göstermektedir. Ek olarak sınıflandırmanın yanısıra eğitim ile birlikte kullanıldığında hem hızlı hem de üretilen kuralların insanların okuyabileceği şekilde olması sağlanmıştır.

Doküman sınıflandırma için ağırlık merkezi tabanlı algoritmalarının analizinin yapıldığı bir çalışmada (Han and Karypis 2002) basit ve güçlü performansına rağmen yaygın olarak analizi yapılmayan basit doğrusal zamanlı ağırlık merkezi tabanlı doküman sınıflandırma algoritması üzerine odaklanılmıştır. Geniş oranda yapılan deneyler ağırlık merkezi tabanlı sınıflandırıcının büyük boyutlu veri setleri üzerinde sürekli olarak ve büyük ölçüde Naive Bayes, kNN ve C4.5 gibi algoritmaları geride bıraktığı görülmüştür. Analizler ağırlık merkezi tabanlı şema tarafından kullanılan benzerlik ölçümünün farklı sınıflara ait dokümanların davranışlarını inceleyip dokümanlar arası ortalama benzerliği hesaplayarak yeni dokümanın hangi sınıfa daha yakın olduğunu eşleştirmeye izin vermektedir. Bu eşleştirme farklı yoğunluklar ile sınıflar için dinamik olarak ayarlanmasına izin vermektedir. Ayrıca ağırlık merkezi tabanlı şemanın benzerlik ölçütü farklı sınıflarda terimler arasındaki bağımlılıkları göz önünde tutmaktadır. Yazarlar ağırlık merkezi tabanlı sınıflandırcının bu özelliğini, bağımlılığı göz önünde bulundurmayan diğer sınıflandırıcıları geride bırakma sebebi olarak düşünmüşlerdir.

Mohammed (2007) tarafından yılında yapılan bir çalışmada otomatik doküman sınıflandırma sistemi yapılarak otomatik sınıflandırmanın yapısını etkileyen farklı parametreler ve tasarım kararları araştırılmıştır. Dokümanlara ağırlık verilerek vektörel olarak ifade edilmiş ve sınıflandırıcı olarak yapay sinir ağları uygulanmıştır. Durumlar ağırlık şeması, ağırlıklandırılmış kelimelerin etkisi ve sınıflandırıcının girdi sayısına göre sınıflandırılmıştır. ITF ve TFIDF olmak üzere iki iki farklı ağırlık şeması çıkarılarak sınıflandırma yapılmıştır ve en iyi model %88.2 performans ölçütü ile 8. durum olup ağırlık şeması olarak TFIDF kullanılmıştır. Ayrıca bu modelde başlık terimlerine yüksek ağırlık verilerek yapay sinir ağı tasarımında 1000 girdi ve 100 gizli birimden oluşturulmuştur (Mohammed 2007).

Yıldız vd (2007) tarafından yılında yapılan bir başka çalışmada Türkçe'nin cümle yapısı ile ilgili bilinmeyen bir metnin sınıflandırılması için yeni bir öznitelik çıkarma yöntemi geliştirilmiştir. Öznitelik olarak kelime gövdeleri temel alınan bu çalışmada her metin, sınıf adedi kadar öznitelik ile gösterilir. Oluşturulan veri setinde dokümanda geçen kelime gövdelerinin her sınıfta kullanılma sıklığının toplam değeri hesaplanarak

ağırlandırma yöntemi kullanılmıştır. Kelime gövdesinin bulunması, gövdenin doküman içerisinde bulunmak sıklığının hızlı bir şekilde hesaplanması için Trie adı verilen bir ağaç yapısı kullanılmıştır. Naive Bayes, Destek vektör makinesi, kNN, C 4.5 ve Rastgele Orman isimli sınıflandırma yöntemleri oluşturulan verisetine uygulandığında en yüksek başarı oranı %96.25 ile Naive Bayes yönteminde görülmüştür.

Doküman sınıflandırma için grafik tabanlı bir yaklaşımın tanımlandığı bir başka çalışmada (Jiang *et al.* 2009) grafik gösterimi, standart çantada kelime ya da cümle yaklaşımından daha etkileyici bir doküman kodlaması avantajını sunmuş ve sonuç olarak gelişmiş bir sınıflandırma doğruluğu vermiştir. Doküman kümeleri, sık olan alt grupları çıkarmak için bir ağırlıklı grafik madenciliği algoritmasının uygulandığı grafik kümeleri olarak gösterilmiş ve bu alt gruplar daha sonra sınıflandırma için özellik vektörü üretmek için işlenmiştir. Ağırlıklı alt grafik madencilik sınıflandırma etkinliği ve hesaplama etkinliğini garantiye almak için kullanılmıştır, sadece en önemli alt grafikler çıkarılmıştır ve önerilen bu yaklaşım gerçek dünyadan alınan metinsel veri setleri ile çeşitli popüler sınıflandırma algoritmaları kullanılarak doğrulanmış ve değerlendirilmiştir. Sonuçlar bu yaklaşımın bazı veri setleri üzerinde var olan metin sınıflandırma yöntemlerinden daha iyi performans verebildiğini göstermiştir. Veri seti boyutu arttığında, çıkarılan sık özellikler üzerinde işlem gereklidir.

Çoğu sınıflandırma algoritmaları kelime çantası olarak doküman toplamayı temsil etmektedir. Sriurai (2011) tarafından yılında yapılan bir çalışmada terimler arasındaki anlamsal ilişkilen tanımlanmasında verilen terim kümesinden benzerleri tanımada kelime çantası gösteriminin yetersiz kaldığı görülmüş ve kelimeleri konular halinde kümelemek için bir konu – model yaklaşımı uygulanmıştır. Kelimeler anlamsal özelliklerine göre aynı konu altına atanmıştır. Yazarın ana amacı BOW olarak adlandırılan kelime çantasının ve konu modelinin özellik işleme yöntemlerini karşılaştırmaktır. Ayrıca ki-kare vc bilgi kazancı olmak üzere iki adet özellik seçimi arasında karşılaştırma yapılmıştır. Naive Bayes, Destek Vektör Makineleri (DVM) ve Karar Ağaçları olmak üzere üç adet sınıflandırma yöntemi kullanılmış ve deneysel sonuçlar, dokğmanları temsil etmede bilgi kazancı ile özellik seçme yöntemi

kullanılarak DVM algoritması altında %79 F1 ölçümü oranı ile konu-model yaklaşımının en iyi performansı verdiğini göstermiştir.

Kural azaltma kullanılarak otomatik metin kategorizasyonu ve özetleme yapılan bir başka çalışmada (Devasena and Hemalatha 2012) İşaret Oluşturma, Öznitelik Tanıma ve Kategorizasyon-Özetleme olmak üzere üç aşamada kural azaltma tekniği kullanılarak girdi metninin yapısını türetmek için bir metin analiz edici cihaz geliştirilmiştir. Bu analiz edici cihaz, örnek girdi metinleri ile test edilmiş ve dikkate değer sonuçlar elde edilmiştir. Kapsamlı deneyler, metin sınıflandırma için paremetrelerin seçimi ve yaklaşımın etkinliğini doğrulamıştır.

Özellik vektörlerinin boyutunun sınıflandırma zamanını olumsuz yönde etkilemesinin eksikliği üzere Türkçe gramer kuralları kullanılarak başarı oranından ödün vermeden özellik vektörlerinin boyutlarının nasıl azaltılacağı araştırılmıştır (Tüfekçi vd 2012). Kelime kökleri özellik olarak seçilerek özellik vektörü, kelime frekansının temelinde ağırlıklandırılmıştır. Bu seçim sırasında sınıflandırma için farklı uzunluk ve tip ile kelime köklerinin seçim etkileri araştırılmış ve isim tipinde kelime kökleri ve maksimum uzunluk özellik olarak seçildiği zaman başarı oranının en yüksek seviyed olduğu sonucuna varılmıştır. Bu seçim boyut azaltan diğer yöntemlere uygulandığında özellik vektörünün boyutu %97.46 oranına düşürülmüştür. Azaltılmış özellik vektörü kullanarak genellik daha iyi başarı oranı elde edilen yöntemler Naive Bayes, SVM, C 4.5 ve Rastgele Orman sınıflandırıcıları olmuştur ve en iyi performans %92.73 oranı ile Naive Bayes sınıflandırıcısı kullanılarak elde edilmiştir.

Nithya *et al.* (2012) tarafından yılında yapılan bir çalışmada cümle tabanlı, doküman tabanlı ve korpus tabanlı kavram analizinden oluşan bir madencilik modeli önerilmiştir. sadece dokümanın analiz edilmesinden farklı olarak cümle, doküman ve korpus seviyeleri üzerinde cümle anlamlarına katkı sağlayan terimler analiz edilmiştir. Her yeni doküman için özellik vektörü çıkarımı yapıldıktan sonra özellik seçimi gerçekleştirilir. Özellik seçiminden sonra KNN sınıflandırma yöntemi kullanılarak metin sınıflandırma işlemi gerçekleştirilmiştir.

Jiang *et al.* (2012) tarafından yılında yapılan bir başka çalışmada ise KNN metin kategorizasyonu ve bir geçişlik kümeleme algoritmasının birleşimi ile bir sınıflandırma modeli oluşturulmuş ve metin kategorizasyonu için gelişmiş kNN algoritması önerilmiştir. INNTC adı verilen bu model alt kategorileri ve kısıtlı şartlar ile kategorilerin ilişkilerini yakalamak için kısıtlı bir geçiş kümeleme algoritmasını kullanmaktadır (her küme sadece bir etiket içerir). INNTC ile orijinal metin örnekleri yerine küme vektörlerine dayalı test dokümanları üzerinden sınıflandırma yapılmıştır. Deneysel sonuçlar, önerilen algoritmanın önemli ölçüde metin benzerlik hesaplamasını azaltmış, standart KNN, Naive Bayes ve Destek vektör makineleri sınıflandırıcısından daha iyi performans sağladığını göstermiştir. Buna ek olarak önerilen algoritmadan inşa edilen sınıflandırma modeli artımlı olarak güncellenebilir ve modelin bir çok gerçek dünya uygulamasında büyük ölçeklenebilirliğe sahip olduğu görülmüştür.

Metinsel yorumlar üzerinde duygu analizi ve sınıflandırma yapılan bir başka çalışmada (Mouthami *et al.* 2013) film yorumlarını içeren veriseti üzerinde sınıflandırma doğruluğunu geliştirmek için kullanılan bir kısım konuşma etiketleri ile Duyarlı Bulanık Sınıflandırma isimli yeni bir algoritma önerilmiştir. Bir doküman içerisinde konuşma modelinin bir parçası vektör olarak gösterilmiş ve bu vektörlerin girişleri bir kelimenin bireysel kelimelerine karşılık olarak belirtilmiştir. Sözcük türü bilgisinin duygu ifadelerinde önemli bir gösterge olduğu düşünülmüştür. Duygu sınıflandırma için oluşturulan film yorumları veri seti, önişleme aşamasından geçirildikten sonra simgeleştirici ve yok sayılan kelimeler kaldırılmış, daha sonra dönüşüm işlemi gerçekleşmiştir. Dönüşüm işleminden sonra özellik seçimi, sınıflandırma ve sonuçların değerlendirilmesi işlemi yapılmıştır.

Destek vektör makineleri kullanılarak doküman sınıflandırma yapılan bir başka çalışmada (Fidan 2013) çekirdek fonksiyonunun etkileri ve parametreleri belirlenerek destek vektör makinelerinden ikiden fazla sınıfa ait veriler için üst uzaya aktarılma işlemi yapılmıştır. Lasvm algoritmasının eşli çekirdek yöntemine uyarlanmış şekilde çalışması sağlanmıştır. Optimum karar sınırı için parametreler belirlendikten sonra veriler eğitim ve test kümelerine ayrılarak farklı kombinasyonlarda deneyler yapılmıştır.

Destek vektör makinelerinin tahmin edilen sınıflandırması ve doğru sınıflandırma oranı ROC eğrisi altında kalan alana göre değerlendirilmiştir. Deneysel sonuçlar incelendiğinde çevrimiçi eşli sınıflandırma yönteminin iki veya çok sınıflı sınıflandırma yöntemlerine alternatif olabileceğini göstermiştir. Ayrıca verilerin yüksek boyutlu uzaylarda olmasından dolayı doğrusal eşli çekirdeklerin gauss eşli çekirdeklere göre daha iyi sonuç verdiği görülmüştür.

Levent Diri (2014) tarafından yılında yapılan bir çalışmada Türkçe gazete köşe yazarlarının yazarlık özelliklerinin yapay sinir ağları yöntemi ile elde edilmesi işlemi gerçekleştirilmiştir. Özellik çıkarma aşamasında Zemberek Kütüphanesi kullanılarak dokümandaki metinlerde sözcük ayrıştırma işlemi yapılmıştır. Cinsiyete göre, aynı ve farklı kategorilerde yazma işlemi olmak üzere köşe yazarları için üç farklı veri seti oluşturulmuştur. Tasarlanan yazar tanıma uygulaması ile yapay sinir ağları kullanılarak köşe yazısı verilen bir metnin hangi yazara ait olduğu kullanıcıya bildirilmiş ve bu uygulama aracılığı ile farklı sayıda sınıf için oluşturulmuş olan öznitelik kümeleri arff dosya formatında saklanabilmiştir. Böylece, veri madenciliğinde yaygın olarak kullanılan Java destekli WEKA uygulamasında da veri setlerinin kullanımı sağlanmıştır. Ayrıca uygulamada WEKA'da olduğu gibi yapay sinir ağının parametlerini değiştirme işlemi olmakla birlikte yazar ekleme, yazarlara doküman ekleme, yalnızca belirli öznitelikleri seçebilme gibi birçok özellik kullanıcıya sunulmuştur.

Bouzaieni *et al.* (2015), olasılıksal grafik modeli kullanarak dipnot uzantısı ve doküman sınıflandırma için bir model önermişlerdir. Bu model Gaussian'ın karışımları ve çok terimli dağıtımların bir karışımına dayanır. Manuel dipnotun maliyetini azaltmak için döngüde kullanıcı ekleyerek iteratif olarak öğrenme geliştirmişlerdir. Daha kesin olarak modeli öğrenmek için görsel ve metinsel karakteristikleri birleştirerek kullanıcı geri bildirim entegrasyonunun dipnot adımını geliştirdiğini göstermişlerdir. İlk olarak 2 anahtar kelime ile her öğrenme dokümanını açıklamışlar ve kullanıcı geri bildirimi ve iteratif bir dipnot uzantısı kullanarak her doküman için 5 anahtar kelimeye erişmeye çalışmışlardır. Bu amaç için üçüncü bir anahtar kelime otomatik olarak 100 öğrenme dokümanı üzerinde genişletilmiştir. 100 anahtar kelime üzerinde 19 dipnotun doğru

olduğu fakat geri kalan 81 dipnotun kullanıcı değişikliğine ihtiyacı olduğu görülmüştür. Bu süreç dördüncüve beşinci anahtar kelimeler in eklenmesi için iki kez tekrar edilmiştir. Kullanıcı düzeltmelerinin sayısı sırasıyla 72 ve 63 olmuştur ve böylece 200 kelime ile başlayan ve 216 (81+72+63) kullanıcı düzeltmesi uygulayarak 500 anahtar kelime elde edilmiştir ve manuel efor ile karşılaştırıldığında %16,8 daha fazla kazanılmıştır.

Silva and Dornales (2015) tarafından yapılan bir çalışmada artan veri ile zorlaşan sınıflandırma işlemi için dokümandan konsept örnekleri türeten ve konsept genellemesi için bir açık alan bilgi tabanını kullanan bir yaklaşım geliştirilmiştir. Bu yaklaşıma Bilgi Tabanlı Doküman Sınıflandırma (KDC) adı verilmiştir. KDC, bir dokümandan alınan örnekleri genelleştirip daha geniş konsept setleri eşitsizlik değerine göre sıralayıp daha sonra en iyi yerleştirilmiş konsept dokümanın sınıf etiketi olarak belirlemektedir. Gerçek dünyadan alınan veriler ile hazırlanan veriseti üzerinde deneyler uygulandığında bu yaklaşımın ontoloji ya da eğitim safhasına gerek olmadan ve bir sınıflandırma modeli kullanmadan makine öğrenmesi yöntemlerine yakın performans ile doküman sınıflandırabildiği görülmüştür.

Akademik dokümanların otomatik olarak sınıflandırıldığı bir çalışmada (Nunez and Ramos 2015) lisans öğrencilerinin final projelerinin otomatik sınıflandırılması için metin madenciliği yöntemi kullanılmıştır. Dört profosyonel kategori içeren dokümanlardan oluşturulan veri seti, farklı indeks metrikleri ile vektör alan modeli vasıtasıyla temsil edilmiştir. Aynı zamanda boyut indirgeme için çok sayıda yöntem, kelime alanı üzerine uygulanmıştır. Sınıflandırma modelini kurmak için K-en yakın komşu (KNN) algoritması uygulanmıştır. 10 katlamalı çapraz doğrulama yöntemi ile %82 oranında doğru tahmine ulaşılmış fakat dokümanlarda iki kategoriye kadar disiplinler arasının dikkate alındığı bir öneri ile %95 doğruluğa ulaşılmıştır. Bu sınıflandırıcı, tavsiye edilen alanlara ait öğretmenlerden gelen atamayı uygulayan yorumcuların otomatik ataması için bir uygulamaya entegre edilmiştir.

Qureshi *et al.* (2015) tarafından yılında yapılan çalışmada Apriori algortiması kullanılarak metin dokümanlarından kuralları çıkarmak için etkili bir yaklaşım önerilmiş ve ayrıca önerilen etkili kural budama yöntemi ile gereksiz kurallar azaltılarak uygun boyutta sınıflandırıcı, deney veri setlerine uygulanmıştır. Eğitim amaçlı, Boyut azaltma yöntemi ve dokümanda özellik sayısını azaltmak için ağırlık kavramına adapte olan yeni bir AC (ilişkisel sınıflandırma) modeli önerilmiştir. AC yöntemi önişleme, kural üretme, sınıflandırıcı yapılandırma, tahmin ve test aşaması olmak üzere 5 aşamadan oluşturulmuştur. Önişleme aşamasında işaret tespiti, boyut azaltma ve terim ağırlıklandırma işlemleri yapılmıştır. Kural üretiminde ise destek ve güven değerleri hesaplanmıştır. Metin dokümanlarının bulunduğu büyük veri setlerinde yapılan test işleminde, normal sınıflandırma yöntemleri ve diğer ilişkisel sınıflandırma yöntemlerine göre karşılaştırıldığında %89.64 F ölçüm değeri ile önerilen algoritmanın daha yüksek doğruluk elde ettiği görülmüştür.

Toplu özellik seçimi ve veri madenciliği yöntemleri kullanılarak metin sınıflandırma yapılan bir çalışmada (Shravankumar and Ravi 2015) çeşitli Web adreslerinden toplanan TechTC veri seti sınıflandırılmıştır. Model inşa zamanını büyük oranda indiren Gini indeksi, ki-kare, t-istatistik, korelasyon özellik seçimi yöntemleri kullanılmıştır. Olasılıksal sinir ağı, veri işlemenin grup yöntemi, çok katmanlı algılayıcı, gibi çeşitli sinir ağı modelleri literatürde uygulanan diğer yöntemler ile karşılaştırıldığında daha yüksek doğruluklar vermiştir.

Parlak ve Uysal (2015) tarafından yılında yapılan bir başka çalışmada ise içerisinde tıbbi metinleri barındıran MEDLINE isimli veritabanından dokümanların bir alt kümesi alınarak tek etikete sahip çok sınıflı bir veri seti oluşturulmuştur. Çalışmada sayısı en yüksek çıkan dokümana ait 10 hastalık kullanılmıştır. Bayes ağı, C4.5 ve Rastgele Orman olmak üzere üç farklı sınıflandırma algoritması ile bu veri setinin testi yapılmıştır. Ayrıca kök bulma işleminin yapıldığı ve yapılmadığı durum olmak üzere iki farklı durum için değerlendirme yapılmıştır. Deneysel sonuçlar incelendiğinde Bayes ağının performansı en yüksek sınıflandırıcı olduğu görülmüş ve en yüksek başarı oranı kök bulma işleminin yapılmadığı durumda elde ediilmiştir.

Veri sınıflandırma kullanarak kitap inceleme madenciliğine etkili bir yaklaşım sunulan bir başka çalılşmada (Harvinder *et al.* 2015) kitapların çevrimiçi inceleme verilerini çıkarmak (inceleme sonuçlarını geliştirme amaçlı) için multinominal Naive Bayes sınıflandırma algoritması kullanılarak fikir analizi ile birlikte TF-IDF metodu birleştirilerek hibrit bir yöntem sunulmuştur. TF-IDF yöntemi bir dokümandaki kelimenin ağırlığını hesaplamak için ağırlıklı yöntemi kullanır ve fikir analizi belirli bir ürün hakkında polarite verir. Daha iyi tavsiyeler için tavsiye verici sistemler tarafından kullanılabilir olabilen sonuçların etkinliğini doğaçlamak için her iki yöntem de birleştirilmiştir.

Türkçe dokümanlar için n-gram tabanlı yaklaşım kullanılan bir çalışmada (Dogan ve Diri 2015) dokümanın sınıfı, dokümanın hangi yazara ait olduğu ve yazarın cinsiyeti n-gram modeli kullanılarak belirlenmiştir. N değeri 1, 2, 3 ve 4 olarak verilerek üç adet veri setinde altı adet öznitelik vektörü oluşturulmuştur. Naive Bayes, Destek Vektör Makineleri, Rastgele Orman, kNN gibi sınıflandırma yöntemleri ile birlikte Ng-ind adlı bir yöntem önerilerek bu yöntem ile diğer yöntemler veri setlerinin başarı oranlarına göre karşılaştırılmıştır. Ng-ind yönteminde her dokümanın n-gram profili hesaplanarak profiller arasındaki benzerlik oranları ele alınmıştır. Kategori profili ile doküman profili arasında benzerlik değeri mutlak fark alınarak hesaplanmıştır. Elde edilen deneysel sonuçlara göre Ng-ind yönteminin cinsiyet ve doküman türünü belirlemede diğer geleneksel sınıflandırma yöntemlerine göre daha iyi performans gösterdiği görülmüştür. Ng-ind yönteminin başarı oran %90'ın üzerinde iken diğer yöntemler üç farklı veri setine göre genellikle %80'in altında başarı oranı vermiştir. Ayrıca geleneksel sınıflandırma yöntemlerinin birleştirilmesi ile elde edilen sonuçlar ve Ng-ind yöntemi karşılaştırıldığında birinci veri setinde %1'lik bir fark ile birleştirilmiş yöntem daha iyi sonuç verirken üçüncü veri setinde %5'lik bir oran ile Ng-ind yöntemi daha başarılı olmuştur.

Türkçe metinlerin sınıflandırılması üzerine yapılan bir başka çalışmada (Çobanoğlu 2015) önişleme adımları ve sınıflandırma yöntemleri arasındaki en iyi kombinasyonu belirlemek hedef alınmıştır ve öznitelik oluşturma için kelimelerin dokümanda olduğu

hali, kökleri, bi-gram ve tri-gram modelleri kullanılmıştır. Bu öznitelik kümelerine farklı ağırlıklandırma ve sınıflandırma yöntemleri kullanılarak 216 adet deneysel sonuç elde edilmiştir. Elde edilen sonuçlara göre %95 oranı ile en yüksek performansı C4.5 yöntemi vermiştir. İkinci en yüksek performansı veren algoritma DVM olmuştur. Naive Bayes ise bu algoritmalar içerisinde en düşük sonucu vermiştir.

Kılıç vd (2015) tarafından yılında yapılan bir çalışmada metin madenciliğinde kullanılan yöntemlerden terim frekansı-tersi doküman frekansı yöntemi (term frequency-inverse document frequency TF-IDF) ile Ters Eksik Sınıf Doküman Frekansı- TESDF ve Sınıflar arası doküman frekansı- SADF olmak üzere iki adet metin ağırlıklandırma yöntemi önerilmiştir. Çalışmada metin sınıflandırmada fiillerin yer almadığı yeni bir yöntem önerilerek önişleme aşamasında farklı bir yöntem uygulanmıştır. Önerilen yöntemlerden TF*SADF C4.5 algoritmasında en iyi performansı göstermiş olup TF-IDF'den daha etkili olmuştur. Ayrıca Naive Bayes algoritmasında RF*TESDFv%88.58 başarı oranı ve DVM algoritmasında yine TF*SADF %87.07 ile en yüksek performansı elde etmiştir. Önişleme aşamasında metin fiillerinin atılması da başarı oranını etkilememiş olup işlenecek veriyi yaklaşık %17 oranında azaltmıştır.

Metin madeciliği ile doküman sınıflandırma dışında benzerlik hesaplamaları yapılarak dokümanlar arası benzerlik oranı tespitleri de yapılmaktadır. Döven tarafından 2013 yılında yapılan bir çalışmada sadece iki doküman arası benzerliğin bulunması ile birlikte bir masaüstü uygulaması aracılığı ile kullanıcı tarafından adet seçimi yapılarak istenilen sayıdaki dokümanların arasındaki benzerlikler bulunmuştur. Masaüstü uygulaması aracılığı ile yüklenen doküman içerisinde bulunan cümleler diğer dokümanlar içerisinde aranarak tüm cümleler ile dokümanda tek tek ele alınan her cümle arasında benzerlik hesaplaması yapılmıştır. Cümlelerin benzerlik hesapları sayısal sonuçlar şeklinde gösterilerek kullanıcının hangi yöntemin daha iyi sonuç verdiğini görmesi sağlanmıştır. Yapılan deneyler sonucunda kosinüs ve jaccard algoritmaları en başarılı yöntemler olarak tespit edilmiştir.

Intarapaiboon (2016) tarafından yılında bir çalışmada sezgisel bulanık kümeler (IFS) açısından dokümanların nasıl gösterileceği ve sezgisel bulanık küme tabanlı gösterimden her sınıfın deseninin nasıl elde edileceği konusunda yaşanan iki zorlukla başa çıkmak için bazı benzerlik ölçümleri kullanılarak metin sınıflandırma için yeni bir çerçeve önerilmiştir. Benzerlik ölçümü doküman için bir IFS ve sınıf deseni arasındaki benzerlik derecesini belirlemek için kullanılmıştır. Öneri iki farklı veri seti üzerinde uygulanmış ve sonuçlar karar ağacı, KNN, Naive Bayes ve destek vektör makineleri ile karşılaştırıldığında tatmin edici sonuçlar vermiştir.

3. MATERYAL ve YÖNTEM

Bu tez çalışmasında daha önce kategorize edilmiş 5 farklı birime ait 225 dilekçe örneği ele alınarak kelime analizi yapılmıştır. Bu dilekçe örnekleri Atatürk Üniversitesi Açık Öğretim Fakültesi'nden elde edilerek birimlere ayrılmıştır. Birimlere ayrılmış dilekçeler MODI (Anonymous 2016a) ile bilgisayar ortamına aktarılmış ve Microsoft Office Interop Word (Anonymous 2016b) kütüphanesi kullanılarak hatalı kelimeler tespit edilmiştir. Hatalı kelimelerin düzeltilmesi Levenshtein Distance Algoritması kullanılarak yapılmıştır. Levenshtein Distance Algoritması için gerekli olan kelime kütüphanesi Microsoft Office Interop Word ile daha önceden taranmış dilekçelerdeki hatalı olmayan tüm kelimelerden oluşturulmuştur. Kelime köklerinin bulunması için Zemberek (Anonim 2016b) adlı kütüphaneden yararlanılmıştır. Her birim için kullanılan kelimeler kullanım frekansına göre kaydedilmiştir. Özellik çıkarımı yapmak için birimlerde kullanım frekansı yüksek olan kelimeler baz alınmıştır. Frekansı yüksek olan kelimeler için de kullanım oranı her birimde aynı olabilecek olan kelimeler elenmiştir. Son adımda bu anahtar kelimelerin birimlerdeki frekansları kaydedilerek veri seti oluşturulmuştur. Veri seti, Weka (Anonymus 2016d) adlı veri madenciliği yöntemlerini içerisinde bulunduran bir program aracılığı ile birçok algoritma üzerinden test edilmiştir. En yüksek benzerlik oranı Naive Bayes Multinominal algoritmasında bulunmuş ve yapılacak sistemin bu algoritma kullanılarak tasarlanması hedeflenmiştir. Sistemede tarama, optik karakter tanıma, hatalı kelimelerin tespiti ve düzeltilmesi, kök bulma ve Naive Bayes Multinominal algoritması ile dilekçenin yönlendirileceği birimin belirlenmesi ve veritabanına kaydedilmesi için masaüstü uygulaması tercih edilmiştir. Birim personelinin gelen dilekçeleri takibi ve işlem yapması için web uygulaması tercih edilmiştir. Ayrıca gelen dilekçelerin işlem durumu, bekleme süresi gibi özelliklerine göre filtreleme yapılabilecek bir web arayüzü tasarlandı.

Bu çalışmada yapılan işlemler veri toplama, başarım testi ve uygulama aşaması olmak üzere üç adımda gerçekleştirilmiştir. Aşağıda alt başlıklar halinde bu adımlar sıralı olarak açıklanmıştır.

3.1. Veri Toplama

Veri toplama aşamasında her birim için alınan örnek dilekçeler öğrenme aşaması için toplu olarak işleme alınmıştır. Dilekçelerin içerdiği kelimelerin her birim için frekansı bulunmuş ve bu verilere göre veri seti oluşturulmuştur. Bu aşamadan sonra veri seti benzerlik tespiti için Weka programı ile test edilmiştir.

3.1.1. Dilekçelerin toplu olarak taranması

Sistemde dilekçelerin bir tarayıcı ile taranarak sisteme kaydedilmesi için açık kaynak bir kütüphane olan Saraff.Twain.NET (Anonymous, 2016c) kütüphanesi kullanılmıştır. Twain günümüzde birçok tarayıcıda mevcut olan bir sürücüdür. Twain sürücüsü olan bir tarayıcı kullanıldığında bu sürücü, tarayıcı ve program arasında bir köprü vazifesi görür. Bu sayede uygulama ile tarayıcıya bağlanmak, taramayı başlatmak ve taranan resme ulaşmak mümkündür.

3.1.2. Kelime kütüphanesi oluşturulması

Kelime düzeltme için kullanılacak kütüphaneyi oluşturmak için taranan 225 örnek dilekçenin içerisindeki tüm hatasız sözcükler alınmıştır. Bu işlemde taramadan sonra MODI ile optik karakter tanıma yapılmış ve dilekçeden elde edilen metin içerisinde Microsoft Office Interop Word ile hatasız olan kelimleler tespit edilmiştir. Microsoft Office Insterop Word ile tüm kelimelerden hatalı olan kelimeler çıkarılmış ve Şekil 3.1 de doğru olan kelimelerin yeni bir diziye aktarıldığı kod bloğu gösterilmiştir. Bu işlemin ardından kelimeler veritabanına kaydedilmiştir. Toplam 2182 tekrarsız kelime bulunmuştur. Bu kelimeler ile yeni gelen dilekçelerdeki hatalı sözcüklerin düzeltilmesi için kelime kütüphanesi oluşturulmuştur.

```
_Document doc = app.Documents.Add(ref template,ref newTemplate,ref documentType,ref visible);
doc.Words.First.InsertBefore(tBox.Text);
Microsoft.Office.Interop.Word.Words kelime = doc.Words;
Microsoft.Office.Interop.Word.ProofreadingErrors hata = doc.SpellingErrors;

string[] dogrukelimeler = new string[kelime.Count-hata.Count];
int k = 0;
for (int i = 1; i < kelime.Count; i++)
{
    for (int j = 1; j < hata.Count; j++)
    {
        if (kelime[i].Text != hata[j].Text )
        {
            dogrukelimeler[k] = kelime[i].Text;
            k++;
        }
    }
}
```

Şekil 3.1. Microsoft Interopt Word ile hatasız kelimelerin tespit edilmesi

Zemberek ile kelime düzeltme de yapılabilmektedir ancak kelimelerin benzerlik oranları aynı olan kelime sayısı çok olduğunda başarı oranı daha düşük çıkmaktadır. Zemberek kütüphanesinin bu dezavantajından dolayı önceden oluşturulan kelime kütüphanesinde Leveinshtein Algoritması kullanılarak hatalı kelimelerin doğru bir şekilde düzeltilmesi amaçlanmıştır. Dilekçe için yazılabilecek anlamlı kelimeler mevcut kütüphanede saklanmıştır. Fakat Zemberek kütüphanesi genel olarak tüm kelimeleri barındırdığı için benzerlik oranı aynı olan birçok kelime bulmaktadır. Bu da düzeltme için doğru sonuç verme olasılığını düşürmüştür. Şekil 3.2'de Leveinshtein Algoritması ve Zemberek ile kelime benzerliği yapılan birkaç örnek gösterilmiştir.

	Leveinshtein	**Zemberek**
snavı	sınavı	sınavı
ayıt	kayıt	acıt
our	onur	olur
tercilerim	tercihlerim	hercilerim

Şekil 3.2. Zemberek ve Leveinshtein ile kelime önerileri

Kelime kütüphanesi oluşturulduktan sonra veri setini oluşturmak amacıyla kelime köklerinin alınma aşamasında Zemberek kütüphanesi kullanılmıştır. Şekil 3.3'de Zemberek kütüphanesi ile kelime kökü alma işleminin kod bloğu gösterilmiştir.

```
static string kokbulma(string kelime)
{
    Zemberek zmbrk = new Zemberek(new TurkiyeTurkcesi());
    string kok = "";

    if (zmbrk.kelimeDenetle(kelime))
    {
        kok = zmbrk.kelimeCozumle(kelime)[0].kok().icerik();
    }

    return kok;
}
```

Şekil 3.3. Zemberek kütüphanesi ile kelime kökü bulma

3.1.3. Veri setinin oluşturulması

Her birime ait farklı dilekçe örnekleri için daha önceden veritabanına kaydedilen kelimeler üzerinden ön işleme yapılmıştır. Bu işlem kategorilere göre farklılık gösterebilecek kelimelerin seçilmesi aşamasıdır. Seçim işlemi yapılırken kullanım frekansı yüksek olan kelimeler ilk aşamada ele alınmıştır. Sonrasında kategorizasyon için herhangi bir anlam ifade etmeyen bağlaçlar, özel isimler ve numaralar elenmiştir. Bu işlemden sonra frekansları her birim için birbirine yakın olan kelimeler elenmiştir. Yapılan işlemler sonucunda 75 adet anahtar kelime elde edilmiştir. Bu anahtar kelimeler kullanım frekansları ile birlikte veri setini oluşturmuştur. Çizelge 3.1'de anahtar kelimelerin bir bölümünün kategorize edilmiş dilekçeler içindeki kullanım sayıları gösterilmiştir.

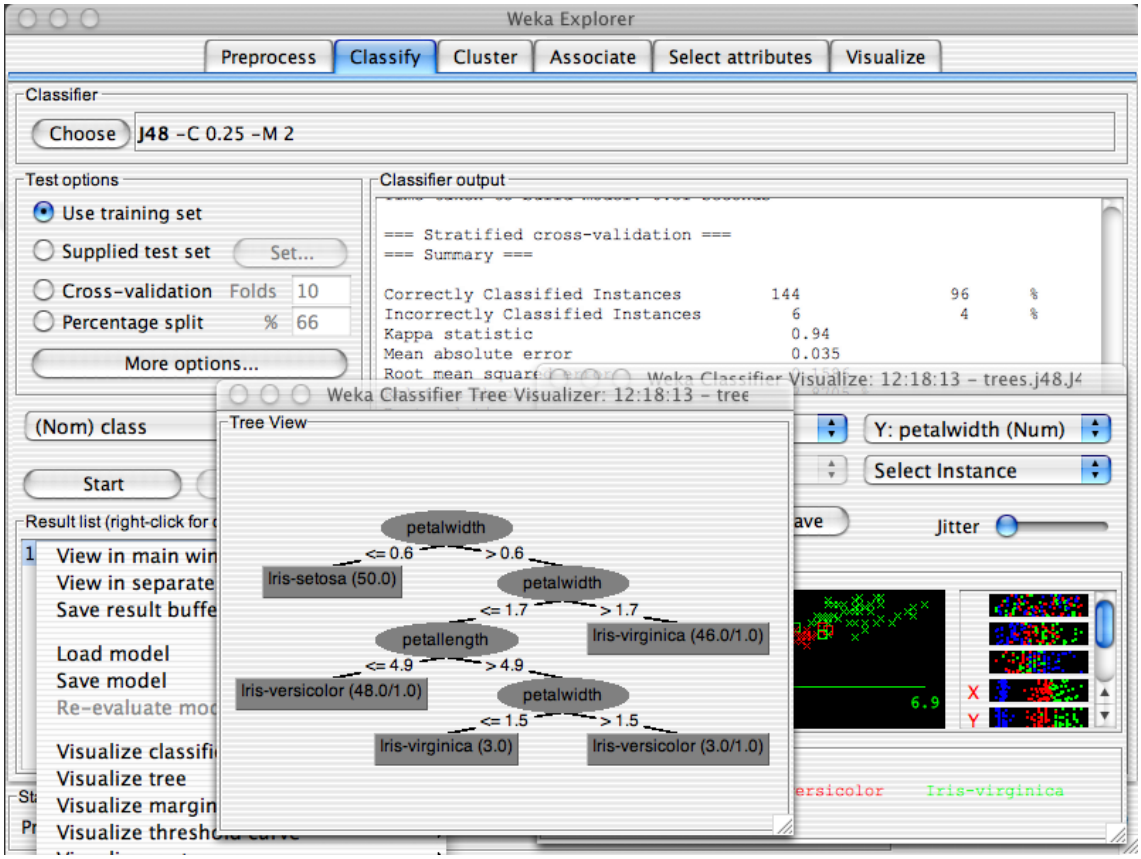
Çizelge 3.1. Anahtar kelimelerin her bir dilekçede kullanım sayısı

	adet	askerlik	banka	başvuru	belge	bina	blok
belgetalep	0	0	0	0	1	0	0
belgetalep	1	0	0	0	1	0	0
belgetalep	0	0	0	0	6	0	0
belgetalep	0	0	0	1	2	0	0
derskayit	0	0	0	0	0	0	0
derskayit	0	0	0	0	0	0	0
derskayit	0	0	0	1	0	0	0
harc	8	0	8	0	1	0	0
harc	0	0	1	0	0	1	0
harc	0	0	0	0	0	1	0
harc	0	0	0	0	1	0	0
harc	0	0	2	0	2	0	0
harc	1	0	1	0	0	0	0
harc	2	0	0	0	2	0	0
sinav	1	0	0	0	0	0	0
sinav	0	0	0	0	0	0	0
sinav	0	0	0	0	0	0	0
sinav	0	0	0	1	0	0	0
staj	0	0	0	0	0	0	7
staj	0	0	0	0	0	0	3
staj	0	0	0	0	0	0	1
staj	0	0	0	0	0	0	1

3.2. Başarım Testi

Çizelge 3.1'de bir kısmı verilen veri setinde gösterildiği gibi dilekçelerde bulunan kelimelerin frekans analizi yapıldıktan sonra veri seti eğitim ve test veri seti olarak ayrılmış ve eğitim veri seti kullanılarak denetlenen sınıflandırma yöntemlerinin test verileri üzerinde başarı oranlarını incelemek hedeflenmiştir. Çalışmada amaç, başarım oranı en yüksek sınıflandırma yönteminin belirlenerek o sınıflandırma yöntemi ile gelen dilekçelerin ait oldukları birimlere elektronik ortamdan iletilmesidir. Bu yüzden sınıflandırmada kullanılan popüler sınıflandırma yöntemlerinin veri setine kolayca uygulanması açısından çoğu sınıflandırma yöntemini içerisinde bulunduran Java programlama dili destekli platform bağımsız WEKA (Waikato Environment for Knowledge Analysis) adlı bir yazılım kullanılmıştır (Anonymus 2016d). Bu yazılım

Waikato Üniversitesinde geliştirilmiş olup herhangi bir kodlama gerektirmeden çeşitli sınıflandırma, bölütleme ve ilişkilendirme olmak üzere üç temel veri madenciliği işlemi ile birlikte veri setleri üzerinde veri önişleme ve görselleme işlemlerini yapmayı sağlar. Şekil 3.4'de WEKA yazılımına ait örnek bir sınıflandırma paneli verilmektedir.



Şekil 3.4. WEKA yazılımına ait sınıflandırma paneli

Dilekçelerin analizi için oluşturulan veri seti, eğitim ve test verilerine 10 katlamalı çapraz doğrulama yöntemi ile ayrılarak Weka yazılımında bulunan Knn, Naive Bayes (NB), Multinominal Naive Bayes (MNB), Destek Vektör Makineleri (DVM), Geri Yayılım (GYA) algoritmaları ile test edilmiştir. Elde edilen sonuçlar doğrultusunda benzerlik oranı en yüksek olan Multinominal Naive Bayes yönteminin tasarlanan sistemde kullanılmasına karar verilmiştir.

3.3. Özellik Çıkarımı

Özellik çıkarımı için Naive Bayes Multinominal algoritması kullanılmuştur. Multinominal Naive Bayes (MNB) algoritması, dokümanlar içerisinde kelimelerin frekans bilgisini kullanır (McCallum and Nigam 1998). Bu yüzden MNB algoritması, dokümanlara uygulanan metin madenciliğinde çıkarılan özniteliklerin frekanslarının yer aldığı veri setlerinde başarılı sonuçlar vermektedir.

Verilen bir doküman için sınıf olasılıkları hesaplandığında C sınıf etiketlerini, N kullanılan veri setindeki kelime boyutunu ifade etmektedir. MNB, Bayes kuralını kullanarak en yüksek olasılığa sahip $\Pr(c|t_i)$ bir sınıfa bir t_i test dokümanı atar ve bu olasılık aşağıdaki gibi hesaplanır (Kibriya *et al.* 2004):

$$\Pr(c|t_i) = \frac{\Pr(c)\,\Pr(t_i|c)}{\Pr(t_i)}, \quad c \in C \tag{1}$$

Öncelikli sınıf $\Pr(c)$, c sınıfına ait doküman sayılarının dokümanların toplam sayısına bölümü ile tahmin edilebilir. $\Pr(t_i|c)$, c sınıfında t_i gibi bir dokümanın elde edilme olasılığını ifade etmektedir ve aşağıdaki gibi hesaplanmaktadır (Kibriya *et al.* 2004):

$$\Pr(t_i|c)=(\sum_n f_{ni})! \quad \prod_n \frac{\Pr(w_n|c)^{f_{ni}}}{f_{ni}!} \tag{2}$$

(2) numaralı eşitlikte bulunan f_{ni} , t_i test dokümanındaki kelime sayısıdır ve $\Pr(w_n|c)$, verilen c sınıfında n kelimenin olasılığıdır. Bir sonraki olasılık eğitim dokümanından aşağıdaki denklem kullanılarak tahmin edilir (Kibriya *et al.* 2004):

$$\Pr(w_n|c) = \frac{1+F_{nc}}{N+\sum_{x=1}^{N} F_{xc}} \tag{3}$$

F_{xc} c sınıfına ait tüm eğitim dokümanlarında x kelimesinin sayısını gösterir ve Laplace tahminleyici sıfır frekans problemini önlemek için kullanılmaktadır. Normalizasyon

faktörü Pr(t_i) eşitlik (1)'de aşağıdaki denklem kullanılarak hesaplanmaktadır (Kibriya *et al.* 2004):

$$\Pr(t_i)=\sum_{k=1}^{|C|} \Pr(k) \Pr(t_i|k) \quad (4)$$

Hesaplama açısından maliyetli olan eşitlik (2)'deki $(\sum_n f_{ni})!$ ve $\prod_n f_{ni}!$ terimleri sonuçlarda herhangi bir değişiklik olmadan silinebilir, çünkü hiçbiri c sınıfına bağlı değildir. Sadeleştirme işleminden sonra eşitlik (2) aşağıdaki gibi yazılabilir (Kibriya *et al.* 2004):

$$\Pr(t_i|c)= a \prod_n \Pr(w_n|c)^{f_{ni}} \quad (5)$$

Normalizasyon aşamasından dolayı a sabiti ihmal edilir.

Bu aşamada anahtar kelimelerin her bir kategoriye göre ağırlıkları bulunmuştur. Örnek olarak adet kelimesi örnek dilekçeler içerisinde belge talep birimi için 4, harç birimi için 12, sınav birimi için 3 ve staj birimi için de 2 defa kullanılmıştır. Ders kayıt birimi için ise bu kelime hiç kullanılmamıştır. Bu kelimenin belge talep birimi için ağırlığının bulunması için formüle göre kelimenin birimdeki kullanım sayısına 1 eklenmiş ve ardından birimde kullanılan tüm tekrarlı kelime sayısı ile tüm birimlerde kullanılan tekrarsız kelime sayısına bölünmüştür. Aynı şekilde diğer birimler için ağırlığı da bulunmuştur. Adet kelimesi için (4+1) / (385+75) işlemi ile belge talep birimindeki ağırlığı 0,013043478 olarak bulunmuştur. Bu işlem mevcut olan 75 anahtar kelime içinde uygulanmış ve veritabanına kaydedilmiştir. Çizelge 3.2'de anahtar kelimelerin bir bölümü için her birimde bulunan ağırlıkları gösterilmiştir. Birimlerin genel ağırlıkları da hesaplanarak veritabanında tutulmuştur. Her taramada bu işlemi uygulamak yerine veritabanına kaydedip uygulamanın daha hızlı çalışması sağlanmıştır.

Çizelge 3.2. Anahtar kelimelerin her bir birim için ağırlığı

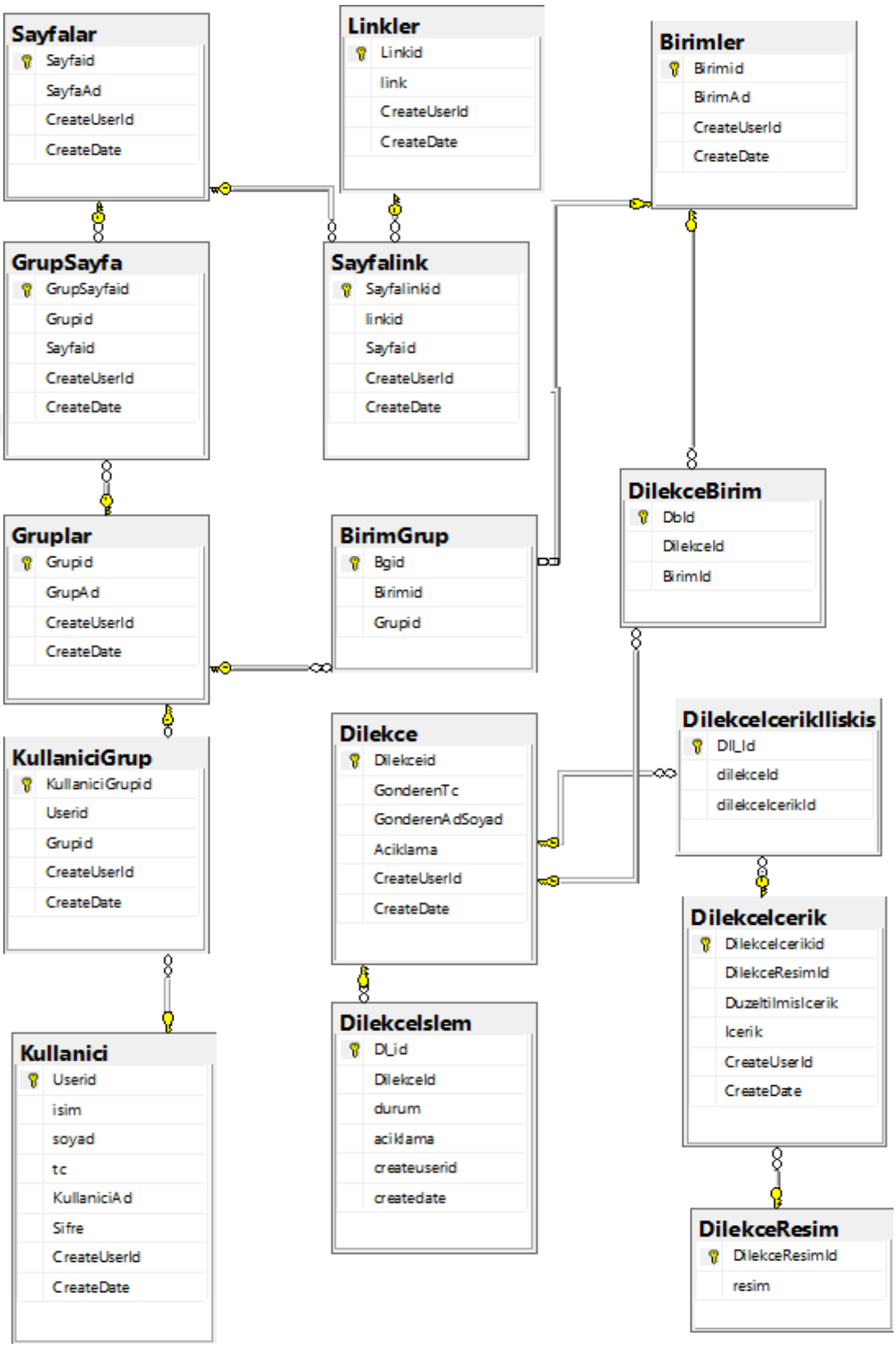
kelime	**belgetalep**	**derskayit**	**harc**	**sinav**	**staj**
muafiyet	0,006521739	0,018494055	0,003514938	0,0025	0,010962241
giriş	0,006521739	0,017173052	0,003514938	0,03	0,002436054
evlilik	0,006521739	0,001321004	0,001757469	0,0025	0,001218027
boşanma	0,008695652	0,001321004	0,001757469	0,0025	0,001218027
banka	0,008695652	0,005284016	0,063268893	0,0025	0,002436054
süre	0,008695652	0,002642008	0,005272408	0,0175	0,002436054
not	0,008695652	0,010568032	0,008787346	0,035	0,014616322
lise	0,008695652	0,001321004	0,001757469	0,0025	0,001218027
askerlik	0,010869565	0,001321004	0,001757469	0,0025	0,001218027
faks	0,010869565	0,010568032	0,003514938	0,01	0,020706456
evrak	0,013043478	0,001321004	0,014059754	0,0125	0,00365408
adet	0,013043478	0,001321004	0,0228471	0,01	0,00365408

3.4. Uygulama

Sistemde dilekçelerin sisteme girişi ve yönlendirilmesi aşaması için masaüstü uygulaması ve birim personelleri için web uygulaması oluşturulmuştur. Bu aşamalara aşağıda sıralı olarak anlatılmıştır.

3.4.1. Veritabanı tasarımı

Sistemde kullanılacak olan veritabanı Microsoft SQL Server ile tasarlanmıştır. Personel için kullanıcı bilgileri, log kayıtları, dilekçe bilgileri ve ilişkiler veritabanında oluşturulmuştur. Şekil 3.5'te oluşturulan veritabanı için tablo bilgileri ve ilişkileri gösterilmiştir. Arayüzlerdeki işlemler için veritabanında Stored Procedured kullanılmuştır. Stored Procedured kullanılmasının sebebi hızlı olması ve SQL Injection saldırılarını engellemesidir.



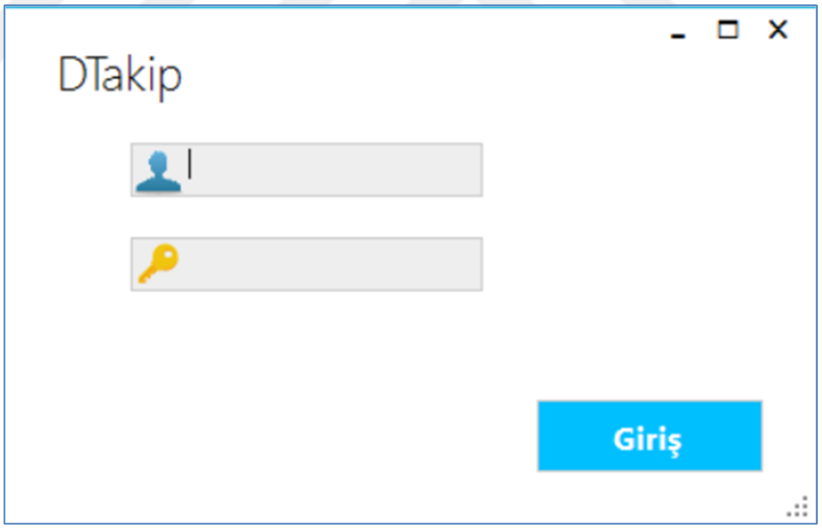
Şekil 3.5. Veritabanı Tasarımı

3.4.2. Masaüstü uygulaması

Dilekçenin fiziksel olarak alındığı birim için masaüstü uygulaması yapılmıştır. Dilekçenin taranması, optik karakter tanıma, kelime düzeltme, kök bulma ve Naive Bayes algoritması ile sınıflandırma yapma işlemleri bu uygula ile gerçekleştirilmiştir. Masaüstü uygulamasının seçilme sebebi dilekçe tarama işleminin çok daha hızlı yapılması ve arkaplanda çalışan kodların sunucu performansını düşürmemesidir.

3.4.2.a. Giriş

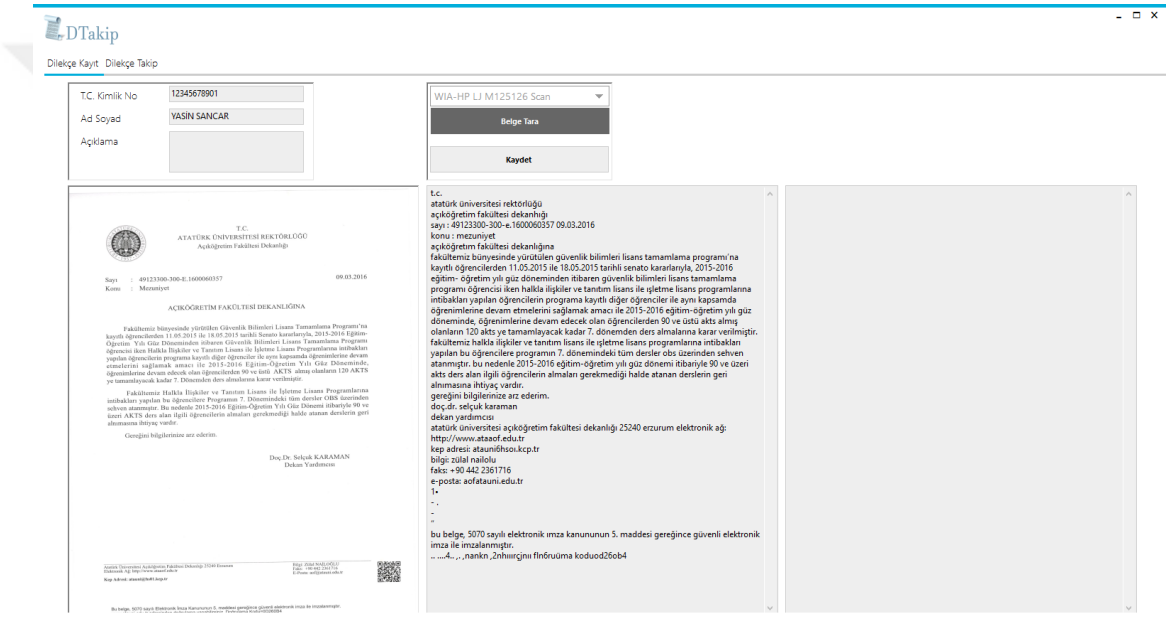
Uygulamada dilekçe kabul personeli için giriş ekranı oluşturulmuştur. Şekil 3.6'da gösterilen bu ekranda personel kullanıcı adı ve şifresiyle sisteme giriş yapabilmektedir.



Şekil 3.6. Masaüstü Uygulaması Giriş Ekranı

3.4.2.b. Dilekçe tarama

Dilekçe tarama işlemi Twain ile yapılmıştır. Şekil 3.7'de tarama için kullanılan ekran gösterilmiştir. Form başladığında Twain desteği olan mevcut tarayıcılar açılır liseye eklenmiştir. Personel dilekçeyi gönderen kişiye ait kimlik numarası, adı soyadı ve açıklama bilgilerinden en az birini girerek Belge Tara butonu ile taramayı yapabilmektedir.



Şekil 3.7. Dilekçe tarama işlemi

Belge tarama işemi gerçekleştikten sonra tarayıcıdan alınan dilekçe resmi sol taraftaki panele eklenmiştir. Ardından MODI ile optik karakter tanıma yapılmıştır. OCR sonucunda alınan metin orta bölümdeki panele aktarılmıştır.

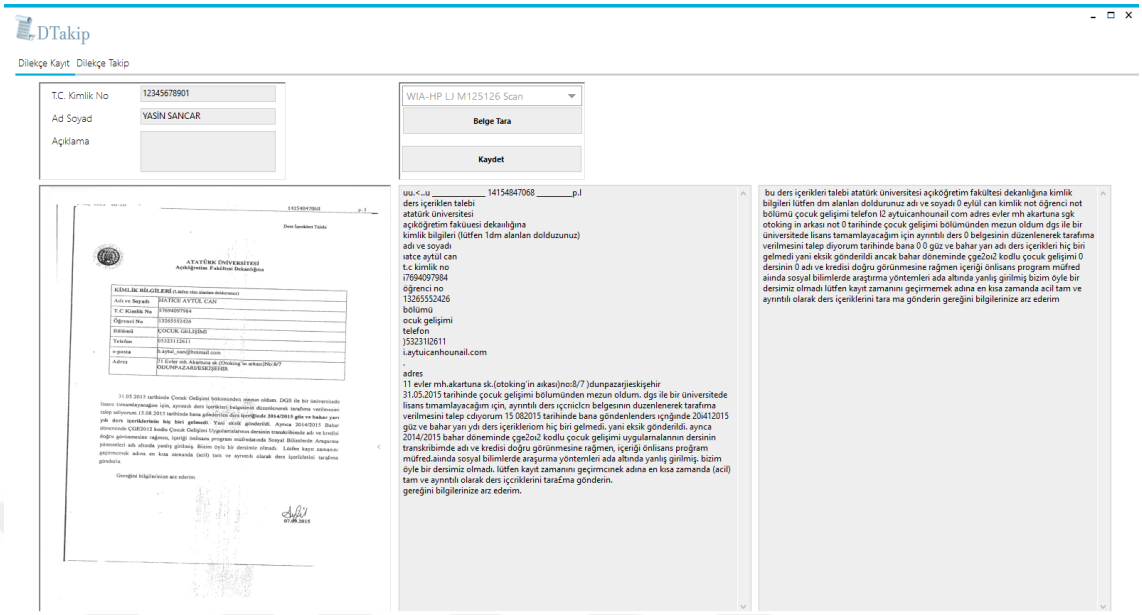
3.4.2.c. Hatalı kelimelerin tespiti ve düzeltilmesi

Kelime düzeltme işlemi için öncelikle OCR ile dönüştürülen metindeki tüm kelimeler bir diziye aktarılmıştır. Sonrasında yine dönüştürülen metin Microsoft Office Interopt Word kütüphanesi ile arkaplanda yeni oluşturulan bir uygulamaya aktarılmıştır. Bu kütüphane ile metin içerisindeki tüm hatalı kelimeler çıkarılmıştır. Sonrasında her yanlış kelime için Leveinshtein algoritması ile doğru kelime bulunmuş ve hatalı metin içerisinde değiştirilmiştir. Şekil 3.8'de bu işlemin yapıldığı kod bloğu gösterilmiştir. Hatalı kelimelerin düzeltilmesinde uzunluğu 3 ve altında olan kelimeler baz alınmamıştır.

```
string correcttext = ",";
string[] dizi = Regex.Split(tBox.Text, @"[^a-zA-ZöüğşÖÜĞŞiÇç1-]");
for (int i = 0; i < dizi.Length; i++)
{
    if (dizi[i].Length > 1)
    {
        correcttext += dizi[i].ToString().Trim() + ",";
    }

}

Microsoft.Office.Interop.Word.Application app = new Microsoft.Office.Interop.Word.Application();
app.DisplayAlerts = Microsoft.Office.Interop.Word.WdAlertLevel.wdAlertsNone;
app.Visible = false;

object template = Missing.Value;
object newTemplate = Missing.Value;
object documentType = Missing.Value;
object visible = true;
object optional = Missing.Value;

_Document doc = app.Documents.Add(ref template, ref newTemplate, ref documentType, ref visible);
doc.Words.First.InsertBefore(tBox.Text);
Microsoft.Office.Interop.Word.Words wrd = doc.Words;
Microsoft.Office.Interop.Word.ProofreadingErrors we = doc.SpellingErrors;

int k = we.Count;
string dogru1 = "", yanlis1 = "";

for (int i = 1; i <= k; i++)
{
    dogru1 = "," + enyakinbul(we[i].Text.ToString().TrimEnd().TrimStart()) + ",";
    yanlis1 = "," + we[i].Text.ToString() + ",";
    correcttext = correcttext.Replace(yanlis1, dogru1);
}

correcttext = correcttext.Replace(",", " ");

app.Options.SavePropertiesPrompt = false;
app.Options.SaveNormalPrompt = false;
doc.Close(Microsoft.Office.Interop.Word.WdSaveOptions.wdDoNotSaveChanges);
app.Quit();
```

Şekil 3.8. Hatalı kelimelerin tespiti ve düzeltilmesi işlemi

Şekil 3.9'da taranmış ve metinleri çıkarılmış bir dilekçede hataların düzeltilmeden önceki durumu ve düzeltildikten sonraki durumu gösterilmiştir.



Şekil 3.9. Hatalı kelimelerin tespiti ve düzeltilmesi işlemi

Şekil 3.9'da görüldüğü gibi Türkçe sözlükte dekanlığına olarak geçen kelime sistem tarafından dekaılığına olarak algılanmış ve hatalı olarak tespiti yapılan bu kelimenin doğru hali Levenshtein yöntemi ile "dekanlığına"olarak doğru bir şekilde düzeltilmiştir. Bu şekilde sözlükte var olmayan, hatalı olarak tespit edilen bütün kelimeler bu aşamada düzeltilmiştir.

3.4.2.d. Kelime köklerinin bulunması

Uygulamanın bu aşamasında Naive Bayes Multinominal algoritmasının son aşaması için kelime köklerinin bulunması Zemberek kütüphanesi kullanılarak yapılmıştır. Daha önceden düzeltilmiş olan metin kelimelere ayrılıp bir diziye aktarılmıştır. Zemberek ile bu dizideki tüm kelimelerin kökü bulunmuş ve bir diğer diziye aktarılmıştır. Bu işlem kod bloğu olarak Şekil 3.10'da gösterilmiştir.

```
string[] kok = Regex.Split(correcttext, @"[^a-zA-Z2öüğşÖÜĞŞiÇçı-]");

kokbul b = new kokbul();
string temp = "";
for (int i = 0; i < kok.Length; i++)
{
    temp += b.kokbulucu(kok[i].ToString()) + " ";

}
```

Şekil 3.10. Kelime köklerinin bulunması

3.4.2.e. Naive bayes multinominal algoritması ile dilekçenin sınıflandırması

Kelime kökleri bulunduktan sonra kelime ağırlıkları veritabanından çekilerek algoritmanın son adımı gerçekleştirilmiştir. Bu aşamada kelime köklerinden anahtar kelimeler ayrıştırılmıştır. Bu anahtar kelimeler birimlere göre ağırlıkları ve birimin genel ağırlığı ile çarpılmıştır. En yüksek çıkan değer benzerlik oranı en yüksek olan birim olarak tespit edilmiştir. Şekil 3.11'de daha önceden veritabanında kayıtlı olan kelime ağırlıklarına göre bu işlemi gerçekleştirilen kod bloğu gösterilmiştir.

```
double belgetalep = 1, derskayit = 1, harc = 1, sinav = 1, staj = 1;
for (int i = 0; i < veriler.Tables[0].Rows.Count - 1; i++)
{
    for (int j = 0; j < hedef.Length; j++)
    {
        if (hedef[j] == veriler.Tables[0].Rows[i][1].ToString())
        {
            belgetalep *= Convert.ToDouble(veriler.Tables[0].Rows[i][2].ToString());
            derskayit *= Convert.ToDouble(veriler.Tables[0].Rows[i][3].ToString());
            harc *= Convert.ToDouble(veriler.Tables[0].Rows[i][4].ToString());
            sinav *= Convert.ToDouble(veriler.Tables[0].Rows[i][5].ToString());
            staj *= Convert.ToDouble(veriler.Tables[0].Rows[i][6].ToString());
        }
    }

}
belgetalep *= Convert.ToDouble(veriler.Tables[0].Rows[75][2].ToString());
derskayit *= Convert.ToDouble(veriler.Tables[0].Rows[75][3].ToString());
harc *= Convert.ToDouble(veriler.Tables[0].Rows[75][4].ToString());
sinav *= Convert.ToDouble(veriler.Tables[0].Rows[75][5].ToString());
staj *= Convert.ToDouble(veriler.Tables[0].Rows[75][6].ToString());

Dictionary<string, double> results = new Dictionary<string, double>();

results.Add("belgetalep", belgetalep);
results.Add("derskayit", derskayit);
results.Add("harc", harc);
results.Add("sinav", sinav);
results.Add("staj", staj);

KeyValuePair<string, double> max = new KeyValuePair<string, double>();

foreach (var kvp in results)
{
    if (kvp.Value > max.Value)
        max = kvp;
}

return max.Key.ToString();
```

Şekil 3.11. Naive Bayes algoritmasının son adımında dilekçenin sınıflandırılması

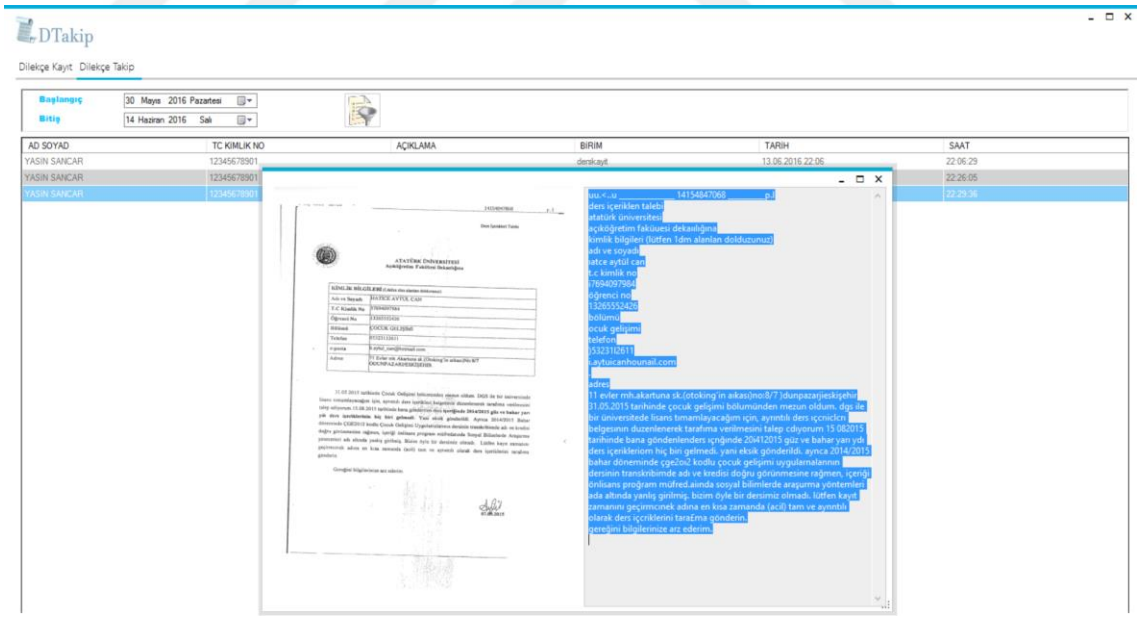
Son aşamada belgenin sisteme kaydedilmesi işlemi gerçekleşmiştir. Ekleme işlemi yapılırken transaction kullanılmıştır. Transaction ya hep ya hiç kuralıyla çalıştığı için sisteme dilekçe bilgileri eksiksiz eklenmesi sağlanmıştır. Aşağıda dilekçe kaydının yapıldığı Stored Procedured gösterilmiştir.

```
USE [DTakipDB]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[YeniDilekceKaydet]
	@tc varchar(13),
	@adsoyad varchar(100),
	@aciklama varchar(200),
	@dilekceresim image,
	@icerik varchar(8000),
	@duzenlenmisicerik varchar(8000),
	@birim varchar(50),
	@userid int
AS
BEGIN

	SET NOCOUNT ON;
		DECLARE @intErrorCode INT
		declare @zaman datetime,@dilekceid int,@resimid int,@dicerikid int
		set @zaman = getdate()
	begin transaction

		insert into
Dilekce(Aciklama,GonderenAdSoyad,GonderenTc,CreateUserId,CreateDate)
		select @aciklama,@adsoyad,@tc,@userid,@zaman

		select  @dilekceid=SCOPE_IDENTITY()

		insert into DilekceBirim
		select @dilekceid,Birimid from Birimler where BirimAd=@birim

		SELECT @intErrorCode = @@ERROR
    IF (@intErrorCode <> 0) GOTO PROBLEM

		insert into DilekceResim(resim)
		select @dilekceresim

		select  @resimid=SCOPE_IDENTITY()

		SELECT @intErrorCode = @@ERROR
    IF (@intErrorCode <> 0) GOTO PROBLEM

		insert into
DilekceIcerik(DilekceResimId,Icerik,duzeltilmisIcerik,CreateUserId,CreateDate)
		select @resimid,@icerik,@duzenlenmisicerik,@userid,@zaman

		select @dicerikid=SCOPE_IDENTITY()

		SELECT @intErrorCode = @@ERROR
    IF (@intErrorCode <> 0) GOTO PROBLEM

		insert into DilekceIcerikIliskisi(dilekceId,dilekceIcerikId)
		select @dilekceid,@dicerikid

		select 'Kayıt işlemi başarıyla tamamlandı.'
```

```
        SELECT @intErrorCode = @@ERROR
    IF (@intErrorCode <> 0) GOTO PROBLEM
            COMMIT TRAN

            PROBLEM:
IF (@intErrorCode <> 0) BEGIN
PRINT 'beklenmeyen bir hata oluştu.'
    ROLLBACK TRAN
END
END
```

3.4.2.f. Sisteme kaydedilen dilekçelerin takibi

Uygulamada daha önceden taranmış dilekçelerin gönderen bilgisi, içeriği ve resmi Şekil 3.12'de görüldüğü gibi filtreleme yapılarak takip edilebilmektedir.



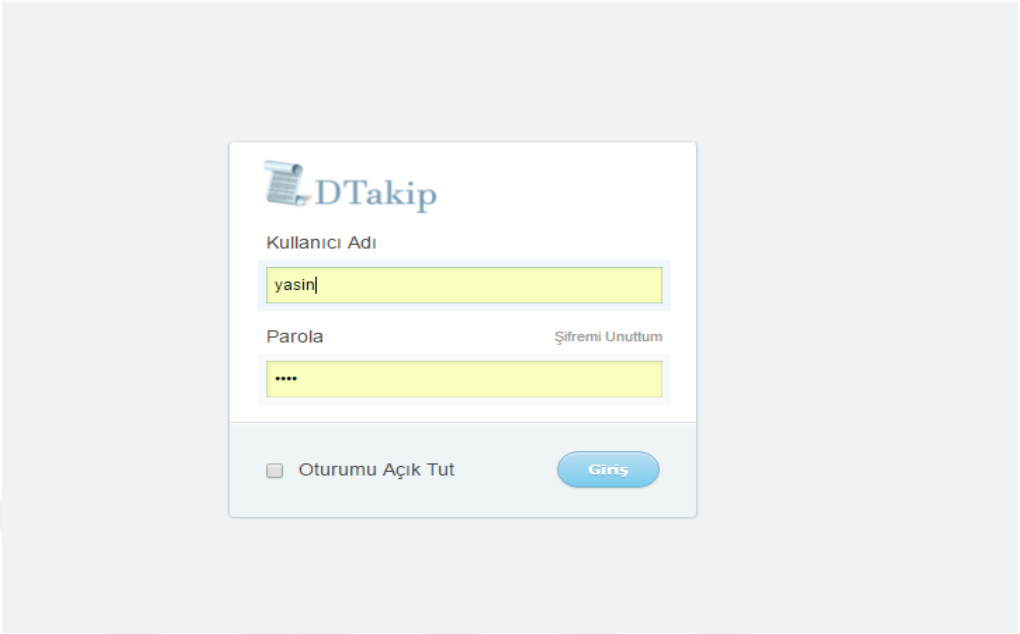
Şekil 3.12. Dilekçe takip sayfası

Bu işlem için veritabanında yazılan Stored Procedured aşağıda gösterilmiştir.

```
USE [DTakipDB]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[dilekceListe]
	@tarih1 varchar(40),
	@tarih2 varchar(40)
AS
BEGIN
	SET NOCOUNT ON;

	select
	d.Dilekceid,
	GonderenAdSoyad as 'AD SOYAD'
	,GonderenTc AS 'TC KİMLİK NO'
	,Aciklama as 'AÇIKLAMA'
	,br.BirimAd as 'BİRİM'
	,dbo.datetovarchar(d.CreateDate) as 'TARİH'
	,dbo.datetotime(d.CreateDate) as 'SAAT'
	from Dilekce d inner join dilekcebirim db on db.dilekceId = d.Dilekceid
	inner join Birimler br on br.Birimid=db.birimId
	where
dbo.datetovarchar(dbo.varchartodate(d.CreateDate))>=dbo.varchartodate(@tarih1)
and
dbo.datetovarchar(dbo.varchartodate(d.CreateDate))<=dbo.varchartodate(@tarih2)

END
```

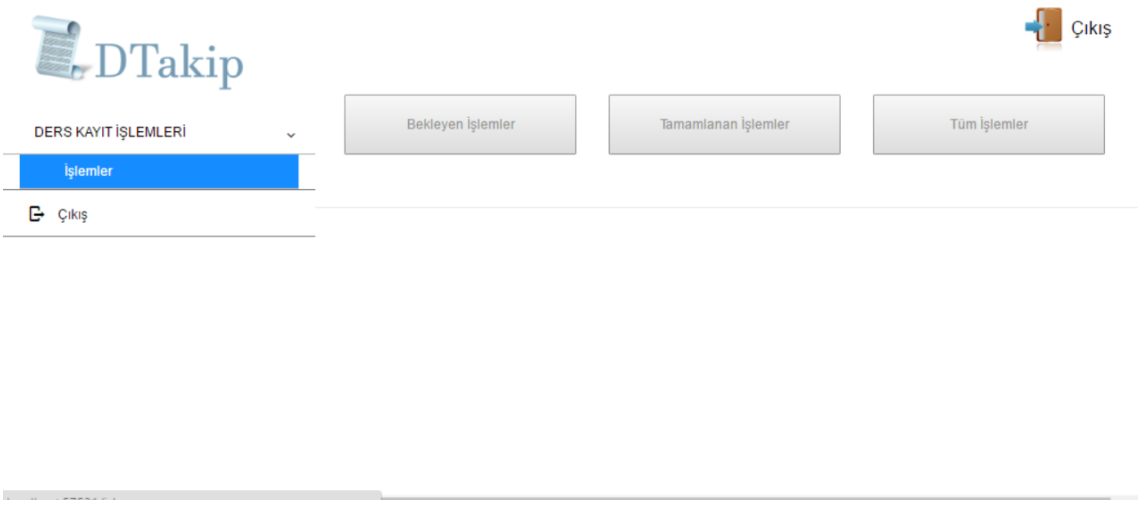
3.4.3. Web uygulaması

Birim personelleri için çalıştığı birime yönlendirme yapılan dilekçeleri görüntüleme ve işlem yapmak amacıyla kullanacakları bir web uygulaması yapılmıştır. Şekil 3.13'de sisteme giriş sayfası gösterilmiştir. Personeller giriş sayfasından yalnızca yetkili oldukları sayfalara yönlendirilirler.



Şekil 3.13. DTakip giriş ekranı

Şekil 3.13'de görüldüğü gibi Kullanıcı Adı ve Parola ile giriş yapıldıktan sonra bekleyen, tamamlanan ve tüm işlemleri görüntüleyebileceği sayfa, Şekil 3.14'de gösterilmiştir.



Şekil 3.14. Dilekçe Takip işlem ekranı

4. ARAŞTIRMA BULGULARI ve DENEYSEL SONUÇLAR

OCR işleminde karakterlerin kaliteli bir biçimde taranması gerekir. Farklı tarayıcılarla yapılan testlerde dpi özelliği yüksek tarayıcılar tarama işlemini çok daha kaliteli yapmıştır. Yüksek dpi özelliğine sahip cihazlarda OCR daha başarılı sonuçlanmıştır. Bu da hem kelime düzeltme hem de yönlendirme için başarı oranını arttırmıştır.

Kullanılan Leveinshtein algoritması, boyutu küçük olan kelimelerde benzer kelime sayısı fazla olduğu için hem düzeltme hem de yönlendirme işlemlerinde başarı oranını düşürmüştür. Bundan dolayı metin için 4 ve üzeri karaktere sahip kelimeler üzerinde düzeltme işlemi yapılmıştır.

Çalışmada dilekçelerin hangi algoritma ile daha iyi sınıflandırdığını tespit etmek amacı ile oluşturulan veri seti üzerine WEKA programında var olan Knn, Naive Bayes, Naive Bayes Multinominal, Destek Vektör Makineleri (SMO), J48 Algoritması, Geri Yayılım Yöntemi uygulanmıştır. Bu uygulama sonucunda sınıflandırma başarısı en yüksek olan algoritma tespit edilerek bu algoritmanın C# ile .NET platformuna aktarılması işlemi yapılmış ve sisteme kayıtlı olan ve kullanıcılara kolaylık olması açısından şahsa özel gelen dilekçelerin internet üzerinden doğrudan kullanıcıya aktarılması gerçekleştirilmiştir.

Yapılan çalışmada anahtar kelimeler önce 50 adet seçilmiştir. Sonrasında bu sayı 75 e çıkarılmıştır. Başarı oranı yapılan testlerde gözle görülür şekilde artmıştır. Daha fazla dilekçe örneği ve daha fazla anahtar kelime Naive Bayes Multinominal algoritmasının başarımını arttırdığı gözlemlenmiştir.

4.1. Veri Setinin Algoritmalara Göre Sınıflandırma Başarı Oranları

Veri setinin eğitim ve test verilerine ayrılmasında 10 katlamalı çapraz doğrulama yöntemi kullanılarak çeşitli algoritmalara uygulanması sonucunda elde edilen başarı yüzdeleri Çizelge 4.1'de verilmiştir.

Çizelge 0.1. Veri Setinin Algoritmalara Göre Sınıflandırma Başarı Oranları

	kNN	NB	MNB	DVM	GY	J48
k=1	%76.5	%87.1	%87.6	%84.9	%85.8	%78.3
k=2	%70.3					
k=3	%71.6					
k=4	%69.9					
k=5	%69.9					

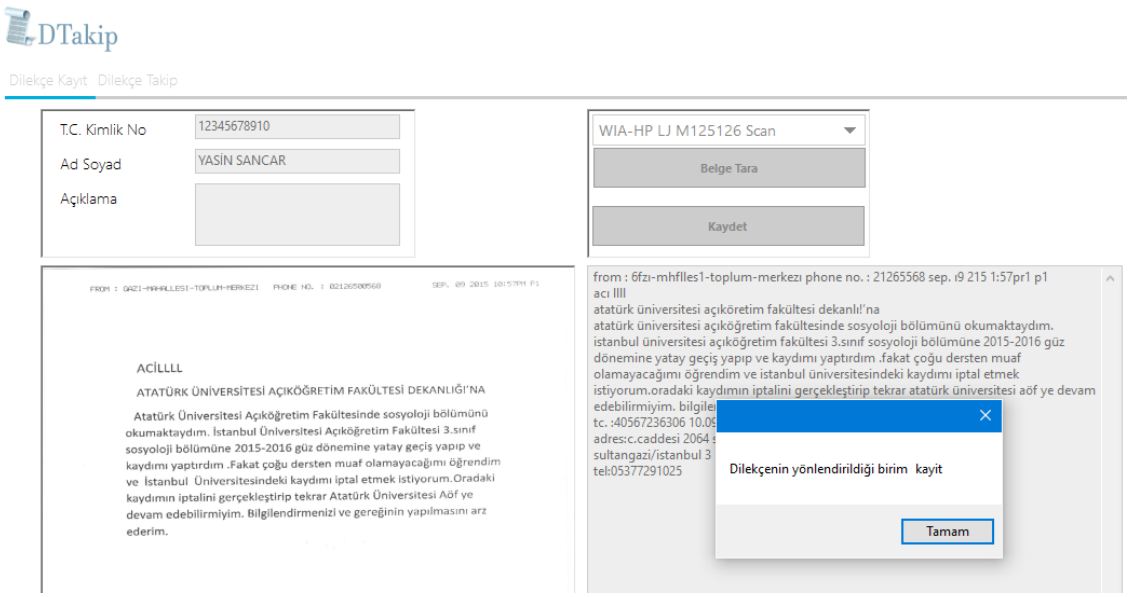
Çizelge 4.1'de görüldüğü gibi oluşturulan veri setine 6 farklı sınıflandırma yöntemi uygulanmış ve bu yöntemler içerisinden en yüksek başarı oranı %87.6 değeri ile Multinominal Naive Bayes algoritmasından elde edilmiştir. Sınıflandırma yöntemlerden Çizelge 4.1'deki sonuçların elde edilmesinde kullanılan varsayılan parametreler Çizelge 4.2'de gösterilmiştir.

Çizelge 0.2. Sınıflandırma yöntemlerinde kullanılan parametrelerin varsayılan değerleri

Algoritma	**Varsayılan Değer**
kNN	k=1
	mesafe fonksiyonu=öklit mesafesi
DVM	c=1.0
	çekirdek=PolyKernel
	tolerans=0.001
	ε= 1.0E-12
MNB	Ondalık yerlerin sayısı =2
GYA	gizli katmandaki nöron sayısı=15
	öğrenme oranı=0.3
	momentum=0.2
	Eğitim safhasının sayısı=500
J48	Güven faktörü: 0.25

4.2. Uygulama ile Tarama ve Yönlendirme Testi

Çalışan uygulamanın testi için örnek dilekçelerin dışında farklı bir dilekçe taranmıştır. Tarama yapıldıktan sonra belge kaydet butonu ile arkaplanda kelime düzeltme, kök bulma ve sınıflandırma işlemleri yapılmış ve dilekçenin yönlendirileceği birim otomatik olarak veritabanına kaydedilmiştir. Şekil de yönlendirme sonucunda alınan ekran görntüsü verilmiştir. Dilekçede öğrenci üniversiteye kayıt yaptırmak istediğini ve bu konuda bilgi almak istediğini belirtmiştir. Sistem MNB algoritması ile dilekçedeki kelimelerin ağırlıklarına göre kayıt birimine olan benzerliği yüksek bulmuştur.



Şekil 4.1. Dilekçe tarama ve yönlendirme için örnek bir dilekçenin test edilmesi

5. SONUÇ ve ÖNERİLER

Bu tez çalışmasında birimlere ayrılmış büyük ölçekli kurumlarda dilekçe takibin kolaylaştırmak adına kurum çalışanlarının masaüstü uygulaması olarak kullanabileceği bir dilekçe tanıma sistemi tasarlanmıştır. Dilekçelerin doğru sınıflandırılması için Atatürk Üniversitesi Açık Öğretim Fakültesi'nden elde edilen ve birimleri belli olan dilekçe örnekleri kullanılmıştır. Bu dilekçe örnekleriz bilgisayar ortamına resim formatında aktarılmış ve daha sonra OCR aracılığı ile resim formatında açılan dilekçede geçen metinler elde edilmiştir. OCR ile tanınan metinlerde bazı kelimelerde yapılan hatalar düzeltilerek düzeltilmiş metin üzerinden özellik çıkarımı yapılmış ve dilekçelerde geçen anahtar kelimeler ile birlikte ait oldukları sınıf etiketleri bir csv dosyasına kaydedilmiştir. Böylece dilekçelerin hangi yöntem ile en başarılı sınıflandırmaya sahip olduğunu test etmeye yarayan bir veri seti oluşturulmuştur. Uygulanan çeşitli sınıflandırma yöntemlerinden en başarılı yöntem olarak seçilen Multinominal Naive Bayes yöntemi kullanılarak gelen dilekçenin ait olduğu birime yönlendirilmesi için bir sistem tasarlanmıştır. Birime ait kurum çalışanının gelen dilekçeyi takip etmesi, dilekçeye cevap vermesi, gelen ve giden dilekçelerin karşı tarafa iletilme durumu, bekleme süresi gibi kontrolleri yapabileği bir web uygulaması tasarlanmıştır.

İleri çalışmalarda sadece dijital ortamda yazılmış dilekçeler üzerinden değil el yazısı ile yazılmış dilekçelerin tanınma işlemleri de gerçekleştirilecektir. Ayrıca sadece Türkçe diline ait dilekçelerin tanımlandığı bu çalışmanın dil bağımsız olarak tasarlanması ve otomatik dil algılama özelliğinin eklenerek tüm dünyada birçok kurumda kullanılabilecek evrensel bir sistem kurulması planlanmaktadır.

KAYNAKLAR

Anonim, 2016a. http://www.ankara.edu.tr/wp-content/uploads/sites/6/2013/06/Dilekce_Genelge.pdf (03/05/2016)

Anonim, 2016b. http://yazilimatlasi.blogspot.com.tr/2015/05/c-ile-kelime-cozumleyici-taklarna.html

Anonymous, 2016a. http://www.codeproject.com/Articles/41709/How-To-Use-Office-OCR-Using-C

Anonymous, 2016b. https://msdn.microsoft.com/en-us/library/ms173188(v=vs.80).aspx

Anonymous, 2016c. https://sarafftwain.codeplex.com/

Anonymous, 2016d. http://www.cs.waikato.ac.nz/ml/weka/

Bouzaieni, A., Barrat, S., and Tabbone S. 2015. Automatic Annotation Extension and Classification of Documents Using a Probabilistic Graphical Model, 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 316-320

Dolgun, M.Ö., Özdemir, T.G., ve Oğuz, D. 2009. Veri Madenciliğinde Yapısal Olmayan Verinin Analizi. İstatistikçiler Dergisi, 48-58

Devasena, C., and Hemalatha. 2012. Automatic Text Categorization and Summarization using Rule Reduction, IEEE- International Conference On Advances In Engineering, Science And Management (ICAESM), pp. 594-598, March 30

Döven, S. 2013. Metin Madenciliği ile Dokümanlar Arasındaki Benzerliklerin Bulunması, Bahçeşehir Üniversitesi, Bilgi Teknolojileri Yüksek Lisans Programı, İstanbul

Eikvil, L. 1993. OCR-Optical Character Recognition, Norsk Regnesentral, P.B. 114 Blindern, N-0314 Oslo, 1-35 (http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=136AD9BB6B806DD6575F9FCE9BBC4E14?doi=10.1.1.25.3684&rep=rep1&type=pdf)

Karasu, K., ve Baştan, M. 2015. Turkish OCR on Mobile and Scanned Document Images, 23nd Signal Processing and Communications Applications Conference (SIU), May 16-19, 2074-2077

Kır, B., Öz, C., ve Gülbağ, A. 2011. Yapay Sinir Ağlarında Negative Correlation Learning Metodunu Kullanarak Optik Karakter Tanıma. E.lektrik-Elektronik Bilgisayar Sempozyumu (FEEB)

Nathiya, N., and Pradeepa K. 2013. Optical Character Recognition for Scene Text Detection, Mining and Recognition, IEEE Computational Intelligence and Computing Research (ICCIC), Dec 26-28, 1-4

Su, Z., Ahn B., Eom, K., Kang, M., Kim, J., and Kim, M. 2008. Plagiarism Detection Using the Levenshtein Distance and Smith-Waterman Algorithm, The 3rd Intetnational Conference on Innovative Computing Information and Control (ICICIC'08), June 18-20, 569

Tan, A. 1999. Text Mining: The state of the art and the challenges, In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases

Stavrianou, A., Andritsos P., Nicoloyannis N. 2007. Overwiev and Semantic Issues of Text Mining, SIGMOD Record, Vol. 36, No. 3

Mohammed, H. 2007. Automatic Documents Classification. Computer Engineering & Systems, 2007. ICCES '07. International Conference, 27-29 November

Silva, G., and Dorneles, C. 2015. Towards Automatic Document Classification by Exploiting only Knowledge Resources, 34th International Conference of the Chilean Computer Science Society (SCCC), 9-13 November, pp. 1-6, Santiago

Núñez, H., and Ramos, E. 2012. Automatic classification of academic documents using text mining techniques, Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En, 1-5 October, pp. 1-7, Medellin

Jiang, S., Pang, G., Wu, M., and Kuang L. 2012. An improved K-nearest-neighbor algorithm for text categorization, Expert Systems with Applications 39, pp. 1503-1509

Kulkarni, A., Tokekar V., and Kulkarni P. 2012. Identifying context of text documents using Naïve Bayes classification and Apriori association rule mining, CSI Sixth International Conference-Software Engineering (CONSEG), Sept 5-7, pp. 1-4

Qureshi M, Aldheleai, H., and Tamandani, Y. 2015. An Improved Documents Classification Technique Using Association Rules Mining, IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pp. 460-465

Mouthami K., Devi K., and Brashkaran V. 2013. Sentiment Analysis and Classification Based on Textual Reviews, Information Communication and Embedded Systems (ICICES), 21-22 Feb., pp. 271-276

Intarapaiboon P. 2016. Text Classification using similarity measures on intuitionistic fuzzy sets, ScienceAsia 42, pp. 52-60

Han, E., and Karypis, G. 2002.Centroid-Based Document Classification Algorithms: Analysis & Experimental Results, Principles of Data Mining and Knowledge Discovery, Volume 1910 of the series Lecture Notes in Computer Science, pp. 424-431,

Jiang, C., Coenen, F., Sanderson, R., and Zito, M. 2009. Text Classification using Graph Mining-based Feature Extraction, Research and Development in Intelligent Systems XXVI, pp. 21-34,

Shravankumar, B., and Ravi, V. 2015. Text Classification Using Ensamble Features Selection and Data Mining Techiques, Swarm, Evolutionary, and Memetic Computing Volume 8947 of the series Lecture Notes in Computer Science pp 176-186,

Harvinder, Soni, D., and Madan, S. 2015. An Efficient Approach to Book Review Mining Using Data Classification, Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2, Volume 338 of the series Advances in Intelligent Systems and Computing pp 629-636

Yıldız, H., Gençtav, M., Usta, N, Diri, B., ve Amasyalı, F. 2007. Metin Sınıflandırmada Yeni Özellik Çıkarımı, IEEE 15th Signal Processing and Communications Applications, pp. 1-4.

Tüfekçi, P., Uzun, E., ve Sevinç, B. 2012. Text classification of web based news articles by using Turkish grammatical features, 20th Signal Processing and Communications Applications Conference (SIU), pp. 1-4.

Parlak, B., ve Uysal, A. 2015. Tıbbi Dokümanların Hastalıklara Göre Sınıflandırılması, 23nd Signal Processing and Communications Applications Conference (SIU), pp. 1635-1638

Levent, V., ve Diri, B. 2014. Türkçe Dokümanlarda Yapay SinirAğları İle Yazar Tanıma, Akademik Bilişim'14 - XVI. Akademik Bilişim Konferansı Bildirileri, Mersin Üniversitesi, pp. 735-741

Sriurai, W. 2011. Improving Text Categorization by Using a Topic Model, Advanced Computing: An International Journal (ACIJ), Vol.2, No.6, November 2011

Fidan, Ü. 2013. Destek Vektör Makineleri ile Doküman Sınıflandırma, Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi

Doğan, S., ve Diri B. 2015. Türkçe Dokümanlar için N-gram Tabanlı Yeni bir Sınıflandırma (Ng-ind): Yazar, Tür ve Cinsiyet, Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, pp. 11-19

Çobanoğlu, Ö. 2015. Türkçe metinler için doküman sınıflandırma yaklaşımlarının karşılaştırılması, İzmir Yüksek Teknoloji Enstitüsü / Mühendislik ve Fen Bilimleri Enstitüsü / Bilgisayar Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi

Kılıç, E., Ateş, N., Karakaya,A., ve Şahin, D. 2015. Metin Sınıflandırma için İki Yeni Öznitelik ÇıkartmaYöntemi: TESDF ve SADF, 23nd Signal Processing and Communications Applications Conference (SIU), pp. 475-478

McCallum A., and Nigam, K. 1998. A comparison of event models for Naive Bayes text classification, AAAI/ICML-98 Workshop on Learning for Text Categorization, Technical Report WS-98- 05, pp. 41-48.

Kibriya, A., Frank, E., Pfahringer, B., and Holmes, G. 2004. Multinominal Naive Bayes for Text Categorization Revisited, AI 2004: Advances in Artificial Intelligence, Volume 3339 of the series Lecture Notes in Computer Science, pp. 488-499

ÖZGEÇMİŞ

Yasin SANCAR 1988 yılında Bayburt'da doğdu. İlk, orta ve lise öğrenimini Erzurum'da tamamladı. 2006-2013 yılları arasında İstanbul Üniversitesi Bilgisayar Mühendisliği bölümünde lisans derecesini aldı. Mezuniyetinin ardından 2013 yılında Atatürk Üniversitesi'nde yüksek lisans eğitimine başladı. 2014 yılında Atatürk Üniversitesi Açık Öğretim Fakültesi'nde uzman olarak göreve başladı ve halen bu görevi sürdürmektedir.