METİN SINIFLANDIRMA TEKNİKLERİ İLE
TÜRKÇE TWITTER DUYGU ANALİZİ

Önder ÇOBAN

Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Yrd. Doç. Dr. Gülşah TÜMÜKLÜ ÖZYER
2016

ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

METİN SINIFLANDIRMA TEKNİKLERİ İLE TÜRKÇE TWITTER DUYGU ANALİZİ

Önder ÇOBAN

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ERZURUM
2016

T.C.
ATATÜRK ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü Müdürlüğü

TEZ ONAY FORMU

METİN SINIFLANDIRMA TEKNİKLERİ İLE TÜRKÇE TWITTER DUYGU ANALİZİ

Yrd. Doç. Dr. Gülşah TÜMÜKLÜ ÖZYER danışmanlığında, Önder ÇOBAN tarafından hazırlanan bu çalışma, 20/06/2016 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans tezi olarak **oybirliği**/~~**oy çokluğu (.../...)**~~ ile kabul edilmiştir.

Başkan: Yrd. Doç. Dr. İbrahim Aykut ERDEM	İmza	:
Üye : Yrd. Doç. Dr. Gülşah TÜMÜKLÜ ÖZYER	İmza	:
Üye : Yrd. Doç. Dr. Levent BAYINDIR	İmza	:

Yukarıdaki sonuç;

Enstitü Yönetim Kurulu'nun 30.06./2016 tarih ve ..27.../..35..... 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 SINIFLANDIRMA TEKNİKLERİ İLE TÜRKÇE TWITTER DUYGU ANALİZİ

Önder ÇOBAN

Atatürk Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Gülşah TÜMÜKLÜ ÖZYER

Metin sınıflandırma doğal dil metinlerini önceden tanımlanmış veya mevcut kategorilerden birisiyle etiketleme işlemidir. Doküman sınıflandırma, istenmeyen mesajların filtrelenmesi ve web sorgulamaları için doğru sonuçların görüntülenebilmesi gibi problemler metin sınıflandırma çalışmalarına örnek olarak verilebilir. Duygu analizi ise kişisel blog ve sosyal medya gibi mecralardan elde edilen metinsel veriler üzerinde semantik bilginin ortaya çıkarılması amacını taşır. İşlenen veriler kısa metinlerden oluştuğu için duygu analizi de bir metin sınıflandırma problemi olarak ele alınır. Metin sınıflandırma ve duygu analizi problemlerinin çözümü için diğer dillerde gerekli sistemler geliştirilmiş olmakla beraber Türkçe için yapılan çalışmalar oldukça sınırlıdır. Bu tezde, Twitter'dan elde edilen Türkçe mesajlar üzerinde iki kategorili duygu analizi çalışılmıştır. Duygu analizi bir metin sınıflandırma problemi olarak düşünülmüş; duygu analizi tekniklerinin yanı sıra klasik metin sınıflandırma teknikleri de kullanılmıştır. Twitter mesajlarında gözlenen baskın duygunun otomatik olarak tespitinde ise makine öğrenmesi yöntemleri kullanılmıştır. Hem metin sınıflandırma hem de duygu analizi deneylerinin gerçekleştirildiği bu çalışmada, asıl hedef ise duygu analizi başarısını artırmak olmuştur. Bu amaçla Türkçe Twitter duygu analizinde farklı önişleme, etiketleme, sınıflandırma ve benzerlik yöntemlerinin etkisi incelenmiştir. Ayrıca topik bilgisine dayalı etiketleme yöntemi önerilmiş ve en yüksek %92,50 oranında başarı elde edilmiştir. Böylece dil işleme ile ilgili işlemler uygulanmadan duygu analizi başarısı önceki çalışmalara kıyasla daha yüksek elde edilebilmiştir. Bunun yanı sıra, metin sınıflandırma ve duygu analizi süreçlerini otomatikleştirme amacıyla hem Türkçe hem de İngilizce metinsel içerikleri işleyebilen bir yazılım prototipi geliştirilmiştir.

2016, 109 sayfa

Anahtar Kelimeler: Twitter, duygu analizi, metin sınıflandırma, örüntü tanıma, makine öğrenmesi

ABSTRACT

MS Thesis

TURKISH TWITTER SENTIMENT ANALYSIS USING TEXT CLASSIFICATION TECHNIQUES

Önder ÇOBAN

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

Supervisor: Asst. Prof. Dr. Gülşah TÜMÜKLÜ ÖZYER

Text classification assigns one of available or predefined categories to natural language texts. Document classification, spam message filtering and retrieving the suitable results for web inquiries are examples of text classification studies. The sentiment analysis aims extracting semantic information from textual data which mostly obtained from personal blogs or social media platforms. Sentiment analysis is also considered as a text classification task due to the processed data consist of short texts. The necessary systems have been developed to solve text classification and sentiment analysis problems for other languages but there are quite a few studies for Turkish. In this thesis, binary sentiment analysis has been performed over Turkish feeds which are collected from Twitter. The sentiment analysis has been considered as a text classification task and classical text classification techniques have been employed in addition to the sentiment analysis techniques. While automatically detecting the dominant sentiment observed in Twitter feeds, machine learning techniques have been used. In this study, both text classification and sentiment analysis experiments have been performed and its main goal has been to increase the success of sentiment analysis. For this purpose the effect of different preprocessing, labeling, classification and similarity methods have been investigated in Turkish Twitter sentiment analysis. Also topic based labeling method has been suggested and the highest success rate has been obtained as %92,50. In this way, the sentiment analysis success could be obtained higher compared to the previous works without applying language processing techniques. And also, in order to automate text classification and sentiment analysis processes a software prototype has been developed with features that can handle textual contents in both Turkish and English languages.

2016, 109 pages

Keywords: Twitter, sentiment analysis, text classification, pattern recognition, machine learning

TEŞEKKÜR

Bu tezin araştırma konusunun yürütülmesinde danışmanlık yaparak yol gösteren danışman hocam Sayın Yrd. Doç. Dr. Gülşah TÜMÜKLÜ ÖZYER başta olmak üzere Sayın Yrd. Doç. Dr. Barış ÖZYER'e, çalışmalarım sırasında bana destek veren ve yardımcı olan mesai arkadaşlarımdan Sayın Arş. Gör. Ferhat BOZKURT'a, Sayın Arş. Gör. Mete YAĞANOĞLU'na, Sayın Arş. Gör. Faruk Baturalp GÜNAY'a ve Sayın Arş. Gör. Şeyma YÜCEL ALTAY'a teşekkürlerimi sunuyorum.

Ayrıca tez çalışmam süresince manevi desteklerini esirgemeyen aileme en içten teşekkürlerimi sunuyorum.

Önder ÇOBAN
Haziran, 2016

İÇİNDEKİLER

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

DA	Duygu Analizi
DDİ	Doğal Dil İşleme
k-NN	*k*-En Yakın Komşu
KTÖS	Korelasyon Tabanlı Öznitelik Seçme
ME	Maksimum Entropi
MÖ	Makine Öğrenmesi
MNB	Multinom Basit Bayes
MS	Metin Sınıflandırma
NB	Basit Bayes
SVM	Destek Vektör Makinesi
TA	Terim Ağırlıklandırma
TDA	Tekil Değer Ayrıştırması
TTM	Türkçe Twitter Mesajları
TSS	Türkçe Şarkı Sözleri
VM	Veri Madenciliği

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

1.1. Motivasyon

Bilgisayar teknolojisinde ortaya çıkan gelişmeler bilişim alanında çalışma yapan araştırmacıların dil işleme alanına ilgi duymasını sağlamıştır. Böylece doğal dil işleme (NLP: Natural Language Processing) alanında yapılan çalışmalar yoğunluk kazanmıştır. Günümüzde metin seslendirme, konuşmayı yazıya dökme, yazım hatalarını giderme, bilgi çıkarma, özetleme, anlama, çeviri ve soru yanıtlama gibi birçok alt alana ayrılabilen problemlere doğal dil işleme (DDİ) çatısı altında çözüm üretilmeye çalışılmaktadır (Adalı 2012). Metin madenciliği (TM: Text Mining) ise basitçe veri kaynağı olarak metin veya dokümanları kabul eden bir veri madenciliği (DM: Data Mining) çalışmasıdır. Ancak DDİ dil bilim bilgisine dayalı çalışmalar üzerine yoğunlaşırken, metin madenciliği çalışmaları daha çok istatistiksel ve matematiksel yöntemlerle sonuca ulaşmaya çalışmaktadır (Tan 1999). Bu bağlamda metin sınıflandırma (TC: Text Classification, Text Categorization); DDİ, veri madenciliği (VM) ve makine öğrenmesi (ML: Machine Learning) gibi bir biriyle yakın bağları bulunan bilgisayar bilimlerini kullanan bir çalışma alanıdır.

Metin sınıflandırmanın (MS), DDİ ile ilişkisi ise çalışmalar sırasında kullanılan özniteliklerin DDİ teknikleri kullanılarak metin veya dokümanlardan çıkarılmasından ileri gelmektedir. MS çalışmaları sayesinde günümüzde elle yapılması imkânsız işlemlerin daha az performans ve zaman kaybı ile etkili bir şekilde gerçekleştirilmesi mümkün olmaktadır (Dalal and Zaveri 2011). Günümüzde MS teknikleri kullanılarak geliştirilen sistemler ile otomatik yazar, tür, cinsiyet tanıma, belge veya doküman organizasyonu, web tabanlı uygulamalar için sorgulama performansının iyileştirilmesi, otomatik bilgi getirimi vb. gibi işlemler gerçekleştirilebilmektedir (Sebastiani 2005). Bu bağlamda günümüzde klasik MS yöntemlerinin kullanıldığı diğer çalışma alanları da oldukça popüler durumdadır. Bu alanlardan birisi de duygu analizidir. Son yıllarda sosyal medya kullanımının artması ve günlük hayatın birçok alanında etkili olması

duygu analizi (SA: Sentiment Analysis) çalışmaları için önemli bir etken olmuştur. Böylece temelde klasik MS tekniklerinin kullanıldığı ve sosyal medya verileri kullanılarak duygu tespitinin hedeflendiği duygu analizi (DA) çalışmaları önemli ölçüde artış göstermiştir (Go *et al.* 2009a). DA ile özellikle sosyal medya ortamlarından elde edilen verilerden toplumun herhangi bir konu, kişi, ürün, şirket vb. gibi varlık veya olaylarla ilgili düşüncesinin otomatik olarak tespit edilmesi amaçlanmaktadır (Liu and Zhang 2012). Günümüzde özellikle sosyal medya kullanım oranlarının yüksek olduğu ülkelerde seçim vb. gibi dönemlerde kampanyaların sosyal medya üzerinden yürütülmesi DA çalışmalarının önemini kanıtlayıcı niteliktedir. Bunun yanı sıra bu alanda yapılan çalışmalar ile günlük borsa tahmini, pazar-fiyat dengesi tahmini, herhangi bir konu ile ilgili kullanıcı düşünce, duygu ve fikirlerinin elde edilmesi vb. gibi birçok konuda DA çalışmaları kullanılmaktadır (Michelson and Macskassy 2010; Bollen *et al.* 2011).

MS ve DA çalışmaları dijital metin tabanlı veri kullanması bakımından aslında dil bağımlıdır. Yani her dil için özel alt yöntem ve yordamlar kullanılarak o dilde hedeflenen probleme çözüm aranmalıdır. Bunun yanı sıra bu çalışmaların dil ile bağlantılı olması geliştirilen sistemlerin o dili konuşan toplum için önemini ortaya koymaktadır. Çünkü her dilde yukarıda bahsedilen işlemleri otomatik olarak gerçekleştirebilecek sistemlerin geliştirilmesine ihtiyaç duyulmaktadır. Buna rağmen yapılan literatür araştırmaları sonucunda, diğer diller için bu alanda oldukça fazla sayıda çalışma yapıldığı ancak Türkçe için yapılan çalışmaların sınırlı sayıda olduğu gözlemlenmiştir (Coban vd 2015a).

Bu nedenle bu tezde, MS ve DA tekniklerinin Türkçe Twitter gönderilerinin duygu tespitinde birlikte uygulanabilirliği araştırılmış ve bu problemin üstesinden gelebilmek adına çeşitli makine öğrenmesi (MÖ) yöntemlerinin kullanılması öngörülmüştür. Bu motivasyonla bu tezde, MS çalışmalarıyla yola çıkılmış ve DA teknikleri ile Türkçe Twitter mesajları üzerinde semantik bilginin ortaya çıkarılması hedeflenmiştir.

1.2. Tezin Amacı ve Katkısı

Bu tez ile MS ve DA teknikleri Twitter DA problemine uygulanarak Türkçe Twitter mesajlarında baskın olan duygunun tespit edilmesi amaçlanmıştır. Bu nedenle öncelikle klasik MS teknikleri uygulanmış daha sonra bu tekniklerin DA için kullanılması hedeflenmiştir. Klasik MS ve DA teknikleri birlikte uygulanarak Türkçe'nin yanı sıra İngilizce mesaj/metinler üzerinde de MS ve DA çalışmalarının gerçekleştirilmesi öngörülmüştür. Bu bağlamda her iki çalışma alanında kullanılan yöntem ve teknikleri barındıran, kullanıcı odaklı bir prototip yazılımın geliştirilmesi de düşünülmüştür. Ayrıca MS ve DA alanlarında Türkçe için standart veri kümeleri olmadığından yeni veri kümelerinin oluşturulması ve diğer araştırmacılarla paylaşılması amaçlanmıştır. Bu hedeflerle bu tezde, Türkçe Twitter mesajları için DA gerçekleştirilmiş ve elde edilen başarı oranının artırılması için kullanılabilecek yöntemler incelenmiştir.

Yapılan bu araştırma ve incelemeler sonucunda MS ve DA alanlarında; istenmeyen mesajların otomatik olarak tespiti, şarkı sözlerinden müzik türü sınıflandırması ve Türkçe Twitter DA ile ilgili çalışmalar yapılmıştır. Bu çalışmalar ile bu tezde literatüre aşağıda verilen katkılar sağlanmıştır:

- (a) Klasik MS teknikleri ile Türkçe Twitter DA için ilk sayılabilecek çalışmalardan birisi gerçekleştirilmiştir (Coban vd 2015a).
- (b) Türkçe Twitter DA'da başarıyı artırabilmek amacıyla duygu sınıflandırması aşamasında LDA yöntemi ile oluşturulan topik bilgisinin kullanılması önerilmiştir (Coban ve Ozyer 2016a).
- (c) Literatürde geleneksel yöntemlerin yanı sıra MS ve Bilgi Getirimi (IR: Information Retrieval) gibi alanlarda etkili olduğu gösterilmiş benzerlik yöntemleri DA problemine uygulanmıştır. Bu amaçla *k*-NN algoritması için önerilen bu metrikler Türkçe Twitter DA problemine uygulanmış ve genişletilmiş bir performans karşılaştırması yapılmıştır (Coban vd 2015b). Literatürde daha önce yapılmış benzer bir çalışma tespit edilemediğinden, bu inceleme Türkçe Twitter mesajları için DA bağlamında oldukça önemli ve değerlidir.

- (d) DA'nın yanı sıra bu tezde MS alanında da çalışmalar yapılmıştır. Bu bağlamda, istenmeyen mesajların otomatik olarak tespit edilmesi için uzman sistem tabanlı MS tekniklerinin kullanıldığı bir sistem modeli önerilmiş ve veri setinin kullanıldığı önceki çalışmaya kıyasla daha yüksek başarı elde edilmiştir (Bozan vd 2015).
- (e) Müzik Bilgi Getirimi (MIR: Music Information Retrieval) alanında MS ve MÖ yöntemlerinin kullanıldığı Türkçe için yapılmış çalışma sayısının sınırlı olduğu tespit edilmiştir. Bu nedenle yine bu tez kapsamında bu alanda MS ve MÖ tekniklerinin kullanıldığı ilk çalışmalardan birisi gerçekleştirilmiş ve Türkçe şarkı sözlerinden otomatik olarak müzik türünün belirlenmesi çalışılmıştır (Coban ve Ozyer 2016b).
- (f) Türkçe Twitter DA ve Müzik Türü Sınıflandırması (MGC: Music Genre Classification) çalışmalarında kullanılabilecek veri kümeleri oluşturulmuştur (Coban vd 2015a; Coban ve Ozyer 2016b).
- (g) MS ve DA süreçlerini otomatikleştirme adına, bu tezde uygulanan teknikleri barındıran kullanıcı odaklı bir Otomatik Metin Sınıflandırma Sistemi (OMESIS) prototip yazılımı geliştirilmiştir.

Çalışmanın geri kalan kısmı şu şekilde düzenlenmiştir. Bölüm 2'de literatür taraması sonucu incelenen ilgili çalışmalar kısaca açıklanmıştır. Bölüm 3'te çalışmamızda kullanılan materyal (veri kümeleri) ile MS ve DA süreçlerine ait önişleme, boyut indirgeme ve sınıflandırma aşamalarında uyguladığımız yöntemlere ait detaylar verilmiştir. Ayrıca bu yöntemlerin uygulanması ve sonuçların gösterilmesini içeren OMESIS'e ait grafik arayüzü anlatılmıştır. Bölüm 4'te bu tez kapsamında MS ve DA alanlarında gerçekleştirilen çalışmalarda elde edilen deneysel sonuçlar verilmiş ve tartışılmıştır. Son bölümde ise sonuçlar özetlenmiş ve öneriler belirtilmiştir.

2. KAYNAK ÖZETLERİ

2.1. Doğal Dil İşleme

Günümüzde teknolojik gelişmelerle birlikte yeni problemler ve bu problemlerin çözümüne yönelik yeni ihtiyaçlar doğmaktadır. Bu bağlamda yeni araştırma alanları ve bilgisayar bilimleri ortaya çıkmaktadır. DDİ de benzer bir şekilde bilgisayarlara, insanlar tarafından kullanılan dilin anlatılması ve dilin doğru bir şekilde çözümlenmesi ihtiyacından doğmuş; özellikle son yıllarda oldukça popüler olmuş bir çalışma alanı ve bilgisayar bilimidir. "1950 ve 1960'larda yapay zekânın küçük bir alt alanı olarak görülen DDİ, araştırmacıların ve gerçekleştirilen uygulamaların elde ettiği başarılar sonunda artık bilgisayar bilimlerinin temel bir disiplini olarak kabul edilmektedir" (Oflazer 2006).

DDİ çalışmalarının temel amacı; bilgisayarların kendisine verilen çeşitli görevleri anlayıp işleyebilmesi ve bu yönde uygun olabilecek türde araç ve tekniklerin geliştirilebilmesi için insanın dili anlama ve kullanma yetisi ile ilgili bilgi toplamaktır (Chowdhury 2003). DDİ çalışmaları kapsamında; yardımcı yazım araçlarının geliştirilmesi, bir metnin özetini çıkarma, metni anlama, tanıma, sınıflandırma, bilgisayarların konuşması (metin seslendirme), doğal diller arası çeviri, konuşmayı anlama (konuşmayı metne dönüştürme), yazım yanlışlarının düzeltilmesi gibi daha çok genişletilebilecek konular yer almaktadır. Temelde bilgisayar ve bilgi bilimi, dil bilim, matematik, yapay zekâ, robotik vb. gibi disiplinlerle bağı bulunan DDİ yukarıda belirtildiği gibi birçok çalışma alanı içermektedir (Adalı 2012).

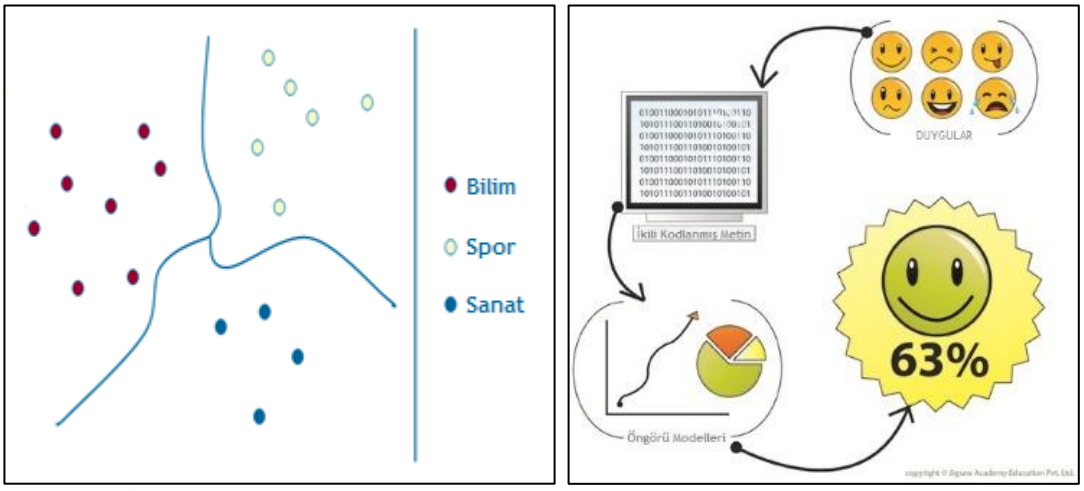
2.2. Metin Sınıflandırma ve Duygu Analizi

MS çalışmalarının başlangıcı 60'lı yıllarda Maron (1961)'un yaptığı çalışmaya dayanmaktadır. Ancak 90'lı yıllarda bu alana olan ilginin artması ve teknolojik gelişmelerle birlikte güçlü donanım kaynaklarının mevcut olması ile bilgi sistemleri

disiplininin büyük bir alt alanı olmuştur (Sebastiani 2002). MS temelde daha önceden tanımlanmış veya mevcut kategorilerden biriyle doğal dil metinlerini etiketleme işlemidir. Burada etiketlemeden kasıt doğal dil içerikli bir metin, doküman veya belgeye kategori atanması işlemidir. Günümüzde dijital ve erişilebilir veri miktarı büyük boyutlara ulaşmış bu sebeple MS çalışmaları da büyük önem ve hız kazanmıştır. Bu bağlamda MS; metin tabanlı belge, doküman vb. gibi verilerin otomatik olarak kategorize edilmesi, istenmeyen mesajların filtrelenmesi, arama motorları için sorgu sonuçlarının geliştirilmesi, fikir tespiti, yazar tanıma vb. gibi birçok çalışma alanında kullanılmaktadır (Dalal and Zaveri 2011).

Sosyal medya artık sadece bir iletişim aracı olmaktan çıkmış, birçok alanda etkili olan ve olaylara yön veren güçlü bir araç haline gelmiştir. Günümüzde kullanıcılar Facebook, Google+, Twitter vb. gibi sosyal medya ortamlarında herhangi bir konu ile ilgili duygu, düşünce, fikir ve deneyimlerini ifade edebilmektedir (Sommer *et al.* 2011). Bunun yanı sıra her geçen gün kullanımı daha da artan bu ortamlar haber paylaşımı ve organize olma gibi amaçlarla da kullanılmaktadır (Michelson and Macskassy 2010). Bu bakımdan sosyal medya; ekonomi, ticaret, siyaset ve fikir madenciliği gibi araştırma alanlarında kullanılabilecek zengin veri kaynağı sunmaktadır. Ancak bu çok büyük boyutlu veriden anlamlı bilgi çıkarılabilmesi için çeşitli otomatik yöntemlere ihtiyaç duyulmaktadır.

DA çalışmaları da bu ihtiyacı giderme amacıyla kullanılan en kullanışlı sosyal medya izleme yöntemlerinden birisidir. Bir ürün ile ilgili olumlu ve olumsuz yorumların, kullanıcıların ruh halinin ve toplumun politik konular ile ilgili fikirlerinin tespit edilmesi gibi konular bu alanda yapılan çalışmalara örnek olarak verilebilir (Liu and Zhang 2012). DA çalışmalarında veri setindeki her bir mesaj içeriği iki (pozitif, negatif) veya daha fazla kategoride (çok iyi, iyi, tatmin edici, kötü, çok kötü vb.) olmak üzere sınıflandırılabileceği için DA; her bir mesajda baskın olan duygunun bir kategoriyi temsil ettiği bir MS işlemi olarak düşünülebilir (Prabowo and Thelwall 2009). Bu nedenle literatürde DA çalışmalarında, DA teknikleri ile beraber MS yöntemleri de kullanılmaktadır.



(a) (b)

Şekil 2.1. Metin sınıflandırma (a), duygu analizi (b)

Şekil 2.1'de gösterilen bu iki çalışma alanının metin tabanlı veri ve MÖ yöntemlerinin kullanılması gibi ortak yönleri mevcuttur (Anonymous 2011a). Ancak kullanılan metin, belge veya mesajın kendine has özellikleri olabileceği için kullanılan teknikler açısından özellikle önişleme aşamasında farklılık gözlenmektedir. Bunun yanı sıra DA bir MS işlemi olarak değerlendirilebilse de klasik MS çalışmalarının aksine DA çalışmalarında anlamlı bilgi elde etmek daha zordur. Bu durumun nedeni genellikle DA çalışmalarında kullanılan metinlerin (mesaj, yorum vb.) kısa olması, resmi olmayan (informal) dil ile yazılması ve gramer kurallarına uyulmadığı için yazım yanlışları içermesidir. Ayrıca özellikle Twitter mesajları gibi kendine özgü özellikleri olan (örneğin 140 karakter sınırlaması) ve özel terimler (kullanıcı adı ve hashtag gibi) içeren metinler üzerinde anlamlı bilgi çıkarımı daha da zorlaşmaktadır.

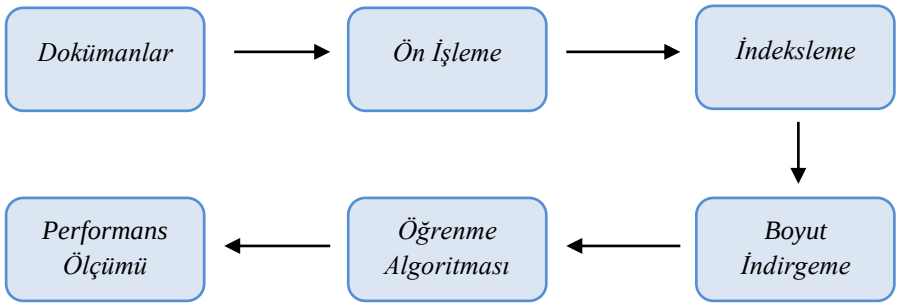
2.3. Metin Sınıflandırma ve Duygu Analizi Teknikleri

Doküman (metin) sınıflandırma (kategorize etme) sürecini bir kategori-doküman matrisindeki her bir hücreyi {0, 1} değer kümesinden biriyle ilişkilendirme işlemi olarak tasvir etmek mümkündür.

Çizelge 2.1. Kategori-doküman karar matrisi

	$\mathbf{d_1}$	…	…	$\mathbf{d_j}$	…	…	$\mathbf{d_n}$
$\mathbf{c_1}$	a_{11}	…	…	a_{1j}	…	…	a_{1n}
…	…	…	…	…	…	…	…
$\mathbf{c_i}$	a_{i1}	…	…	a_{ij}	…	…	a_{in}
…	…	…	…	…	…	…	…
$\mathbf{c_m}$	a_{m1}	…	…	a_{mj}	…	…	a_{mn}

Çizelge 2.1'de verilen $C = \{c_1, \dots, c_m\}$ önceden belirlenmiş kategori setini ve $D = \{d_1, \dots, d_n\}$ kümesi de sınıflandırılacak doküman setini temsil eder. Bir a_{ij} değerinin 1 olması d_j dokümanının c_i kategorisinde; 0 olması aksi durumda olduğunu gösterir (Sebastiani 1999). Sınıflandırma yapılırken sadece metin içeriği dikkate alınır. Her dokümana mutlaka bir kategori ataması yapılır ancak bu atama işleminin doğru olması oluşturulan sistemin başarısına bağlıdır. MS çalışmaları ile MÖ'nün kullanıldığı diğer çalışma alanları arasında süreç açısından bir farklılık yoktur. Ancak Şekil 2.2'de verilen MS sürecinde farklılığı oluşturan temel konu dokümanların önişlenmesi ve temsil edilmesidir (Ikonomakis *et al.* 2005; Baharudin *et al.* 2010; Korde and Mahender 2012).

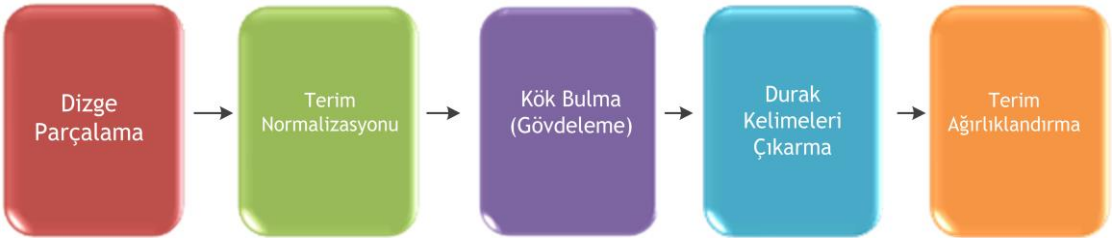


Şekil 2.2. Metin sınıflandırma süreci

MS ve DA çalışmalarının ortak yönü kullanılan verinin metin (text) tabanlı olmasıdır. MS'de metinler için önceden belirlenen kategori, DA'da ise MS çalışmalarındaki kategorinin yerini alan duygu (sentiment) sınıflandırması yapılır. Bu nedenle DA çalışmaları da bir MS problemi olarak ele alınır. Her iki çalışma alanında da temel bir sınıflandırma sürecinde kullanılan önişlemden geçirme ve temsil etme, öznitelik seçme ve MÖ yöntemleriyle sınıflandırma aşamaları ortaktır. Ancak bu aşamaların alt adımları MS ve DA çalışmalarında birbirinden farklılık göstermektedir.

2.3.1. Metin önişleme ve temsil etme

VM ve MÖ alanlarında yapılan çalışmalarda genellikle verinin boyutu ve türüne bağlı olarak değişiklik gösterebilen çeşitli önişlemlerin (preprocessing) uygulanması gerekmektedir. MS ve DA çalışmalarında da veriler (doküman, belge, mesaj, tweet vb.) yapılandırılmadığı ve doğal dil ile yazıldığı için çeşitli önişleme adımlarının uygulanması gereklidir. Önişleme işleminin temel amacı metinsel verilerden örnek kategorilerinin birbirinden ayırt edilmesini sağlayabilecek önemli özniteliklerin ortaya çıkarılması ve uygun formata dönüştürülmesidir (Srividhya and Anitha 2010). Buradaki formattan kasıt verinin MÖ algoritmalarının işleyip yorum yapabileceği şekilde sayısallaştırılmasıdır (Brücher *et al.* 2002). Önişleme aşaması, sınıflandırma işleminin başarımı ve amaca uygun doğru özniteliklerin elde edilebilmesi açısından çok önemli bir aşamadır. Bu aşama kendi içerisinde metinsel verinin diline, yapısına ve hangi amaçla işleneceğine bağlı olarak değişiklik gösterebilen alt adımlardan oluşmaktadır. MS sürecinde yaygın olarak kullanılan temel önişleme adımları Şekil 2.3'te verilmiştir (Saad 2010).



Şekil 2.3. Metin sınıflandırmada temel önişleme adımları

Yukarıdaki şekilde temel önişleme adımları verilmiştir. Bu adımlar uygulanmadan önce bir MS sisteminde anlamlı bilgi çıkarımını sağlamak amacıyla metinler noktalama işaretleri ile rakamlardan temizlenir ve küçük harf dönüşümü uygulanır (Ng *et al.* 1997). Metinsel içerik temizlendikten sonra uygulanan temel adımlar dışında daha çok dilbilimsel analiz gerektiren önişleme adımları da bulunmaktadır (Sebastiani 1999). Bu adımlarda uygulanabilen sözcük türü belirleme (POST: Part of Speech Tagging) ve sözcük anlamı belirleme (WSD: Word Sense Disambiguation) gibi dilbilimsel yöntemler ise daha çok DDİ ile ilgili işlemleri gerektiren yöntemlerdendir (Hotho *et al.*

2005). Burada açıklanan yöntemler MS ile ilişkilendirilse de bu adımlar temelde metin tabanlı veri kullanılan diğer sınıflandırma çalışmalarında da kullanılmaktadır. Çünkü veri olarak kullanılan metin bir Twitter mesajı, Facebook durumu, cümle, paragraf, köşe yazısı, istenmeyen mesaj veya e-posta vb. gibi metin tabanlı içerik olabilir (Grimmer and Stewart 2013). MS ve DA çalışmalarında kullanılan öznitelikler (feature) ise metin veya sosyal medyada paylaşılan mesaj içeriklerinden çeşitli modeller kullanılarak elde edilir. MS çalışmalarında yaygın olarak kullanılan ve öznitelik olarak kelimelerin kullanıldığı model kelime torbası (BoW: Bag of Words) modeli olarak adlandırılır. BoW modelde kelimelerin metin içeriğindeki sırası önemli değildir.

Kelimelerin dilbilimsel özellikleri ile değil metin içeriğinde gözlenme frekansları ile ilgilenilir. Ancak bu modelde öznitelik olarak kelimeler kullanıldığı için dil bağımlıdır ve metnin dile bağımlı önişlemlerden geçirilmesi gerekir. Öznitelik olarak metin içerisinden çıkarılan karakter katarlarının kullanıldığı model ise karakter seviye n-gram (Ngram) model (bag of character n-grams) olarak adlandırılır. Bu model gramer hatalarına, kısaltma ve noktalama işareti kullanımlarına karşı güçlüdür. Ayrıca karakter tabanlı Ngram model önişlem gerektirmez ve dilden bağımsızdır (Cavnar and Trenkle 1994; Kanaris *et al.* 2007).

Öznitelik olarak kelime veya birden fazla kelimenin kullanıldığı model ise kelime seviye Ngram (word-grams) olarak adlandırılır. Bunun yanı sıra yapısal ve istatistiksel model (SSTF: Structured and Statistical Text Features) kendine özgü biçimsel ve istatistiksel özellikleri olan yapılandırılmış veya yarı yapılandırılmış metinlerin (köşe yazısı, şarkı sözü vb.) kullanıldığı yazar tanıma, metin türü tanıma, müzik türü sınıflandırması vb. gibi çalışmalarda yaygın olarak kullanılmaktadır (Stamatatos *et al.* 2000; Diri and Amasyali 2003; Mayer *et al.* 2008). Bu modelde metnin kendine özgü yapısal özellikleri ve çeşitli istatistiksel bilgiler öznitelik olarak kullanılmaktadır.

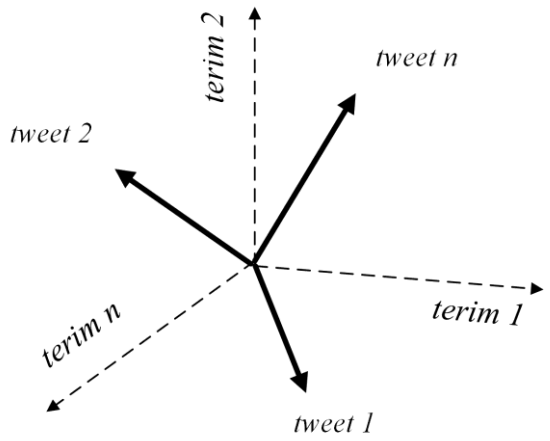
Yapısal öznitelikler elde edilirken metnin yapısına bağlı olarak değişebilen özellikler öznitelik olarak değerlendirilir. İstatistiksel öznitelikler ise problemin türü ve çalışmanın amacına göre çeşitli dilbilimsel işlemler kullanılarak da elde edilebilir. MS

ve DA alanında öznitelikler hangi model ile elde edilmiş olursa olsun terim olarak da ifade edilir. Öznitelikler elde edildikten sonra önişlemin son aşamasında terim ağırlıklandırma (TW: Term Weighting) işlemi uygulanır. Terim ağırlıklandırma (TA) ile her bir terim; ilgili terimin önemini ölçen ve gözlendiği dokümanın sınıflandırılmasına yaptığı katkıyı belirten bir ağırlık değeri ile ilişkilendirilir (Patra and Singh 2013). TA sınıflandırma başarısında doğrudan etkilidir ve bir terim ağırlıklandırılırken terim frekansı faktörü (TF: Term Frequency), ters doküman frekansı faktörü (IDF: Inverse Document Frequency) ve normalizasyon (N: Normalization) faktörü olmak üzere üç farklı bileşenden faydalanılır (Salton and Buckley 1988).

Çizelge 2.2. Yerel ve global kapsamlı bazı ağırlıklandırma yöntemleri

Yerel Kapsam	**Global Kapsam**
İkili (Binary or Boolean)	Ters Doküman Frekansı (IDF)
Terim Frekansı (TF)	Olasılıklı IDF (Probabilistic)
Logaritmik TF (Logarithm)	Global Frekans-IDF (GF-IDF)
Alternatif LTF (Alternate Logarithm)	Entropi (Entropy)

Bu bileşenlerden TF faktörü bir *t* teriminin bir *d* dokümanındaki ağırlığını (local weight), IDF veri setindeki ağırlığını (global weight) temsil eder. Ağırlıklandırma yöntemleri bu bileşenler kullanılarak geliştirildiği için literatürde, genellikle yöntemde kullanılan bileşene göre yerel (local) ve global kapsamda değerlendirilir (Chisholm and Kolda 1999). Literatürde kullanılan bazı yerel ve global kapsamlı ağırlıklandırma yöntemleri Çizelge 2.2'de verilmiştir (Polettini 2004). Bu yöntemlerin dışında yaygın olarak kullanılan ve TF ile IDF bileşenlerinin birleştirilmesiyle ortaya çıkan terim frekansı-ters doküman frekansı (TF-IDF) ve türevi yöntemler de kullanılmaktadır. Terimler ağırlıklandırıldıktan sonra önişleme süreci tamamlanır ve önişlemden geçirildikten sonra karmaşıklığı azaltmak ve sayısallaştırma işlemini kolaylaştırmak amacıyla özniteliklerin bir vektör ile temsil edilmesi (indexing) işlemi gerçekleştirilir (Korde and Mahender 2012).



Şekil 2.4. Vektör uzay modeli

Dokümanlar indekslenirken yaygın olarak kullanılan yöntem ise Şekil 2.4'te verilen vektör uzay modelidir (Salton *et al.* 1975). Vektör uzay modelinde (VSM: Vector Space Model) her bir doküman örneği içerdiği terimlerin ilişkilendirildiği ağırlık değerlerinden oluşan bir vektör ile temsil edilir. Veri seti önişlemden geçirilip indekslendikten sonra öznitelik uzay boyutunun çok yüksek olduğu durumlarda boyut indirgeme (DR: Dimensionality Reduction) işlemi uygulanır ve verinin boyutu düşürülür.

2.3.2. Öznitelik uzay boyutu indirgeme

Metin tabanlı bilgi (textual information) miktarının çok fazla artmasıyla birlikte indeksleme ve özetleme gibi yöntemler kullanılmadan doküman içeriklerinden anlamlı bilgi çıkarmak oldukça zorlaşmıştır. Bu sorunun çözümlerinden biri olan MS çalışmalarının en büyük problemi veya zorluğu ise öznitelik uzayının çok yüksek boyutlara ulaşmasıdır (Yang and Pedersen 1997). Bu nedenle, MS çalışmalarında yüksek boyut problemini aşmak, sürecin etkinliğini ve doğruluğunu artırmak amacıyla genellikle öznitelik seçme (FS: Feature Selection) ve öznitelik çıkarma (FE: Feature Extraction) yaklaşımları kullanılır (Zheng *et al.* 2004; Sebastiani 2005).

Öznitelik seçme işlemi yüksek boyut sorununu çözmek amacıyla sistem eğitilmeden önce mevcut özniteliklerden veriyi tanımlayan bir öznitelik alt kümesinin seçilmesidir

(Dasgupta *et al.* 2007). Bu işlemin veriyi görselleştirme ve anlamayı kolaylaştırma, ölçüm ve veri depolama gereksinimlerini azaltma, eğitim ve test zamanını kısaltma ve boyut indirgemeyi sağlayarak performansı artırma gibi faydaları bulunmaktadır (Guyon and Elisseeff 2003). Öznitelik çıkarma işleminde ise öznitelik boyutu daha etkili ve daha düşük boyutlu bir alt uzaya geçirilir ve her bir yeni öznitelik orijinal özniteliklerin kombinasyonu olarak elde edilir (Li and Jain 1998). Bu yaklaşımlar ile veriden ilgisiz (bilgi sağlamayan) ve gürültülü öznitelikler elenir ve öznitelik uzay boyutu düşürülür. Bu durumun da sınıflandırıcıya eğitim ve test aşamalarında performans ve zaman bakımından olumlu etkisi olmaktadır. Literatürde kullanılan birçok öznitelik seçme yöntemi olmakla beraber Ki-Kare (CS: Chi-Square), Karşılıklı Bilgi (MI: Mutual Information), Bilgi Kazanımı (IG: Information Gain) ve Doküman Frekansı (DF: Document Frequency) yaygın olarak kullanılan yöntemlerdendir (Yang and Pedersen 1997; Forman 2003).

Temel Bileşen Analizi (PCA: Principal Component Analysis) ve Saklı Anlam İndeksi (LSI: Latent Semantic Analysis) gibi lineer cebir teorisini kullanan yöntemler ise öznitelik çıkarma amacıyla kullanılan yöntemlerdendir (Dash and Liu 2008). Boyut indirgeme işleminin global ve yerel kapsamda olmak üzere iki farklı şekilde uygulanması da mümkündür. Global kapsam veri seti bazında, yerel kapsam kategori bazında (her kategori için ayrı) öznitelik uzay boyutunun düşürülmesini temsil eder. Boyut indirgeme işleminin kapsamı hangi indirgeme yönteminin kullanılacağına etki etmez (Sebastiani 2002).

2.3.3. Makine öğrenmesi ve sınıflandırma

Sınıflandırma işlemi mevcut örneklerin birbirinden yüksek doğruluk derecesinde ayırt edilmeye çalışılmasıdır. Bu işlemin gerçekleştirilebilmesi için gerçek dünyada olduğu gibi bilgisayar bilimlerinde de daha önceden gözlenmiş örneklere ve bu örneklerden elde edilen deneyimlere ihtiyaç duyulmaktadır. Genel anlamda örüntü tanıma problemi olarak tanımlanabilecek bu ve benzeri problemleri çözebilmek amacıyla bilgisayar bilimlerinde istatistik ve diğer bilimlerle bağlantılı olan MÖ bileşen ve yöntemlerini

kullanan sistemler kullanılmaktadır. Otomatik metin sınıflandırıcı sistemlerin oluşturulmasında 80'li yıllarda kullanılan uzman sistemler (expert systems) uzman bilgisi ve el ile tanımlanan çeşitli kurallar gerektirmekteydi.

Sistemin başka bir çalışma alanına uygulanması durumunda ise güncelleme amacıyla sürekli müdahale gereksinimi duyulmaktaydı. Ancak 90'lı yılların başlarından itibaren bilgi işlem kapasitesi ve kaynakların artması ile birlikte geniş kullanım alanı bulan MÖ teknikleri, diğer alanlarda olduğu gibi MS alanında da öne çıkmayı başarmıştır (Tantuğ 2012). Bu nedenle MS alanında, MÖ yöntemleri başarılı bir şekilde uygulanmaya başlanmıştır ve halen yaygın olarak kullanılmaktadır (Sebastiani 1999). DA'da bir MS problemi olarak ele alındığı için MÖ yöntemleri DA çalışmalarında da yaygın olarak kullanılmaktadır.

2.3.3.a. Makine öğrenmesi

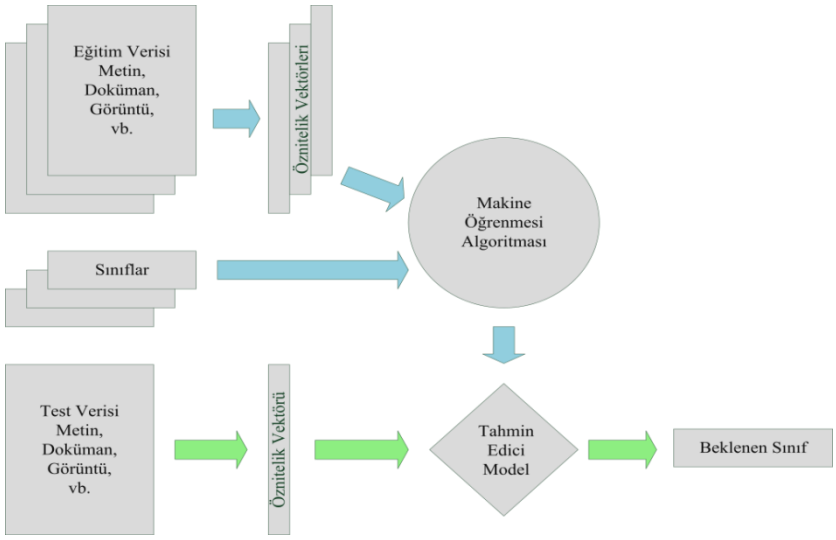
MÖ, önceki gözlemlere dayanarak doğru tahminler yapabilmeyi öğrenebilmek amacıyla otomatik tekniklerin geliştirilmesidir (Schapire 2003). Ayodele (2010)' ye göre ise MÖ, otomatik olarak öğrenme işlemini deneyimlerden yola çıkarak geliştiren ve gerçekleştiren bilgisayar sistemlerinin geliştirilmesidir. MÖ üzerine yapılan araştırmaların artmasıyla birlikte hesaplamalı öğrenme teorisi, yapay sinir ağları, istatistik ve örüntü tanıma gibi araştırma alanları arasında bağlantı kurulmuş ve bu alanlar birlikte çalışılabilmiştir. Böylece MÖ teknikleri daha geleneksel problemlerin (örneğin yüz tanıma) yanı sıra veritabanlarında bilgi keşfi, dil işleme ve robot kontrolü gibi yeni problemlere uygulanmaya başlamıştır (Dietterich 1997).

MÖ'de veri çok önemli bir rol oynamakta ve öğrenme algoritmaları veriye ait bilgi ve özelliklerin keşfedilmesinde kullanılmaktadır (Chao 2011). Kullanılan veri ise etiketli ve etiketsiz olmak üzere iki ayrı türden oluşmaktadır. Etiketli veri seti bir algoritmayı eğitmek, etiketsiz veri ise eğitilmiş algoritmayı (model veya sistemi) test etmek için kullanılmaktadır. Bu nedenle bu veri tipleri eğitim ve test seti olarak da

adlandırılmaktadır. MÖ kullanılarak oluşturulan sistemler genelde iki farklı öğrenme modeli kullanmaktadır. Bu iki model denetimli (supervised) ve denetimsiz (unsupervised) öğrenme modeli olarak adlandırılmaktadır (Gentleman *et al.* 2008). Bunun yanı sıra her iki modelin birlikte kullanıldığı öğrenme sistemleri (örneğin semi-supervised learning) de bulunmaktadır.

2.3.3.b. Denetimli öğrenme

Denetimli Makine Öğrenmesi (SML: Supervised Machine Learning) sistemin etiketli veriler kullanılarak eğitilmesi ile öğrenmenin sağlanmasıdır. Sistem eğitilirken veri setinde bulunan her bir örneğe ait giriş ve çıkışlar verilir. MS çalışmalarında giriş metnin içeriğini, çıkış ise kategorisini temsil eder. Test veri seti ise sistemin doğrulanması amacıyla kullanılır. Sistemin doğrulanması aşamasında öğrenme algoritması kategorisi bilinmeyen bir test verisine, eğitim verisinde bulunan çıkışlardan herhangi birini atar (Kotsiantis *et al.* 2007). Denetimli öğrenme modeli süreci Şekil 2.5'te verildiği gibi gerçekleşmektedir (Afrin and Nahar 2015).



Şekil 2.5. Denetimli öğrenme modeli

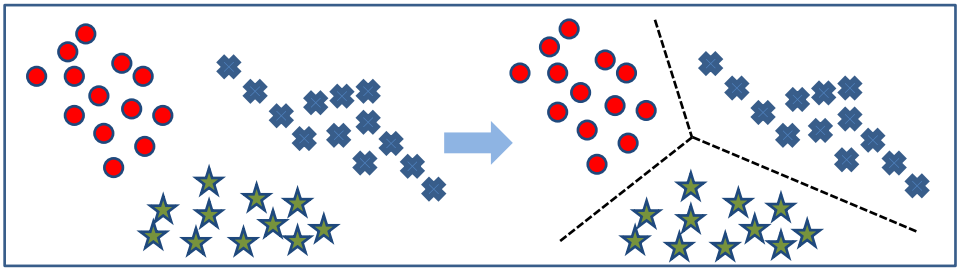
Denetimli öğrenme modelinde problem, sınıflandırma problemi olarak ele alınır ve eğitilmiş sistem test setine yönelik tahmin ve tanıma amacıyla kullanılır (Chao 2011).

Destek Vektör Makinesi, Yapay Sinir Ağları, Lojistik Regresyon, Basit Bayes, Multinom Basit Bayes, *k*-En Yakın Komşu, Rastgele Orman ve Karar Ağaçları algoritmaları denetimli öğrenme modeli oluşturulurken yaygın olarak kullanılan yöntemlerdendir (Caruana and Niculescu-Mizil 2006).

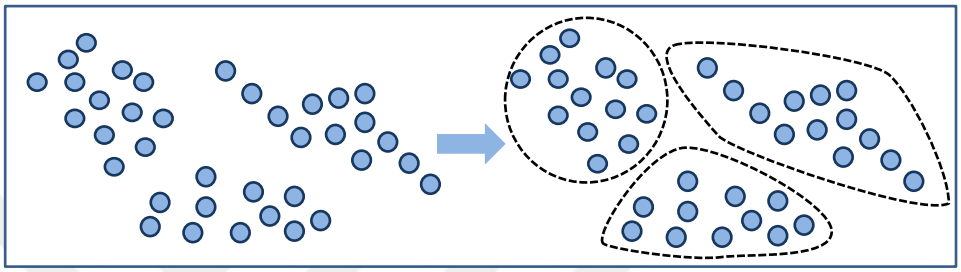
Sınıflandırma problemlerinde kullanılan yöntemler de uygulanan problem veya sonuç beklentisindeki farklılığa göre değişiklik gösterebilmektedir. Bu nedenle sınıflandırma sonuçlarının tek etiketli veya çok etiketli olarak üretilmesi gerekebilmektedir. Tek etiketli sınıflandırmada (single-label) test örneğine mevcut etiket kümesinden sadece bir etiket atanırken, çok etiketli sınıflandırmada (multi-label) birden fazla etiket atanmaktadır. Ayrıca tek etiketli sınıflandırmanın özel bir durumu olan ikili sınıflandırmada (binary classification) ise mevcut iki kategori arasından tek bir etiket atanarak sınıflandırma yapılmaktadır (Sebastiani 2002).

2.3.3.c. Denetimsiz öğrenme

Denetimsiz Makine Öğrenmesi (UML: Unsupervised Machine Learning) modelinde sistem eğitilirken etiketsiz veri kullanılarak öğrenmesi sağlanır. Denetimsiz öğrenmede amaç veri setindeki örneklerin çıkışları bilinmediği için tanıma veya sınıflandırma değildir. Genellikle kümeleme, olasılık yoğunluk tahmini, öznitelikler arasındaki ilişkilerin bulunması ve boyut indirgeme gibi amaçlarla kullanılmaktadır. Ayrıca denetimsiz öğrenme algoritması ile elde edilen sonuçlar denetimli öğrenme için de kullanılabilmektedir (Chao 2011). Parçalayıcı ve hiyerarşik kümeleme algoritmaları ise genellikle denetimsiz öğrenme modeli oluşturulurken kullanılan algoritmalardır (Ozgür 2004). Denetimli ve denetimsiz öğrenme modellerinin kullanım amaçları arasındaki fark Şekil 2.6'da gösterilmiştir (Chao 2011).



(a)



(b)

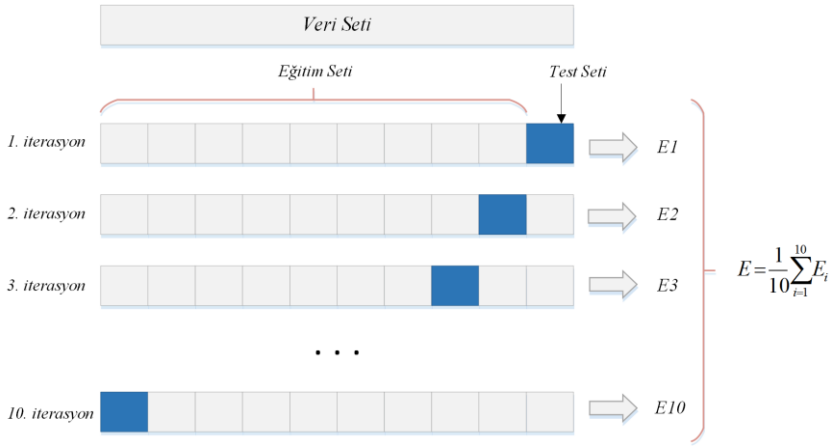
Şekil 2.6. Denetimli öğrenme modeli (a), denetimsiz öğrenme modeli (b)

2.3.3.d. Model geçerleme ve performans metrikleri

MÖ'de model geçerleme, sistemin performansını veya doğruluk derecesini ölçmek amacıyla kullanılacak en uygun modelin seçilmesi işlemidir. Model geçerleme, denetimli öğrenme modelinin kullanıldığı sistemlerde kullanılır. Ancak hem denetimli hem de denetimsiz öğrenme modellerinde sistemin (seçilen veya oluşturulan modelin) performansını ölçmek ve değerlendirmek amacıyla çeşitli performans metrikleri kullanılmaktadır. MS ve DA çalışmaları denetimli öğrenme modeli kapsamında yapılan çalışmalar olduğundan bu başlık altında denetimli öğrenme modeli için model geçerleme ve performans metrikleri açıklanmıştır. Model geçerlemede amaç sistemin en düşük sınıflandırma hatası ile sınıflandırma yapabilmesini sağlayan modeli seçmektir. Temel olarak N toplam örnek sayısı olmak üzere; $N - K$ adet örnek (base set) sistemin eğitiminde, K adet örnek (validation set) ise sistemin test edilmesinde kullanılmak üzere iki parçaya ayrılır. Sınıflandırma hatası en düşük olan model geçerleme modeli olarak seçilir.

Model geçerleme amacıyla kullanılan çapraz geçerleme (CV: Cross Validation) ve birini dışarda bırakma (LOO: Leave One Out) yöntemlerinin yanı sıra Holdout ve Bootstrapping gibi farklı yöntemler bulunmaktadır. Holdout yönteminde veri belirli bir oranda iki parçaya bölünür. Bu bölme işleminde genellikle verinin 2/3'ü eğitim, 1/3'ü test seti olarak alınarak eğitim ve test seti oluşturulur. Bootstrapping yöntemi genellikle küçük boyutlu, örnek sayısının az olduğu veri setlerinde kullanılır. Temel olarak veri setindeki örneklerin kopyalanarak (bir ya da daha fazla kez) rastgele yeni bir örnek kümesine eklenmesi mantığına dayanır. Orijinal veriler test seti, elde edilen yeni veri kümesi ise eğitim seti olarak kullanılır (Kohavi 1995; Alpaydin 2014).

Birini dışarda bırakma yönteminde bir örnek test seti, geri kalan $N-1$ örnek ise eğitim seti olarak alınır. Ancak birini dışarda bırakma yöntemi performans bakımından dezavantajlıdır ve en yaygın olarak kullanılan yöntem çapraz geçerleme yöntemidir (Kotsiantis *et al.* 2007). Bu nedenle bu tez kapsamında deneysel sonuçlar elde edilirken model geçerleme yöntemi olarak n-kat çapraz geçerleme (n-fold cross validation) yöntemi kullanılmıştır. n-kat çapraz geçerleme yönteminde ise veri seti rastgele n tane alt kümeye bölünür. Bu kümelerden bir tanesi test seti, $n-1$ tanesi ise eğitim seti olarak alınır ve genellikle $n=10$ kabul edilir. Bu işlem n defa tekrarlanır ve elde edilen n adet sonucun ortalaması modelin başarısı olarak kabul edilir. Şekil 2.7'de 10-kat çapraz geçerleme modeli gösterilmiştir.



Şekil 2.7. 10-kat çapraz geçerleme

MÖ yöntemlerinin etkinliğini değerlendirmek amacıyla ise özellikle iki sınıflı sınıflandırma problemlerinde kullanılabilecek birçok metrik önerilmiştir. Ancak kesinlik (precision), duyarlılık (recall) ve doğruluk (accuracy) metrikleri yaygın olarak kullanılan değerlendirme yöntemleridir (Ikonomakis *et al.* 2005). Bu değerlerin elde edilebilmesi için iki sınıflı bir sınıflandırma probleminde muhtemel olan dört farklı durumun bilinmesi gerekmektedir (Alpaydin 2014). Bu durumlar Çizelge 2.3'te verilen karışıklık (confusion) veya olasılık (contingency) tablosu ile elde edilmektedir (Sebastiani 2002).

Çizelge 2.3. Karışıklık matrisi

Kategori (c)		**Örnek Kategorisi**	
		Evet	Hayır
Sınıflandırıcı Kararı	Evet	TP	FP
	Hayır	FN	TN

Çizelge 2.3'te verilen tablo sınıf sayısının ikiden fazla olduğu durumlar için de oluşturulabilir ve böylece sınıf sayısı kadar satır ve sütun içeren bir kare matris elde edilebilir (Alpaydin 2014). Tabloda verilen TP (True Positive) kategori ataması doğru yapılan, FP (False Positive) kategori ataması yanlış yapılan, FN (False Negative) kategorisi yanlış reddedilen ve TN (True Negative) ise kategorisi doğru reddedilen örnek sayısını temsil eder. Yukarıda açıklanan durumlar kullanılarak MÖ ve MS çalışmalarında yaygın olarak kullanılan recall (r), precision (p), fallout (f), accuracy (Acc) ve error (Err) metrikleri aşağıda verilen eşitlikler ile elde edilir (Yang 1999).

$$r = TP/(TP + FN) \tag{2.1}$$

$$p = TP/(TP + FP) \tag{2.2}$$

$$f = FP/(FP + TN) \tag{2.3}$$

$$Acc = (TP + TN)/N \tag{2.4}$$

$$Err = (FN + FP)/N \tag{2.5}$$

Yukarıdaki eşitliklerde payda kısmındaki değerler sıfırdan büyük olmak zorundadır ve $N = TP + FP + FN + TN$'dir. Precision rastgele bir örneğin bir c sınıfına atandığı

bilindiğinde; sınıfın doğru olma koşullu olasılığını yani sınıflandırıcının örneği doğru sınıflandırabilme yeteneğini ölçer. Recall rastgele bir örneğin bir *c* kategorisine atanma olasılığını, accuracy doğru sınıflandırma oranını, fallout (false positive rate) yanlış reddedilen atama oranını ve error sınıflandırma işleminin hata oranını temsil eder. Bunların dışında yukarıdaki metriklerin birleştirilmesiyle elde edilen performans metrikleri de mevcuttur (Sebastiani 2002; Ikonomakis *et al.* 2005). MS çalışmalarında kullanılan metriklerden precision, fallout metriğinin bir alternatifidir ve daha yaygın olarak kullanılmaktadır (Lewis and Ringuette 1994). Bir sınıflandırıcının değerlendirme performansı ise sıklıkla doğru tahmin edebilme yeteneğine yani doğruluk metriğine bağlıdır (Kotsiantis *et al.* 2007).

2.4. Metin Sınıflandırma ve Duygu Analizi Alanında Yapılmış Çalışmalar

MS ve DA çalışmaları geniş uygulama alanlarına sahiptir. Literatürde yapılan çalışmalar incelendiğinde metin tabanlı verilerin otomatik bir şekilde sınıflandırılarak veya sosyal medya ortamından toplanan veriler kullanılarak yapılmış birçok çalışma olduğu görülmektedir. Bu alanlarda yapılan çalışmaların oldukça çok olması ve sosyal medya kullanımına olan ilginin her geçen gün artması özellikle DA olmak üzere metin tabanlı sınıflandırma ve analiz çalışmalarının önemini ortaya koymaktadır. Bu başlık altında, literatür taraması sonucunda incelenen ve her iki alanda daha önce yapılmış akademik çalışmalara yer verilmiştir.

Bu çalışmalar incelendiğinde MS alanında Türkçe, İngilizce ve diğer diller için farklı amaçlar doğrultusunda MÖ yöntemlerinin kullanıldığı çalışmaların bulunduğu görülmektedir. Diri and Amasyalı (2003), 18 farklı yazarın köşe yazıları ile oluşturdukları veri setinden elde ettikleri 22 biçimsel ve istatistiksel özniteliği yazar tanıma amaçlı kullanmışlardır. Bu özniteliklerden en etkili olan 4 özniteliği seçerek, yapay sinir ağları yöntemiyle köşe yazılarının otomatik yazar tespitinde %84 başarı elde etmişlerdir. Kaşıkçı ve Gökçen (2014), otomatik olarak e-ticaret sitelerini belirlemeye çalışmış ve Basit Bayes algoritmasıyla %85,30 başarı elde etmişlerdir.

Güran vd (2009), web ortamından topladıkları otomobil, politika, tıp, magazin, ekonomi ve spor kategorilerindeki dokümanları unigram, bigram ve trigram seviyelerinde Ngram model kullanarak sınıflandırmışlardır. Sınıflandırma aşamasında farklı sınıflandırıcıların performansını karşılaştırmış ve unigram modelde en yüksek başarıyı (%95,83) Multinom Basit Bayes ile elde etmişlerdir. Doğan ve Diri (2010), Türkçe köşe yazılarını toplayarak üç farklı veri seti oluşturmuş; yazar, tür ve cinsiyet sınıflandırması amacıyla kullanmıştır. Öznitelikleri karakter seviye Ngram modelde bigram, trigram ve four-gram olmak üzere üç farklı şekilde elde etmişlerdir. Ayrıca çalışmada kullanılan diğer sınıflandırma yöntemlerinden daha başarılı olan Ng-ind isminde yeni bir yöntem önermişlerdir. En yüksek başarıyı ise doküman türü belirlemede, önerdikleri Ng-ind yöntemi ile %93,8 olarak elde etmişlerdir.

Amasyalı vd (2012), farklı amaçlar doğrultusunda kullanılabilecek altı farklı Türkçe veri seti üzerinde metin temsilinde kullanılan özniteliklerin performans karşılaştırmasını yapmıştır. Deneysel sonuçlar Ngram özniteliklerinin metin temsilinde diğer özniteliklere kıyasla daha başarılı olduğunu göstermiştir. Yıldız vd (2007), MS çalışmaları için yeni bir öznitelik çıkarım modeli önermiştir. Önerdikleri modelde, geleneksel kelime torbası modelinin aksine öznitelik vektörlerinin sınıf sayısı boyutunda olabileceğini ve geleneksel yöntemden daha başarılı olduğunu göstermişlerdir. Deneysel sonuçlar MÖ yöntemleri ile elde edilmiş ve %96,25 oranında Bayes yöntemi en başarılı sonucu vermiştir.

Tüfekçi vd (2012), web ortamından toplanarak oluşturulan Türkçe haber metinleri üzerinde boyut indirgeme ve gövdeleme yaklaşımlarının etkisini incelemişlerdir. Gövdeleme işleminden sonra elde edilen kelimelerden uzun gövdeli olanların daha etkili olduğunu ve uygulanan boyut indirgeme yöntemleriyle öznitelik boyutunun %97,46 oranında düşürüldüğünü tespit etmişlerdir. Sınıflandırma aşamasında geleneksel yöntemler kullanılmış ve en yüksek %93,73 oranında Bayes yöntemi ile sınıflandırma başarısı elde etmişlerdir. Torunoğlu vd (2011) ise Türkçe haber metinlerinden oluşan veri setleri üzerinde önişleme (durak kelime çıkarımı, gövdeleme vb.) ve TA tekniklerinin sınıflandırma sonuçları üzerindeki etkisini incelemişlerdir.

İngilizce ve diğer dillerde yazılmış metinler üzerinde de literatürde daha önce yapılmış çeşitli çalışmalar mevcuttur. Bu bağlamda taranan çalışmalardan bir kısmı yine aşağıda açıklanmıştır. Nigam *et al.* (2000), Beklenti-Maksimizasyon (EM: Expectation–Maximization) algoritmasıyla İngilizce metinler üzerinde MS çalışmış sistemin eğitim aşamasında belirli oranda etiketsiz veriler kullanarak performansı artırmayı hedeflemişlerdir. Veri seti olarak MS alanında yaygın olarak kullanılan 20 Newsgroups, WebKB ve Reuters veri kümelerini kullanmışlardır. Jiang *et al.* (2012) MS çalışmaları için kümeleme tabanlı geliştirilmiş *k*-NN algoritması önermiştir. Önerdikleri sınıflandırıcının Reuters (ModApte) ve diğer iki veri seti üzerinde standart *k*-NN algoritmasından daha başarılı olduğunu göstermişlerdir.

Lin *et al.* (2014), MS ve kümeleme için yeni bir benzerlik ölçüm yöntemi önermiş, *k*-NN ve *k*-Means yöntemleri ile test etmişlerdir. WebKB, Reuters-8 ve RCV1 veri setleri üzerinde yapılan testlerde önerilen benzerlik yönteminin literatürde kullanılan diğer yöntemlerden daha başarılı olduğu gösterilmiştir. Ko (2012), sınıf bilgisini kullanarak MS'de farklı TA yöntemlerini karşılaştırmış ve önerdiği yöntemin daha başarılı olduğunu göstermiştir. Armagon *et al.* (2007), 19. yy. yazarlarının eserlerinden oluşan veri kümesi üzerinde çeşitli (yazarlık özelliği, yazar cinsiyeti, kitabın bilimsel etkinliği vb. gibi) sınıflandırma işlemleri uygulamışlardır. Bu sınıflandırmaları yaparken literatürde yaygın olarak kullanılan SSTF öznitelikleri yerine elde ettikleri fonksiyonel sözcükleri öznitelik olarak kullanmışlardır.

Çizelge 2.4. MS alanında daha önce yapılmış bazı çalışmalar

Yazar	**Konu**	**Öznitelik Modeli**	**Yöntem**	**Başarı Oranı (%)**	**Yıl**
Diri ve Amasyalı	Yazar Tanıma	SSTF	YSA	84,00	2003
Kaşıkçı ve Gökçen	E-Ticaret Sitelerinin Belirlenmesi	BoW	NB	85,30	2014
Güran vd	Metin Sınıflandırma	Ngram	MNB	95,83	2009
Doğan ve Diri	Yazar, Tür ve Cinsiyet Sınıflandırma	Ngram	Ng-ind	93,80	2010
Amasyalı vd	Metin Temsilinde Kullanılan Özniteliklerin Performans Karşılaştırması	Farklı Öznitelik Grupları	DVM *vd.*	99,62	2012
Yıldız vd	Metin Sınıflandırma için Yeni Bir Öznitelik Çıkarım Modeli	Önerilen Model	Bayes *vd.*	96,25	2007
Tüfekçi vd	Haber Metinlerinin Sınıflandırılmasında Türkçe Dilbilgisi Özelliklerinin Etkisi	BoW	Bayes *vd.*	92,73	2012
Torunoğlu vd	Önişleme Tekniklerinin Türkçe Metinlerin Sınıflandırılmasına Etkisi	BoW	MNB *vd.*	-	2011
Nigam *et al.*	Etiketli ve Etiketsiz Dokümanlar ile Metin Sınıflandırma	BoW	EM	-	2000
Jiang *et al.*	Metin Sınıflandırma için Geliştirilmiş *k*-NN algoritması	BoW	*k*-NN	-	2012
Lin *et al.*	Metin Sınıflandırma ve Kümeleme için Yeni Bir Benzerlik Metriği	BoW	*k*-NN, *k*-Means	-	2014
Ko	Terim Ağırlıklandırma Yöntemlerinin Karşılaştırılması	BoW	*k*-NN, SVM	94,90	2012
Armagon *et al.*	Fonksiyonel Sözcük Öznitelikleri ile Stilistik Metin Sınıflandırma	BoW	SVM	-	2007

MS alanında yukarıda açıklanan çalışmaların dışında farklı amaçlar doğrultusunda gerçekleştirilen ve MÖ tekniklerinin kullanıldığı birçok çalışma mevcuttur. Ancak bu bölümde, incelenen bu çalışmalardan sadece bir kısmı açıklanmış ve özet bilgiler Çizelge 2.4'te verilmiştir.

Bu tezin konusu olan DA için de literatürde yapılan çalışmalar incelenmiş ve bir kısmı yine aşağıda açıklanmıştır. DA çalışmaları incelendiğinde ise bu alanda yapılan çalışmaların genellikle Twitter üzerinde yoğunlaştığı görülmektedir. Literatürde yapılan DA çalışmaları bu alana özgü yöntemlerin yanı sıra klasik MS teknikleri ile gerçekleştirilmektedir.

Kaya vd (2012), Türkçe politik haber metinleri üzerinde DA çalışmış, haberlerin olumlu veya olumsuz eleştiri içerip içermediğini tespit etmişlerdir. Öznitelikler Ngram modelde elde edilmiş ve etkili olduğu tespit edilen kelime öznitelikleri ile birlikte kullanılmıştır. Farklı sınıflandırıcılar ile yapılan deneylerde yüzde [65-77] aralığında başarı elde edilmiştir. Türkmenoğlu ve Tantuğ (2014), Twitter mesajları ve film yorumları üzerinde DA çalışmıştır. MÖ tabanlı (machine learning based) ve sözlük tabanlı (lexicon based) DA testlerinde MÖ tabanlı yöntemin daha etkili olduğunu göstermişlerdir.

Küçük ve Steinberger (2014), Türkçe Twitter mesajları üzerinde geliştirilmesi muhtemel bir Adlandırılmış Varlık Tanıma (NER, Named Entity Recognition) sistemi için yol gösterici olabilecek testler gerçekleştirmiştir. Go *et al.* (2009), Twitter mesajları üzerinde DA çalışmış, en yüksek %82,2 oranında başarı elde etmişlerdir. Kelime seviye Ngram modelin kullanıldığı çalışmada, Twitter mesajları his simgeleri kullanılarak etiketlenmiştir. Kouloumpis *et al.* (2011), Twitter mesajları üzerinde farklı veri setleri kullanarak DA çalışmıştır. Farklı özniteliklerin kullanıldığı çalışmada kullanılan özniteliklerin ve veri setinin sonuçlar üzerindeki etkisi incelenmiştir.

Bollen *et al.* (2011), Twitter mesajlarının DA'sı ile Dow Jones Industrial Average (DJIA) borsa değerini tahmin etmeye çalışmışlardır. Çalışmada Ngram model ile öznitelik çıkarılmış ve %86,7 oranında başarı elde edilmiştir. Pak and Paroubek (2010), Twitter mesajları üzerinde yapılan ilk ve önemli DA çalışmalarından biri olan çalışmalarında, his simgelerini kullanarak topladıkları mesajları etiketlemiştir. Çalışmada Ngram ve POS özniteliklerini kullanmışlar ve geliştirdikleri duygu sınıflandırıcısının önceki yöntemlerden daha iyi sonuç verdiğini göstermişlerdir.

Coban vd (2015a), Türkçe Twitter mesajları üzerinde DA gerçekleştirmiş ve temel MS yöntemleri ile %66,06 oranında başarı elde etmişlerdir. Çalışmada BoW ve Ngram model ile öznitelik çıkarılmış ve mesajlar his simgeleri kullanılarak etiketlenmiştir. Boiy *et al.* (2007), DA'da kullanılan teknikleri incelemiş ve MÖ yöntemleri ile otomatik olarak DA gerçekleştirmişlerdir. Çalışmada film eleştirileri ve kişisel blog yazılarından oluşan iki farklı veri seti kullanılmış, öznitelikler Ngram model ile elde edilmiştir.

Martineau and Finin (2009), DA çalışmalarında kullanılabilecek Delta TF-IDF isimli yeni bir ağırlıklandırma yöntemi önermiş, TF ve TF-IDF yöntemlerinden daha başarılı olduğunu göstermişlerdir. Çalışmada film eleştirilerini içeren veri seti kullanılmış ve öznitelikler Ngram model ile elde edilmiştir. DA testlerinde ise %88,1 oranında başarı elde edilmiştir. Davidov *et al.* (2010), Twitter mesajlarından oluşan veri seti üzerinde his simgeleri ve Twitter etiketlerini (hashtag) kullanarak DA gerçekleştirmiş ve performansı artırmayı hedeflemişlerdir. Ayrıca farklı öznitelik modellerini kullanarak öznitelik modelinin etkisini incelemişlerdir.

Agarwal *et al.* (2011), Twitter veri seti üzerinde unigram, çekirdek (kernel) ve öznitelik tabanlı olarak adlandırdıkları üç farklı model ile DA gerçekleştirmişlerdir. Deneysel sonuçları tasarladıkları ağaç yapısı ve diğer öznitelik modelleri için elde etmiş ve önerdikleri yöntemin literatürde kullanılan temel öznitelik modellerinden daha başarılı olduğunu göstermişlerdir. Pang *et al.* (2002) ise film eleştirilerinden oluşan veri seti üzerinde MÖ yöntemlerini kullanarak DA gerçekleştirmişlerdir. Çalışmada farklı öznitelik modeli ve sınıflandırıcı kombinasyonlarının sonuçlar üzerindeki ektisi de incelenmiştir.

Çizelge 2.5. DA alanında daha önce yapılmış bazı çalışmalar

Yazar	**Konu**	**Öznitelik Modeli**	**Yöntem**	**Başarı Oranı (%)**	**Yıl**
Kaya vd	Türkçe Politik Haber Metinleri Üzerinde Duygu Analizi	Ngram	SVM *vd.*	77,00	2012
Türkmenoğlu ve Tantuğ	Türkçe Medya Verileri için Duygu Analizi	Ngram, BoW	SVM *vd.*	89,50	2014
Küçük ve Steinberger	Türkçe Twitter Mesajları Üzerinde NER Performansının Artırılması	BoW	Kural Tabanlı NER Sistemi	-	2014
Go *et al.*	Twitter Duygu Sınıflandırması	Ngram, POS	SVM *vd.*	82,20	2009
Kouloumpis *et al.*	Twitter Duygu Analizi	Ngram, POS *vd.*	AdaBoost	~75,00	2011
Bollen *et al.*	Twitter Duygu Analizi ile Borsa Değer Tahmini	Ngram	Self Organizing Fuzzy Neural Network *vd.*	86,70	2011
Pak and Paroubek	Fikir Madenciliği ve Duygu Analizi için Veri Kaynağı Olarak Twitter	Ngram, POS	NB, SVM	-	2010
Coban vd	Türkçe Twitter Mesajlarının Duygu Analizi	BoW, Ngram	MNB *vd.*	66,06	2015
Boiy *et al.*	Çevrimiçi Metinler için Otomatik Duygu Analizi	Ngram, POS	ME *vd.*	87,40	2007
Martineau and Finin	Delta TFIDF ile Duygu Analizi için Geliştirilmiş Öznitelik Uzayı	Ngram	SVM	88,1	2009
Davidov *et al.*	Twitter Etiketleri ve His Simgelerini Kullanarak Duygu Öğrenme	Ngram, BoW *vd.*	*k*-NN	-	2010
Agarwal *et al.*	Twitter Verilerinin Duygu Analizi	Ngram *vd.*	SVM	75,39	2011
Pang *et al.*	Makine Öğrenmesi Teknikleri ile Duygu Sınıflandırması	Ngram, POS	SVM *vd.*	82,90	2002
Coban ve Ozyer	Türkçe Twitter Mesajları için LDA ile Duygu Sınıflandırması	BoW, Ngram	MNB *vd.*	92,50	2016

DA alanında yukarıda açıklanan çalışmaların dışında farklı amaçlar doğrultusunda gerçekleştirilmiş birçok çalışma mevcuttur. Ancak bu bölümde, incelenen bu çalışmalardan sadece bir kısmı açıklanmış ve özet bilgiler Çizelge 2.5'te verilmiştir.

3. MATERYAL ve YÖNTEM

Bu bölümde bu tez kapsamında gerçekleştirilen çalışmalarda kullanılan materyal ve yöntemler açıklanmıştır. MS ve DA çalışmaları MÖ teknikleri kullanılarak gerçekleştirildiği için bu bölümde bu tezde kullanılan MÖ yöntemlerinin açıklanmasına gerek görülmüştür. Materyal (veri setleri) başlığı altında, Türkçe DA çalışmalarında kullanılabilecek standart bir veri kümesi mevcut olmadığından bu tez kapsamında oluşturulan Türkçe Twitter veri setinin yanı sıra deneylerde kullanılan diğer veri setleri açıklanmıştır. Yöntem başlığı altında ise MS ve DA süreçlerini otomatikleştirme amacıyla geliştirilen OMESIS yazılımında kullanılan sistemin mimarisi ve alt yapısı açıklanmıştır. Ayrıca bu sistemin sırasıyla önişleme, boyut indirgeme ve sınıflandırma aşamalarında bu tezde kullanılan yöntem ve tekniklerin uygulanması detaylı bir şekilde anlatılmıştır. Son olarak ise OMESIS yazılımı ile ilgili bilgiler ve grafik arayüzü açıklanmıştır.

3.1. Veri Setleri

Çalışmamızda temelde ortak yönleri oldukça fazla olsa da iki farklı konu (MS ve DA) üzerinde çalışılmış ve deneyler yapılmıştır. Bu yüzden farklı dil ve yapıya sahip veri setleri kullanılmıştır. Bu bağlamda literatürde MS alanında İngilizce dili için oldukça yaygın olarak kullanılan Reuters-21578, WebKB (Anonymous 2001) ve RCV1 (Lewis *et al.* 2004) veri setlerinden Reuters-21578 verisinin (Anonymuos 1987a) bir alt kümesi olan Reuters-8 (Anonymous 1987b) verisi (R8) kullanılmıştır. DA deneylerinde ise sorgu anahtarı olarak his simgelerini kullanarak oluşturduğumuz Türkçe Twitter mesajlarından oluşan veri seti (TTM) kullanılmıştır. MS teknikleri kullanılarak yapılan istenmeyen mesajların tespiti amaçlı çalışmada (Bozan vd 2015) İngilizce kısa mesajlardan oluşan veri seti (SpamSMSCollection) kullanılmıştır. Yine bu tez kapsamında gerçekleştirilen şarkı sözlerinden müzik türü sınıflandırması amaçlı çalışmada ise Türkçe Şarkı Sözleri (TSS) veri kümesi kullanılmıştır.

3.1.1. Reuters-8

Reuters tarafından 1987 yılında yayınlanan, el ile sınıflandırılmış İngilizce dokümanları içeren ve kısaca Reuters-21578 olarak adlandırılan Reuters-21578 ModeApte Split Text Categorization Test Collection veri kümesi içerdiği doküman sayısından dolayı 21578 sayısı ile isimlendirilmiştir. Veri seti herhangi bir konuyla ilgisi olmayan veya bir ya da birden fazla konuyla ilgili olan dokümanlar içermektedir. Bu dokümanların veri setinden çıkarılıp eğitim ve test verisi şeklinde ayrılmasıyla ModeApte Split versiyonu elde edilmiştir. Reuters-21578 veri setinde dokümanlar dengeli bir dağılım göstermediği için sık gözlenen kategorilerdeki dokümanlar alınarak daha dengeli alt koleksiyonlar oluşturulmuştur. MS çalışmalarında içerdiği kategori sayısı (dokümanların daha sık gözlendiği) ile isimlendirilen R90 (ModApte), R52, R10 ve R8 gibi alt koleksiyonlar kullanılmaktadır. En çok kullanılan alt koleksiyon ise toplam 90 kategoride 12902 doküman içeren R90'dır.

Çizelge 3.1. R8 verisi için sınıf-doküman dağılımı

Sınıf	**Eğitim Dokümanı Sayısı**	**Test Dokümanı Sayısı**	**Toplam**
acq	1596	696	2292
crude	253	121	374
earn	2840	1083	3923
grain	41	10	51
interest	190	81	271
money-fx	206	87	293
ship	108	36	144
trade	251	75	326
Toplam	**5485**	**2189**	**7674**

Bu tez kapsamında yapmış olduğumuz çalışmada (Coban vd 2015b) veri seti olarak R90 versiyonunda bulunan ve en çok kullanılan 8 kategori ile sadece tek bir konu ile ilgili olan dokümanlar alınarak elde edilen R8 versiyonu kullanılmıştır. Bu yöntemle kategori başına düşen doküman sayısı bakımından daha dengeli bir dağılım gösteren daha düşük boyutlu bir veri seti elde edilmiştir. R8 veri setine ait sınıf doküman dağılımına ilişkin sayısal bilgiler Çizelge 3.1'de verilmiştir.

3.1.2. SpamSMSCollection

Bu çalışma kapsamında yapılan deneylerde kullanılan bir diğer veri seti SpamSMSCollection (Anonymous 2006) veri setidir. İstenmeyen mesaj (Spam SMS) filtreleme çalışmaları için veri seti elde etmek önemli bir problemdir. Ayrıca bu konuda Türkçe kısa mesajlar üzerinde yapılması muhtemel akademik çalışmalarda kullanılabilecek standart bir veri seti henüz mevcut değildir. Bu nedenle, çalışmamızda önerilen sistemin filtreleme başarısını değerlendirmek için SpamSMSCollection verisi kullanılmıştır (Almeida *et al.* 2011). Veri seti spam olup olmadığı etiketlenmiş toplam 5574 adet İngilizce mesaj içermektedir. Veri setiyle ilgili detaylı istatistiksel bilgiler sırasıyla Çizelge 3.2 ve Çizelge 3.3'te verilmiştir.

Çizelge 3.2. SpamSMSCollection verisi sınıf-doküman dağılımı

Sınıf	**Mesaj Sayısı**	**Yüzdesi (%)**
Normal (ham)	4827	86,6
İstenmeyen (spam)	747	13,4
Toplam	**5574**	**100**

Çizelge 3.3. SpamSMSCollection verisi öznitelik istatistikleri

Özellik	**Sınıf**	
	Normal (ham)	**İstenmeyen (spam)**
Öznitelik sayısı	17543	63632
Ortalama öznitelik sayısı	13,18	23,48

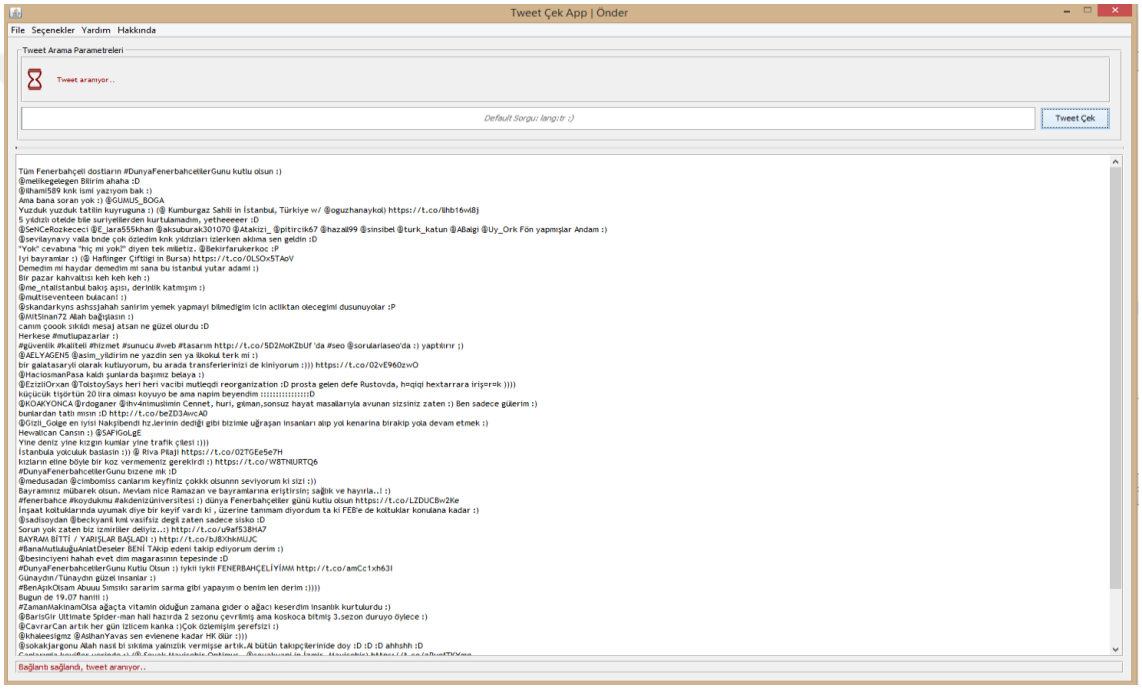
3.1.3. Twitter

Twitter günümüzde en popüler sosyal medya ortamlarından birisidir. Dünya üzerinde birçok farklı ülke ve kültürden yaklaşık 302 milyon kullanıcı kitlesine sahip olan Twitter (Anonymous 2015a), DA alanında yapılan akademik çalışmalar için de zengin bir veri kaynağı sunmaktadır. Bu bağlamda sosyal medya üzerinde yapılan çalışmalara katkıda bulunmak amacıyla Twitter tarafından, araştırmacıların elde edecekleri içerikleri çalışmalarında kullanabilmelerini sağlamak amacıyla Twitter API (Anonymous 2015b) sunulmuştur.

3.1.3.a. Twitter API

Twitter uygulama geliştirme arayüzü (API: Application Programming Interface), hitap ettiği araştırmacı kitlesi ve yetenekleri bakımından farklı özelliklere sahip Search, Rest ve Stream kütüphanelerini barındırmaktadır (Anonymous 2015c). Bunların dışında geliştiriciler tarafından sunulan ve bu kütüphaneleri kullanan harici üçüncü parti yazılımlar da mevcuttur. Bu yazılımlar sayesinde Twitter ortamından mesaj (tweet) toplamak oldukça kolaylaşmıştır. Ayrıca mesaj toplamak dışında Twitter mesaj paylaşımı, herhangi bir kullanıcının gönderdiği mesajları listeleyebilme, çeşitli parametrelere (dil, konum, tarih vb.) bağlı olarak sorgu gönderme ve benzeri gibi birçok işlemi gerçekleştirmek mümkündür. Gönderilen sorgular herhangi bir kelime, terim veya hashtag içerebilmektedir. Ancak Twitter'ın güvenlik ve gizlilik koşulları gereği sadece herkese açık olarak paylaşılan (public) mesajlara erişmek mümkün olmaktadır. Bu tez çalışması kapsamında DA deneylerinde kullanılan veriler, yukarıda bahsedilen nedenlerden dolayı popüler bir sosyal medya ortamı olan Twitter üzerinden, Twitter API kullanılarak toplanmıştır. Ayrıca aşağıda verilen nedenler de veri seti oluşturmak için mecra olarak Twitter'ın tercih edilmesinde etkili olmuştur:

- Twitter kullanıcıların herhangi bir konu ile ilgili fikir, düşünce veya deneyimlerini, “tweet” olarak adlandırılan ve en fazla 140 karakterden oluşan mesajlar aracılığıyla, paylaşabildikleri popüler sosyal ağlardan birisidir.
- Paylaşılan mesajlar kullanıcıların farklı konular ile ilgili fikir ve duygularını içermektedir.
- Farklı kültür ve seviyeden milyonlarca kullanıcı kitlesine sahip olduğu için farklı dillerde ve içeriklerde mesajların toplanması mümkündür.



Şekil 3.1. Twitter API tabanlı, mesaj toplama amaçlı uygulamanın genel görünümü

3.1.3.b. TTM

TTM veri seti bu tez kapsamında yapılan DA çalışmasında (Coban vd 2015a) kullanılmış veri setidir. Veri seti popüler bir sosyal medya ortamı olan Twitter ortamından toplanan Türkçe mesajları içermektedir ve bu çalışma kapsamında oluşturulmuştur. DA çalışmalarında kullanılabilecek Türkçe Twitter mesajlarından oluşan ve herkese açık bir Twitter veri seti mevcut değildir. Bu nedenle mesajların his simgeleri kullanılarak etiketlenmesi olarak bilinen yöntemle (Go *et al.* 2009a) Twitter

API kullanılarak Türkçe mesajlar içeren bir veri seti oluşturulmuştur. Twitter API, kendisine gönderilen her bir sorgu için maksimum 100 mesaj alınmasına izin verdiği için geliştirdiğimiz Java tabanlı uygulama (Şekil 3.1) ile daha fazla sayıda mesaj alınması sağlanmıştır. Mesajlar çekilirken hem gönderilen sorgularda hem de çekilen mesajın kategorisinin belirlenmesinde sadece his simgeleri kullanılmıştır. Mesajlar içerdiği his simgesinin bulunduğu gruba göre, pozitif veya negatif olarak etiketlenmiştir (Tang *et al.* 2009; Pak and Paroubek 2010). Anahtar kelime olarak sorgulama aşamasında ve mesajın etiketlenmesinde kullanılan iki gruba ayrılmış his simgeleri ise aşağıda verilmiştir:

- Pozitif grup: “:-)”, “:)”, “=)”, “:D”
- Negatif grup: “:-(”, “:(”, “=(”, “;(”

Mesajlar etiketlenirken bir mesajın ilgili grupta (pozitif veya negatif) bulunan his simgelerinden birisini içermesi, o kategoriye atanması için yeterli koşul olarak kabul edilmiştir. Yukarıda açıklanan işlemler sonucunda, 10000 pozitif ve 10000 negatif kategorili olmak üzere toplam 20000 mesaj içeren TTM veri seti oluşturulmuştur.

3.1.4. TSS

TSS veri seti bu tez kapsamında MS alanında şarkı sözlerinden müzik türü sınıflandırması amacıyla gerçekleştirilen çalışmada kullanılan veri setidir (Coban ve Ozyer 2016b). Müzik türü sınıflandırması amacıyla yapılması muhtemel çalışmalarda kullanılabilecek Türkçe şarkı sözlerinden oluşan standart bir veri kümesi mevcut değildir. Bu nedenle yine bu çalışma kapsamında geliştirilen web örümceği ile otomatik olarak toplanan Türkçe şarkı sözlerinden oluşan TSS veri kümesi oluşturulmuştur.

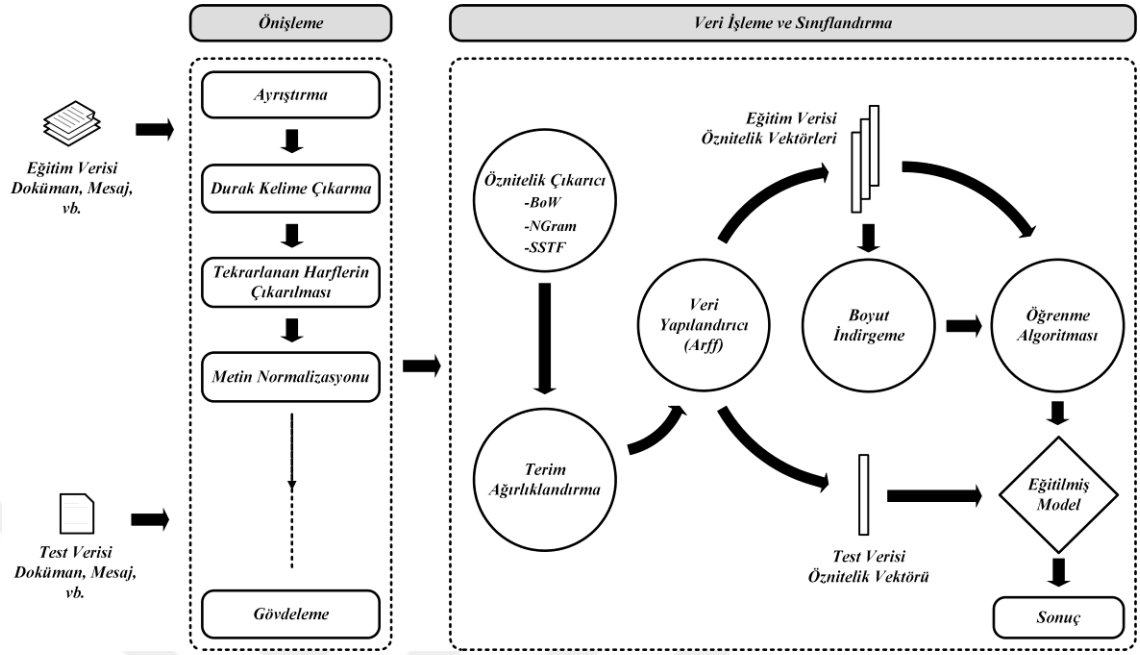
TSS rock, rap, arabesk, pop ve halk müziği türlerinde olmak üzere ilgili türde popüler olmuş 5 farklı sanatçının seslendirdiği şarkı sözlerinden oluşmaktadır. Bir sanatçının hangi müzik türüne dahil edileceğine karar verilirken ise sadece ilgili türde şarkı seslendiriyor olması koşulu aranmıştır. Böylece her bir sanatçının seslendirdiği, rastgele

seçilen 50 şarkı sözü alınarak toplamda 1250 Türkçe şarkı sözü içeren TSS verisi elde edilmiştir.

3.2. Sistemin Altyapısı ve Mimarisi

Çalışma kapsamında geliştirilen OMESIS; metin tabanlı belge, doküman, mesaj vb. gibi dijital içeriklerin işlenmesi gereken farklı uygulama alanlarında kullanılabilmektedir. Sistem geliştirilirken bu tez kapsamında yapılan araştırmalar çerçevesinde geliştirilmiş olmakla beraber metin tabanlı birçok sınıflandırma işlemine cevap verebilecek yetenektedir. Geliştirilen bu sistem ile temelde ortak noktaları bulunduğundan farklı metin işleme çalışmaları yapılabilmektedir. Örneğin sistemimizde hem MS hem DA işlemleri gerçekleştirilebilmekte bunun yanı sıra metin tabanlı herhangi bir dijital içerik üzerinde klasik MS teknikleri kullanılarak sınıflandırma işlemi yapılabilmektedir.

Sistem modeli önişleme, veri işleme ve sınıflandırma olmak üzere üç ana aşamadan oluşmaktadır. Önişleme aşaması kendi içinde dizgeciklere ayırma (tokenization), durak kelimeleri çıkarma (removing stopwords) ve kök bulma (stemming) aşamalarından oluşmaktadır. Veri işleme aşamasında önişlenmiş içerikten modele uygun öznitelik çıkarılmakta daha sonra çıkarılan özniteliklerin ağırlığı belirlenmektedir. Veri işleme aşamasının son adımında ise elde edilen veri ARFF (Attribute-Relation File Format) dosya formatına dönüştürülmektedir. Sistemin son aşaması olan sınıflandırma aşamasında, işlenmiş veri bir öğrenme algoritmasına verilmekte ve eğitilmiş model oluşturulmaktadır. Son olarak ise gelen her bir yeni örnek bu eğitilmiş model ile sınıflandırılmaktadır.



Şekil 3.2. Sistemin akış diyagramı

Veri işleme aşamasında Şekil 3.2'de verilen akış diyagramında gösterildiği gibi veri seti üzerinde boyut indirgeme işlemi de uygulanabilmektedir. Böylece öznitelik uzay boyutu düşeceğinden zaman ve performans kazancı sağlanabilmektedir. Sistemde boyut indirgeme işlemi sadece eğitim verisi üzerinde uygulanıyormuş gibi gösterilse de eğitim ve test setlerinin aynı veri kümesinden elde edildiği sistemlerde bu işlem test verisi üzerinde de uygulanmaktadır. Ancak sistemimize ait akış diyagramı eğitim ve test verilerinin aynı veri kümesinden elde edilmediği varsayılarak gösterilmiştir.

Sistemde önişleme aşamasındaki bazı işlemler (kök bulma, durak kelimeleri çıkarma vb.) geliştirilen prototip üzerinde istendiği takdirde devre dışı bırakılabilmektedir. Ağırlıklandırma, modele uygun öznitelik çıkarma, boyut indirgeme ve sınıflandırma işlemleri ise sistemde bulunan farklı yöntemler ile gerçekleştirilebilmektedir. Bu bölümde alt başlıklar halinde literatürde kullanılan ve Bölüm 2'de açıklanan yöntemlerin bu tez kapsamında geliştirdiğimiz OMESIS sisteminde uygulanma adımları açıklanmıştır. Aşamaların her bir deneye özgü detayları Bölüm 4'te deneysel sonuçlar ile birlikte verilmiştir.

3.2.1. Önişleme

3.2.1.a. Dizge parçalama

Önişleme aşamasının ilk adımı olan dizge parçalama (dizgeciklere ayırma) adımında sınıflandırmanın türüne göre (MS, DA vb. gibi) metinsel içerik üzerinde yapılan işlemler değişiklik gösterebilmektedir. Bu aşamada anlamlı özniteliklerin elde edilebilmesi için içerik temizlenir. Nasıl bir temizleme (cleaning) işleminin uygulanacağı; üzerinde çalışılan metin tabanlı belgelerin türüne göre değişiklik gösterir. Eğer işlenen veri web sayfalarından oluşuyorsa html elementlerinin, köşe yazılarından oluşuyorsa özel karakterlerin, sosyal medya mesajlarından oluşuyorsa bazı özel terimlerin içerikten ayrıştırılması gerekir. Bu bağlamda sistemimizin bu aşamasında parçalanan doküman veya mesaj içeriği anlamsız her türlü karakterden (rakam, noktalama işaretleri vb.) temizlenerek anlamlı terimler ayrıştırılmıştır.

3.2.1.b. Durak kelimeleri çıkarma

Durak kelimeler bir dilde yaygın olarak kullanılan ve genellikle tek başına kullanıldığında bir anlam ifade etmeyen kelimelerdir. Bu nedenle MS ve DA çalışmalarında genellikle durak kelimeleri çıkarma işlemi uygulanmaktadır. Ancak bu kelimeler çıkarılırken kullanılması gereken bir kelime listesine ihtiyaç duyulmaktadır ve her dilin durak kelimeleri olmakla beraber standart bir liste mevcut değildir. Sistemimizin bu adımında bu amaçla İngilizce ve Türkçe için Lucene API (Anonymous 2011b)'de ön tanımlı olarak bulunan durak kelime listeleri kullanılmıştır. Önişleme sırasında işlenen veriden bu listelerde bulunan terimler çıkarılmıştır. Durak kelimelerin çıkarılması işleminin, yapılan sınıflandırma türüne ve öznitelik çıkarım modeline bağlı olarak uygulanabilirliği değişiklik göstermektedir. Çalışmamızda her iki dil için kullanılan durak kelimeler Şekil 3.3'te verilen kelime bulutu ile gösterilmiştir.

Şekil 3.3. Türkçe (a) ve İngilizce (b) durak kelimeler

Bu aşamada, İngilizce için olumsuzlama işlemi uygulanmadığında kısaltma terimleri (haven't, isn't, he's, you're vb. gibi) de sistemimiz tarafından durak kelime olarak kabul edilir. Ancak DA çalışmalarında bu tür terimlerin elenmesi içerikteki olumsuzluk duygusunun kaybedilmesine neden olabileceği için genellikle olumsuzlama işlemi uygulanır ve bu tür kelimelerin korunması sağlanır.

3.2.1.c. Tekrarlanan harflerin çıkarılması

Sistemimizde uygulanan tekrarlanan harflerin çıkarılması (text normalization, removing repetitive letters) işlemi DA ve MS amaçlı çalışmalarda kullanılmaktadır. Ancak çoğunlukla sosyal medya ve web ortamından alınan içerikler yazım yanlışları içermekte ve dil kurallarına uyulmamaktadır. Bu durumun bir sonucu olarak sosyal medya kullanıcılarının herhangi bir duyguyu daha güçlü bir şekilde ifade etmek amacıyla kelimelerin bazı harflerini tekrarlayarak yazması çok sık karşılaşılan bir durumdur. Bu nedenle özelikle DA çalışmalarında öznitelik uzay boyutunun artmasını engellemek amacıyla bu tür kelimeler üzerinde tekrarlanan harflerin çıkarılması işlemi uygulanmaktadır. Sistemimizde referans alınan çalışmalara paralel olarak İngilizce için terim içerisindeki tekrar eden harf sayısı ikiye (Go *et al.* 2009b), Türkçe için ise bire **(**Türkmenoğlu and Tantuğ 2014**)** düşürülmektedir.

Çizelge 3.4. Tekrarlanan harflerin çıkarılması

Türkçe		**İngilizce**	
günaydıuuuuın	*günaydın*	*cooooooool*	*cool*
geceleeeeeerrrrrrr	*geceler*	*gooood*	*good*

Örnek bir tekrarlanan harflerin çıkarılması işlemi Çizelge 3.4'te verilmiştir.

3.2.1.d. Olumsuzlama

Olumsuzlama (negation) işlemi genellikle DA çalışmalarında başvurulan bir yöntemdir. Bir metin veya mesaj içerisindeki olumsuz terim veya terim gruplarının tespit edilmesi özellikle ifade edilen olumsuzluk duygusunun tespit edilmesinde çok önemlidir. Sistemimizin bu adımında olumsuzlama işlemi yerel kapsamda (phrase level) uygulanmıştır. İngilizce için Çizelge 3.5'te verilen terimler olumsuzlama işleminden geçirilmiştir (Dadvar *et al.* 2011).

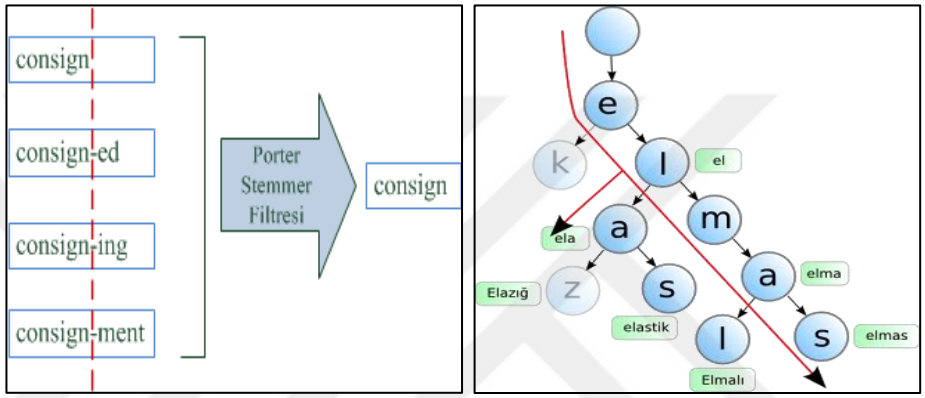
Türkçe için ise yine yerel kapsamda Türkmenoğlu ve Tantuğ (2014)'un çalışmasında kullandığı yönteme ek olarak olumsuzluk eklerini (me, ma, mez, maz, meden, madan, sız, suz, sızın, suzun) içeren terimler olumsuzlama için kullanılmıştır. Olumsuzluk ekinin gözlendiği terimlerde bu ek çıkarılıp terim köküne indirgenmiş ve yanına "değil" terimi eklenmiştir. Şekil 3.3'te görüldüğü gibi "değil" terimi Türkçe durak kelime listesinde bulunmaktadır. Bu nedenle sistemimizde olumsuzlama işlemi, durak kelime çıkarma işleminden sonra gerçekleştirilmekte ve "değil" teriminin öznitelik grubundan çıkarılması engellenmektedir.

Çizelge 3.5. İngilizce için kullanılan olumsuzlama terimleri

Terim	**Olumsuzlama**	**Terim**	**Olumsuzlama**
couldn't	could not	shouldn't	should not
wasn't	was not	weren't	were not
didn't	did not	don't	do not
wouldn't	would not	doesn't	does not
haven't	have not	won't	will not
hasn't	has not	hadn't	had not

3.2.1.e. Gövdeleme

Sistemimizde kök bulma (gövdeleme) işlemi, kelimeleri köküne indirgeyerek öznitelik uzay boyutunun artmasını engelleme amaçlı kullanılmıştır. Türkçe için bu işlem Türkçe bir DDİ kütüphanesi olan Zemberek ile gerçekleştirilmiştir (Anonymous 2007; Akın and Akın 2007).



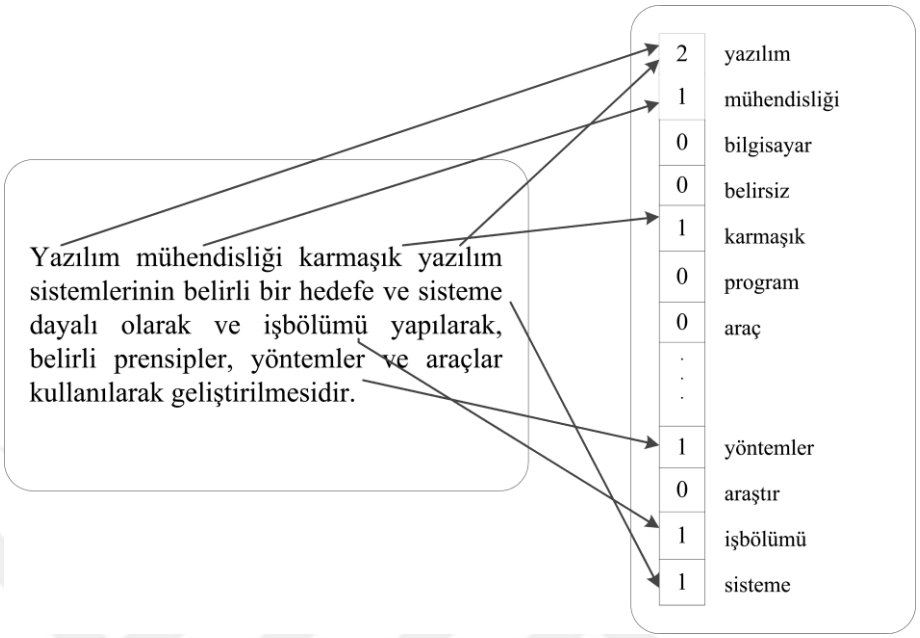
(a) (b)

Şekil 3.4. Gövdeleme; (a) Porter Stemmer yöntemi (b) Zemberek kök aday bulucu

İngilizce için ise literatürde yaygın olarak kullanılan Porter Stemmer algoritması ile kelimeler köküne indirgenmiştir (Porter 1980). Gövdeleme işleminin kelimeler üzerinde Porter Stemmer algoritması ve Zemberek kök aday bulucu ile uygulanışı Şekil 3.4'te gösterilmiştir.

3.2.1.f. Modele uygun öznitelik çıkarma

MS ve DA çalışmalarında öznitelikler, genellikle kelimeler veya içerikten çıkarılan farklı boyutlardaki karakter katarları olmaktadır. Bunun yanı sıra metne özgü çeşitli yapısal ve istatistiksel özellikler de SSTF model ile öznitelik olarak kullanılmaktadır. Bu nedenle, sistemimizin bu aşamasında BoW, Ngram ve SSTF olmak üzere üç farklı model kullanılarak öznitelik çıkarılmıştır. BoW modelde öznitelikler metin içerisinden çıkarılan anlamlı kelimelerdir (Şekil 3.5).



Şekil 3.5. BoW model

Ngram modelde ise öznitelikler metin içerisinden çıkarılan farklı uzunluklardaki karakter katarları (karakter seviye) veya kelime kombinasyonlarından (kelime seviye) oluşmaktadır. Sistemimizde hem karakter, hem de kelime seviyesinde Ngram öznitelikleri çıkarmak mümkündür. Karakter seviyede Ngram'lar karakter katarının uzunluğu ile isimlendirilirken, kelime seviye Ngram'lar içerdiği kelime sayısı ile isimlendirilir. Karakter seviyede katar uzunluğu iki ise bigram, üç ise trigram özel ismi verilir ve üçten fazla uzunluktaki katarlar uzunluğu (4-fourgram, 5-fivegram vb.) ile isimlendirilir. Kelime seviyede ise sadece tek bir kelime ile oluşturulan Ngram'lar için unigram özel ismi verilir. Birden fazla kelime içeren Ngram'larda ise isimlendirme karakter seviye ile aynıdır. Ngram model ile öznitelik çıkarılmak istenen metin "Java nesne yönelimli dildir" olsun. Bu durumda karakter seviye Ngram'lar aşağıdaki gibi elde edilmektedir (Kanaris *et al.* 2007). Boşluk karakteri "_" karakteri ile gösterilmiştir.

- Bigram: "Ja", "av", "va", "a_", "_n", "ne", "es", "sn", "ne", "e_", "_y", "yö", "ön", "ne", "el", "li", "im", "ml", "li", "i_", "_d", "di", "il", "ld", "di", "ir"

- Trigram: “Jav”, “ava”, “va_”, “a_n”, “_ne”, “nes”, “esn”, “sne”, “ne_”, “e_y”, “yön”, “öne”, “nel”, “eli”, “lim”, “iml”, “mli”, “li_”, “i_d”, “dil”, “ild”, “ldi”, “dir”

Kelime seviye Ngram’lar ise aşağıdaki gibi elde edilmektedir:

- Unigram: “Java”, “nesne”, “yönelimli”, “dildir”
- Bigram: “Java_nesne”, “nesne_yönelimli”, “yönelimli_dildir”
- Trigram: “Java_nesne_yönelimli”, “nesne_yönelimli_dildir”

Yukarıda açıklanan Ngram modelden anlaşılacağı üzere aslında kelime seviye Ngram model, kelime torbası model ile yöntem olarak aynıdır (her iki modelde de öznitelikler kelimeler olur) ancak önişleme aşamasında Ngram modelde ayrıştırma dışında bir işlem uygulanmaz. Ayrıca karakter seviye Ngram modelde en düşük katar uzunluğu ikiden az olamaz (unigram çıkarılmaz). SSTF modelde ise metnin yapısına bağlı olarak değişebilen çeşitli öznitelikler (örneğin köşe yazılarında başlık uzunluğu, paragraf sayısı vb. gibi) ve istatistiksel bilgiler (hece sayısı, cümle başına düşen kelime sayısı vb. gibi) öznitelik olarak kullanılmaktadır. Bu modelin en önemli avantajı BoW ve Ngram model ile kıyaslandığında elde edilen öznitelik sayısı çok daha az olduğu için zaman ve performans kazancı sağlamasıdır.

3.2.1.g. Terim ağırlıklandırma

Sistemimizin bu adımında, Şekil 2.4’te verilen vektör uzay modeli ile temsil edilen özniteliklerin ağırlıklandırılması amacıyla aşağıda açıklanan ağırlıklandırma yöntemleri kullanılmıştır. TA aşamasında farklı ağırlıklandırma yöntemlerini kullanmak ve sınıflandırma sonuçları üzerindeki etkisini incelemek amacıyla aşağıdaki yöntemler uygulanmaktadır. Sistemimizde vektör ile temsil edilen örneklere ait öznitelikler, bu yöntemlerden isteğe bağlı olarak seçilen bir yöntem ile ağırlıklandırılabilmektedir. W ağırlıklandırma fonksiyonunu, c bir terimi (BoW, Ngram veya SSTF model ile çıkarılmış bir öznitelik), d bir doküman veya mesajı, N toplam doküman sayısını ve $|D|$

tüm dokümanları temsil etmek üzere; TF, c teriminin bir d dokümanında gözlenme sıklığını (frekansını) ifade eder.

$$W_{TF} = TF(c, d) \tag{3.1}$$

BINARY yöntemi, öznitelikleri ağırlıklandırırken frekanslarının etkisini göz ardı eden bir yöntemdir. Bu yöntemde önemli olan, bir dokümanın bir terimi içerip içermediğidir. Eğer içeriyorsa hangi sıklıkta içerdiğine bakılmaz ve ağırlığı 1 kabul edilir.

$$W_{Boolean}(c, d) = \begin{cases} 1, & TF \geq 1 \\ 0, & diğer \end{cases} \tag{3.2}$$

IDF yöntemi, özniteliklerin ağırlıklarını yerel (bulunduğu doküman) değil global (tüm veri setinde) kapsamda değerlendirmeye tabi tutar. Bu yöntemle veri setinde yaygın olarak kullanılan terimlerin ağırlığı düşürülürken, çok nadir olarak gözlemlenen terimlerin ağırlığı yükseltilmeye çalışılır. DF, bir c teriminin veri setinde gözlendiği doküman sayısı olmak üzere (eşitlik 3.3), IDF 3.4 eşitliği ile formüle edilir.

$$W_{DF} = \sum_{i=1}^{n} TF(d_i, c) \tag{3.3}$$

$$W_{IDF}(c, d) = log \frac{N}{DF(c, d)} \tag{3.4}$$

TF-IDF, bir terimin ağırlığını bir dokümandaki frekansıyla orantılı olarak artırır ancak veri seti genelindeki frekansıyla bunu dengelemeye çalışır. IDF yönteminde olduğu gibi amacı dokümanlarda ortak olarak kullanılan terimlerin ağırlığını düşürmektir ancak bunu yaparken terimin yerel frekansını da dikkate alır.

$$W_{TF-IDF}(c, d) = TF(c, d).IDF(c, d) \tag{3.5}$$

Entropi ağırlıklandırmasında, eğer bir terim tüm dokümanlarda bir kez gözleniyorsa ağırlığı sıfır, sadece bir dokümanda bir kez gözleniyorsa bir olacaktır. Entropi yöntemi

de nadir olarak gözlenen terimlerin ağırlığını yükselttiği için kullanışlı bir yöntemdir (Chisholm and Kolda 1999). Bu yöntemde terim ağırlığı 0 ile 1 arasında bir değer almaktadır.

$$W_{Entropy}(c,d) = 1 + \frac{\sum_{d \in D} P_{d,c} log_2 P_{d,c}}{log_2 |D|} \; ve \; P_{d,c} = \frac{TF_{(d,c)}}{DF_{(c,d)}} \tag{3.6}$$

LTF, normal terim frekansının logaritmadan geçirilerek elde edilen bir varyasyonudur.

$$W_{LTF}(c,d) = log_2(1 + TF(c,d)) \tag{3.7}$$

Ölçeklenmiş terim frekansı (STF: Scaled Term Frequency) da yine terim frekansının logaritması alınarak elde edilen ölçeklenmiş bir ağırlıklandırma fonksiyonudur.

$$W_{STF}(c,d) = \begin{cases} 1 + log_2 TF(c,d), & TF(c,d) > 0 \\ 0, & diğer \end{cases} \tag{3.8}$$

Genişletilmiş normalize terim frekansı (ANTF: Augmented Normalized Term Frequency), terim frekansının nozmalize edilmiş bir versiyonudur. Özellikle içerik bakımından çok uzun olan metin veya mesajlardan oluşan veri kümelerinde herhangi bir terimin frekansı çok daha yüksek olmaktadır. ANTF, bu tür veriler üzerinde çalışırken ağırlıklandırma aşamasında frekansın bir nevi normalize edilerek kullanılmasını sağlar.

$$W_{ANTF}(c,d) = \frac{1 + TF(c,d)/max_c TF(c,d)}{2} \tag{3.9}$$

Logaritmik entropi (LE: Logarithmic Entropy) yöntemi, entropi yönteminin logaritma fonksiyonundan geçirilmiş versiyonudur.

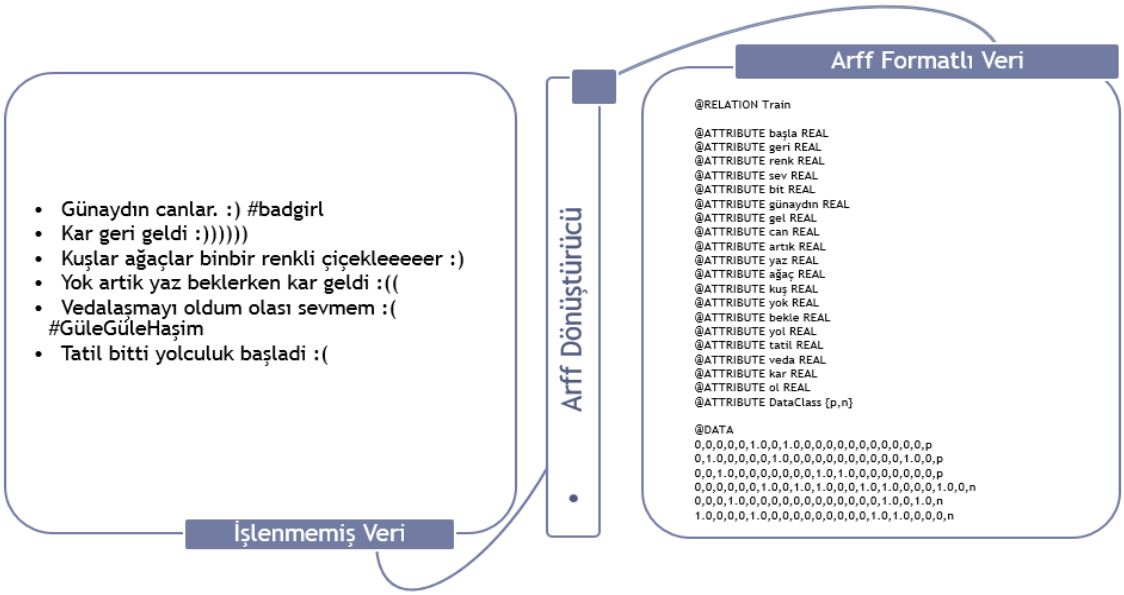
$$W_{LE}(c,d) = Entropy(c,d).LTF(c,d) \tag{3.10}$$

GF-IDF yöntemi ise terimin veri seti genelinde gözlenme frekansını kullanarak ağırlık atayan bir yöntemdir.

$$W_{GF-IDF}(c,d) = \frac{\sum_d TF(c,d)}{\sum_d Boolean(c,d)} \tag{3.11}$$

3.2.1.h. ARFF formatına dönüştürme

Sistemimizin bu adımında, işlenmiş ham verinin sınıflandırma işleminin uygulanabileceği uygun formata dönüştürülmesi gerekir.



Şekil 3.6. ARFF dönüşümü

Geliştirdiğimiz OMESIS yazılımı WEKA (Anonymous 2009; Hall *et al.* 2009) tabanlı olduğu için bu aşamada önişlenmiş ham veri üzerinde Şekil 3.6'da gösterildiği gibi ARFF dosya formatına dönüştürme işlemi uygulanmaktadır. ARFF formatlı dosya içeriği (önişlenmiş) kullanılan öznitelik çıkarım modeli ve ağırlıklandırma yöntemine göre değişiklik göstermektedir. Sistemimizde sınıflandırma testi yapılabilmesi için verinin bu aşamadan geçirilerek dönüştürme işlemine tabi tutulması gerekmektedir.

3.2.2. Boyut indirgeme

Sistemimizde boyut indirgeme aşamasında öznitelik seçme yöntemlerinden Ki-Kare, Karşılıklı Bilgi, Bilgi Kazanımı, Korelasyon Tabanlı Öznitelik Seçme ve öznitelik çıkarma yöntemlerinden ise Saklı Anlam Analizi yöntemi kullanılmaktadır.

3.2.2.a. Ki-kare

Ki-Kare yöntemi, öznitelik gözlemlenebilirliğinin sınıftan bağımsız olduğu varsayımı ile beklenen dağılımdan sapmayı ölçen ve yaygın olarak kullanılan istatistiksel bir test yöntemidir (Forman 2003). Yani istatistikte ki-kare, iki olay arasındaki bağımsızlığı ölçmek için kullanılır. Bu iki olayın birbirinden bağımsız olması ise aşağıdaki koşul gereği varsayılır (eşitlik 3.12).

$$P(XY) = P(X).P(Y) \tag{3.12}$$

Metin öznitelikleri seçilirken, bu iki olay sırasıyla belirli bir terim ve sınıfın gözlemlenebilirliği olaylarına karşılık gelir (Uysal and Gunal 2012). Veri setindeki her bir benzersiz öznitelik için mevcut sınıflar ile arasındaki bağımsızlık ilişkisini çıkarmak amacıyla ki-kare skoru hesaplanır. Öznitelikler seçilirken, bir t teriminin bir c sınıfıyla arasındaki ki-kare skoru ise aşağıdaki formülle hesaplanır (Yang and Pedersen 1997).

$$X^2(t,c) = \frac{N.(AD - CB)^2}{(A + C).(B + D).(A + B).(C + D)} \tag{3.13}$$

Yukarıdaki eşitlikte N: toplam doküman sayısını, A: t terimini içeren ve c sınıfında olan doküman sayısını, B: t terimini içeren ancak c sınıfında olmayan doküman sayısını, C: t terimini içermeyen ancak c sınıfında olan doküman sayısını ve D: t terimini içermeyen ve c sınıfında olmayan doküman sayısını temsil eder. Bir terimin veri seti için iyiliğini tahmin etmek amacıyla yaygın olarak maksimum (max) ve ortalama (mean) fonksiyonları kullanılır.

$$X^2(f) = argmax_{c_i} X^2(f, c_i) \tag{3.14}$$

$$X^2(f) = \sum_i P(c_i) . X^2(f, c_i) \tag{3.15}$$

Bir terimin her bir kategori ile arasında hesaplanan kategori bazlı (category specific) ki-kare skorları yukarıdaki iki formülle birleştirilir (genelleştirilir). Ki-kare skoru terim ile sınıf arasında bir ilişki yoksa yani birbirinden bağımsız ise doğal olarak sıfır değerini alır (Yang and Pedersen 1997). Terim ile sınıf arasındaki ki-kare skoru arttıkça aralarındaki ilişki de artmaktadır. Bu nedenle bu yöntemle ki-kare skoru yüksek olan terimler (öznitelikler) seçilir (Galavotti *et al.* 2000).

3.2.2.b. Karşılıklı bilgi

Karşılıklı Bilgi birçok yöntemin altyapısını oluşturan entropi kavramı kullanılarak istatistikte iki değişken veya olay arasındaki bağımlılığı ölçme, metin madenciliği alanında ise öznitelik seçme amacıyla kullanılmaktadır. Uygulandığı alanda değişkenlerin türüne (ayrık, sürekli) göre çeşitli varyasyonları (Pointwise, Information Theoretic vb.) geliştirilmiştir. Ancak metin madenciliği alanında karşılıklı bilgi yönteminin Information Theoretic MI olarak isimlendirilen versiyonu kullanılmaktadır. Çünkü terim ve sınıf burada ayrık iki değişkeni temsil etmektedir (Xu *et al.* 2007). X, ayrık değerler alabilen rastgele bir değişken ve H entropi olmak üzere bu değişkenin belirsizliği (uncertainity) aşağıdaki gibi tanımlanır.

$$H(X) = -\sum_{x \in X} p(x) log p(x) \tag{3.16}$$

$p(x)$, X değişkeninin olasılık yoğunluk fonksiyonudur. Benzer şekilde X ve Y değişkenlerine ait ortak entropi ise 3.17 eşitliği ile tanımlanır.

$$H(X, Y) = -\sum_{y \in Y} \sum_{x \in X} p(x, y) log p(x, y) \tag{3.17}$$

Koşullu entropi, değişkenlerden birisi bilindiğinde diğer değişkenin değerindeki belirsizliğin azaltılması anlamına gelir. Bu bağlamda Y değişkeni bilindiğinde X ile Y arasındaki koşullu entropi aşağıdaki gibi olacaktır.

$$H(X|Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) log p(x|y) \tag{3.18}$$

Burada $p(x,y)$, y verildiğinde x değişkeninin tahmin edilme olasılığını (posterior probability) gösterir. Buna göre eğer x tamamen y değişkenine bağımlı ise $H(X|Y) = 0$ olur. İki değişken arasında ne kadar bilgi paylaşıldığını ölçmek için ise karşılıklı bilgi yöntemi aşağıdaki gibi tanımlanır.

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y)\, log \frac{p(x,y)}{p(x).p(y)} \tag{3.19}$$

Bu eşitliğe göre, karşılıklı bilgi x ve y değişkenleri birbirleriyle ilişkili ise yüksek olacaktır. Aralarında bir ilişki olmaması durumunda ise bu değer sıfır olacaktır (Liu *et al.* 2009). Yukarıda verilen karşılıklı bilgi yöntemi metin madenciliği alanında uyarlandığında ise aşağıdaki gibi tekrar formüle edilmektedir (Manning *et al.* 2008).

$$MI = \frac{N_{11}}{N} log_2 \frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N} log_2 \frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N} log_2 \frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N} log_2 \frac{NN_{00}}{N_{0.}N_{.0}} \tag{3.20}$$

Yukarıdaki eşitlikte verilen N_{00}, N_{01}, N_{10} ve N_{11} sırasıyla c sınıfında olmayan ve t terimini içermeyen, c sınıfında olan ancak t terimini içermeyen, t terimini içeren ancak c sınıfında olmayan, t terimini içeren ve c sınıfında olan doküman sayısıdır. $N_{0.} = N_{01} + N_{00}$ ve $N_{1.} = N_{10} + N_{11}$ ifadeleri sırasıyla t terimini içeren ve içermeyen doküman sayısını temsil eder. $N_{.1} = N_{01} + N_{11}$ ve $N_{.0} = N_{10} + N_{00}$ sırasıyla kategorisi c olan ve olmayan doküman sayısını temsil eder. $N = N_{00} + N_{01} + N_{10} + N_{11}$ ise toplam doküman sayısıdır.

3.2.2.c. Bilgi kazanımı

Bilgi Kazanımı MÖ alanında bir özniteliğin iyiliğinin ölçülmesi amacıyla kullanılan yöntemlerden birisidir. Bilgi kazanımı yöntemi veri setinde bulunan öznitelikleri, iyilik derecesine göre sıralar ve derecesi önceden belirlenen bir eşik değerin altında olan öznitelikler veri setinden çıkarılır. Bilgi kazanımı yöntemi bir terimin bir dokümanda bulunup bulunmaması bilgisine dayanarak kategori tahmini için kazanılan bilgi miktarını ölçer (Yang and Pedersen 1997). Bir t terimi için m kategorili bir veri setindeki bilgi kazanımı aşağıdaki gibi tanımlanır.

$$IG(t_i) = -\sum_{k=1}^{m} p(C_k) log p(C_k) + p(t_i) \sum_{k=1}^{m} p(C_k|t_i) log p(C_k|t_i) + p(\bar{t}_i) \sum_{k=1}^{m} p(C_k|\bar{t}_i) log p(C_k|\bar{t}_i) \quad (3.21)$$

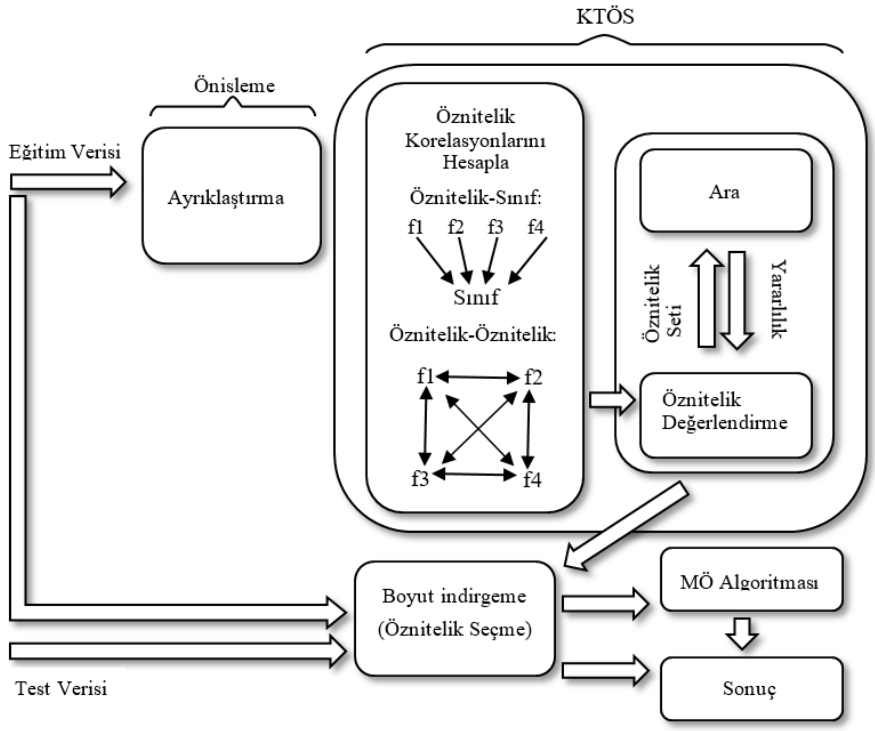
Yukarıdaki eşitlikte $p(C_k) = (a+c)/N$, $p(t_i) = (a+b)/N$, $P(\bar{t}_i) = (c+d)/N$, $p(C_k|t_i) = a/(a+b)$ ve $p(C_k|\bar{t}_i) = c/(c+d)$ olur. Burada $a, b, c\ ve\ d$ sırasıyla t_i terimini içeren ve C_k kategorisinde olan, t_i terimini içeren ancak C_k kategorisinde olmayan, t_i terimini içermeyen ancak C_k kategoroisinde olan, t_i terimini içermeyen ve C_k kategorisinde olmayan doküman sayısını temsil eder. $N = a + b + c + d$ ise veri setindeki toplam doküman sayısıdır.

3.2.2.d. Korelasyon tabanlı öznitelik seçme

Korelasyon tabanlı öznitelik seçme (CFS: Correlation-based Feature Selection) yöntemi M. Hall (1999) tarafından MÖ çalışmaları için önerilmiştir. Bu yöntem öznitelik alt kümelerini sezgisel bir fonksiyona dayalı ilişkiye göre derecelendiren bir filtredir. Bu sezgisel fonksiyon sınıflar ile yüksek derecede korelasyon olan ancak birbirleri arasında korelasyon bulunmayan öznitelikler içeren alt kümelerle ilgilenir. Bir sınıf ile arasında düşük korelasyon bulunan öznitelikler gereksiz (ilgisiz) kabul edilir ve göz ardı edilir. Benzer şekilde kendisi dışındaki diğer özniteliklerden bir ya da daha fazla öznitelikle arasında yüksek derecede korelasyon bulunan öznitelikler de gereksiz kabul edilir ve dikkate alınmaz. KTÖS yönteminde bir özniteliğin kabul edilebilirliği örnek uzayındaki kategorilerin tahmin edilebilmesine yaptığı katkıya bağlıdır. Yöntem aşağıdaki eşitlikle formüle edilir.

$$M_s = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}} \tag{3.22}$$

Burada M_s, k tane öznitelik içeren S alt kümesinin sezgisel değerini (yararlılığını), $\bar{r}_{cf}$ ortalama öznitelik-sınıf (öznitelik ile sınıf) arası korelasyonu ($f \in S$), ve $\bar{r}_{ff}$ ortalama öznitelik-öznitelik arası kolerasyonu temsil eder. Eşitliğin pay kısmında bulunan ifade öznitelik setinin sınıfın tahmin edilmesinde ne kadar etkili olduğunu gösterirken paydada bulunan ifade öznitelikler arasında ne kadar ilgisizlik olduğunu gösterir. Şekil 3.7 KTÖS algoritmasının aşamalarını ve MÖ yöntemleriyle birlikte nasıl kullanıldığını göstermektedir (Hall 1999).



Şekil 3.7. KTÖS yönteminin bileşenleri ve makine öğrenmesi ile birlikte kullanımı

Şekilden de anlaşılacağı üzere eğitim ve test verileri sadece KTÖS tarafından seçilen öznitelikleri içerecek şekilde indirgenir. Öznitelik seçme işleminden sonra boyutu indirgenen veri, MÖ yöntemlerinde kullanılabilir.

3.2.2.e. Saklı anlam analizi

Saklı anlam indeksi (LSI: Latent Semantic Indexing) olarak da bilinen saklı anlam analizi (Deerwester *et al.* 1990) dokümanlar temsil edilirken eş, yakın ve çok anlamlı kelimelerin kullanımından kaynaklanan boyut problemini çözmek için geliştirilmiş bir boyut indirgeme yöntemidir (Sebastiani 2002). Yöntemin önerilmesindeki temel amaç büyük boyutlu doküman veya metin veritabanlarından, gönderilen sorgu metni ile eşleşen ilgili dokümanları elde edebilmektir. Bu amaca ilişkin ilk yaklaşımlar, anahtar kelime eşleme ve kelimelerin dokümanlarda gözlemlenmesine dayalı vektör tabanlı temsili gibi işlemleri içermiştir. Ancak daha sonra saklı anlam analizi bu vektör tabanlı yaklaşımı tekil değer ayrıştırması (SVD: Singular Value Decomposition) yöntemini

kullanarak genişletmiştir (Wiemer-Hastings 2004). TDA (Tekil Değer Ayrıştırması) yönteminin temel amacı yüksek boyutlu bir veriyi daha temiz ve küçük boyutlu alt bir yapıya dönüştürebilmektir. Bu yönüyle TDA öznitelik seçimi veya boyut indirgeme işlemlerinin temelini oluşturur. DDİ uygulamalarında TDA belirli bir eşik değerin altındaki değişimleri (varyasyon) ihmal ederek verinin boyutunu indirgemek için kullanılır ancak bunu yaparken veri setindeki ana ilişkilerin korunmasını da sağlar (Baker 2005). TDA yöntemi lineer cebir tabanlı bir teoreme dayanmaktadır. Bu teoreme göre mxn boyutlu bir C matrisi aşağıdaki eşitlikte verildiği gibi üç farklı matrise ayrıştırılabilir (Manning *et al.* 2008).

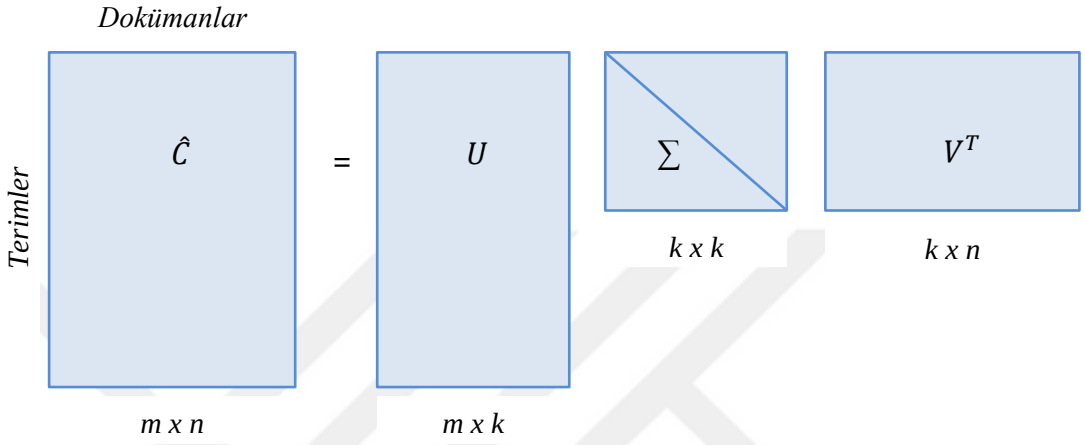
$$C = U \sum V^T \tag{3.23}$$

Yukarıdaki eşitlikte mxm boyutlu U matrisinin sütunları CC^T matrisinin ortogonâl (birbirine dik) öz vektörlerinden (eigenvectors) oluşmaktadır. nxn boyutlu V matrisinin sütünları ise C^TC matrisinin ortogonal öz vektörlerinden oluşmaktadır. C^T ifadesi C matrisinin transpozunu ifade eder. Bir C matrisinin öz değer ve öz vektörleri ise aşağıdaki eşitlik ile elde edilir.

$$C\vec{v} = \lambda\vec{v} \tag{3.24}$$

Yukarıdaki eşitlikte λ, C matrisinin öz değerini ve $\vec{v}$ ise öz vektörünü temsil eder (Baker 2005). Eşitlik 3.23'te verilen U ve V matrislerinin öz değerleri (eigenvalues) birbirinin aynısıdır ve $\sum$ vektörü bu öz değerlerin karekökünü içeren bir köşegen matrisidir. MS çalışmalarında, U matrisi terimler matrisidir ve her bir i. satır ile j. sütundaki değer terimlerin dokümanlarda birlikte gözlenme ölçüsünü temsil eder. Genelde r, C matrisinin rankı olmak üzere $\sum$ vektörünün rxr boyutlu kısmı alınır ve bunun dışında kalan öz değerler sıfır kabul edilir. MS uygulamalarında ise önceden belirlenen bir k değerine köre $\sum$ matrisinin kxk boyutlu kısmı alınır. U matrisinin sağından, $\sum$ matrisinden alınan satırlara karşılık gelen $M - r$ adet sütun alınır diğer sütunlar alınmaz. Benzer şekilde V matrisinin en sağından $N - r$ adet sütunu alınır diğer sütunları dikkate alınmaz (Manning *et al.* 2008). Böylece U, $\sum$ ve S matrislerinin

boyutları indirgendiği için C matrisinin boyutları da indirgenmiş yani veri setindeki ana ilişkiler korunarak en iyi terimler çıkarılmış olur. Saklı anlam analizinde TDA kullanılarak bir terim-doküman matrisinin boyutunun indirgenmesi Şekil 3.8'de gösterilmiştir (Deerwester *et al.* 1990).



Şekil 3.8. Terim-doküman matrisinin TDA kullanılarak indirgenmesi

Yukarıdaki şekilde verilen boyut indirgeme işleminden sonra orijinal terim-doküman matrisi, en büyük k adet öz değer ve bu öz değerlere karşılık gelen öz vektörler kullanılarak yaklaşık olarak elde edilebilmektedir. Böylece saklı anlam analizi ile veri setini temsil edici özelliği yüksek öznitelikler seçilmekte ve aynı zamanda verinin boyutu düşürülmektedir.

3.2.3. Sınıflandırma

Sistemimizde sınıflandırma aşamasında MS ve DA çalışmalarında sıklıkla kullanılan Basit Bayes (NB: Naive Bayes), Multinom Basit Bayes (MNB: Multinomial Naive Bayes), *k*-En Yakın Komşu (*k*-NN: *k*-Nearest Neighbor), Destek Vektör Makineleri (SVM: Support Vector Machines) ve Maksimum Entropi (ME: Maximum Entropy) sınıflandırıcıları kullanılmaktadır. Eğitilen sınıflandırıcı ile oluşturulan modelin değerlendirilmesi, veri seti belirli bir oranda bölünerek veya çapraz geçerleme uygulanarak gerçekleştirilebilmektedir. Sınıflandırıcının performansı ise doğruluk metriği kullanılarak değerlendirmektedir.

3.2.3.a. Basit Bayes

Basit Bayes sınıflandırıcısı, birçok uygulama alanında kullanılan Bayes'in olasılık teorisine dayalı pratik bir yöntemdir. Basit Bayes, örneklerin sahip olduğu özniteliklerin birbirinden (bir kategorideki özniteliklerin bir başka kategorideki özniteliklerden) bağımsız olduğunu varsayar (Kim *et al.* 2006). $C_1, \dots, C_n$ olmak üzere; n tane sınıf ve herhangi bir probleme ait vektör uzay modeli ile temsil edilen birbirinden bağımsız özniteliklere sahip test örneği $X = (x_1, \dots, x_n)$ olsun. Bu test örneğinin Bayes'in olasılık teorisi kullanılarak sınıflandırılması eşitlik 3.25 ile gerçekleştirilir.

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)} \tag{3.25}$$

Bayes sınıflandırıcı, pratikte sınıflandırma yaparken yukarıdaki eşitlikte sadece $P(X|C_k)P(C_k)$ ile ilgilenir. Çünkü $P(X)$ sınıf ve özniteliklere bağlı olmadığı (tüm sınıflar için hesaplandığından dolayı) için sabit kabul edilir ve ihmal edilir (Zhang and Li 2007). $P(C_k)$ yani sınıf önceliği ifadesi 3.26 eşitliği ile elde edilir ve C_k kategorisinde bulunan doküman sayısının toplam doküman sayısına oranını temsil eder. Eğer sınıf önceliği bilinmiyor veya veri setinden elde edilemiyorsa, tüm sınıfların önceliği birbirine eşit kabul edilir.

$$P(C_k) = \frac{C_k}{|C|} \tag{3.26}$$

Paydaki eşitlik zincir kuralı ve özniteliklerin bağımsız olduğu varsayımı kullanılarak aşağıdaki gibi yazılır.

$$P(C_k|x_1, \dots, x_n) = P(C_k) \prod_{i=1}^{n} P(x_i|C_k) \tag{3.27}$$

Yukarıdaki eşitlikten de anlaşılacağı üzere sınıflandırma işlemi için en anlamlı ifade $P(x_i|C_k)$ ifadesi olmaktadır. Bu olasılık değeri de veri setinde bulunan örnekler aracılığıyla tahmin edilebilir ve aşağıdaki gibi tanımlanır.

$$P(x_i|C_k) = \frac{C_{ik}}{C_k} \tag{3.28}$$

Burada C_{ik}, x_i niteliğini bulunduran ve kategorisi C_k olan örnek sayısını, C_k ise aynı kategorideki toplam örnek sayısını temsil eder. Bayes sınıflandırıcı, test örneği için 3.29 eşitliği ile mevcut her sınıfa ait bir olasılık hesaplar ve örneği en yüksek değere sahip sınıfa atar.

$$\underset{k\in\{1,\ldots,K\}}{argmax}\, P(C_k) \prod_{i=1}^{n} P(x_i|C_k) \tag{3.29}$$

3.2.3.b. Multinom basit Bayes

Multinom Basit Bayes yönteminde örnekler (öznitelik vektörleri) multinom dağılım gösteren olayların frekansları ile temsil edilir. Bu yüzden özniteliklerin frekansları ile temsil edilebildiği veriler üzerinde iyi çalışmaktadır. Bu bağlamda MS çalışmalarında da öznitelikler frekanslarıyla (veya atanan ağırlıkları ile) temsil edildiği için başarılı bir şekilde uygulanabilmektedir. Uygulanması kolay, hızlı ve etkili bir yöntem olduğu için sınıflandırıcı olarak yaygın bir şekilde kullanılmaktadır (Rennie *et al.* 2003). McCallum and Nigam (1998) doküman sınıflandırma çalışmalarında uygulanabileceğini ve frekans bilgisini kullandığı için Bernoulli modele göre performansı geliştirdiğini göstermişlerdir. MNB algoritması da klasik Basit Bayes sınıflandırıcı gibi 3.25 eşitliğinde verilen Bayes teoremine dayanır. Bu teoremden yola çıkarak bir d test dokümanının bir c sınıfına ait olma olasılığını yine 3.27 eşitliğini kullanarak hesaplamaktadır. Ancak MNB klasik Basit Bayes yönteminden farklı olarak özniteliklerin frekans bilgisiyle ilgilendiği için 3.27 eşitliğinde verilen $P(x_i|C_k)$ olasılığını (multinom parametre) aşağıda verilen eşitlikle tahmin eder.

$$P(x_i|C_k) = \frac{N_{ik}}{N_k} \tag{3.30}$$

Burada N_{ik}, x_i özniteliğini içeren C_k kategorili örneklerde x_i'nin toplam frekansını ve N_k aynı kategorideki örneklerin içerdiği özniteliklerin toplam frekansını temsil eder (Rennie *et al.* 2003). Bu aşamadan sonra Basit Bayes yönteminde olduğu gibi $P(x_i|C_k)$ ve $P(C_k)$ olasılıkları ile eşitlik 3.29 kullanılarak sınıflandırma yapılır. Ancak eğer eğitim setinde x_i özniteliği C_k kategorisinde gözlenmiyorsa $P(x_i|C_k)$ sıfır olacaktır. Bu durumda diğer öznitelikler için böyle bir durum olmasa bile test örneğinin C_k kategorisine ait olma olasılığı 3.29 eşitliğindeki çarpım durumundan dolayı sıfır olacaktır. Bu problemin önüne geçmek için uygulamada yumuşatma (Laplace Smoothing) yöntemi kullanılarak 3.30 eşitliği aşağıdaki gibi tekrar düzenlenir (Kibriya *et al.* 2005).

$$P(x_i|C_k) = \frac{N_{ik} + 1}{N_k + |V|} \tag{3.31}$$

Burada $|V|$ veri setinden çıkarılan benzersiz öznitelik listesinin boyutunu temsil eder.

3.2.3.c. *k*-en yakın komşu

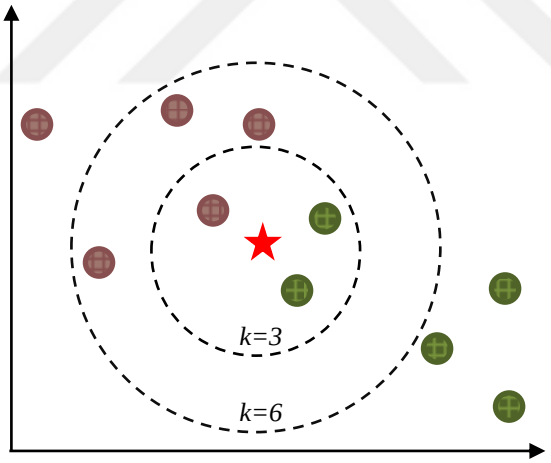
Örnek tabanlı olan k-NN algoritması örnekler arasındaki uzaklık veya benzerliğe dayalı öğrenme yöntemlerinden birisidir. Oldukça etkili ve uygulanması kolay olan bu yöntemde sınıflandırma zamanı uzun sürmekte ve en uygun k değerini tahmin etmek zor olmaktadır (Korde and Mahender 2012). k-NN sınıflandırıcısı, verilen bir test örneği için eğitim setinde bulunan örneklerden uzaklık (veya benzerlik) fonksiyonuna göre en yakın k tane komşuyu bulur ve tahmin edeceği kategori için bu k tane komşunun kategori ağırlıklarını kullanır (Yang and Liu 1999). Test örneğinin kategorisinin belirlenmesi aşamasında her bir komşunun ağırlığı; d_0 test örneğini, $d_j \in KNN(d_0)$ en yakın komşuları ve C kategoriyi temsil etmek üzere aşağıdaki gibi hesaplanır.

$$skor(d_0, C_j) = \sum_{d_j \in KNN(d_0)} Sim(d_0, d_j)(d_j, C_i) \tag{3.32}$$

Eşitlik 3.32'de verilen (d_j, C_i) ifadesi d_j dokümanının kategorisi C_i ise 1, aksi durumda 0 değerini alır. Son olarak sınıflandırıcı 3.33 eşitliği ile ağırlığı en yüksek olan kategoriyi test örneğinin kategorisi olarak belirler (Tan 2006).

$$C = argmax_{c_i}(skor(d_0, C_i)) \tag{3.33}$$

Yöntemde kullanılan uzaklık (distance) ya da benzerlik (similarity) fonksiyonu farklı olabilmektedir. Ancak k-NN sınıflandırıcısında en yakın komşuların belirlenmesinde genellikle örnekler arasındaki Öklid (Euclidean) uzaklığı kullanılmaktadır. k-NN sınıflandırıcısının çalışma prensibi Şekil 3.9'da gösterilmiştir (Baharudin *et al.* 2010).



Şekil 3.9. *k*-En Yakın Komşu sınıflandırması

Sistemimizde uzaklık fonksiyonunun, k-NN sınıflandırıcısı ile yapılan sınıflandırma sonuçlarına etkisini incelemek ve performans karşılaştırmasını yapabilmek amacıyla farklı uzaklık ve benzerlik fonksiyonları uygulanmıştır. t_1 ve t_2 vektör uzay modeli ile temsil edilmiş iki doküman (veya mesaj) vektörü ve $t_1.t_2$ bu iki vektörün nokta (dot product) veya iç çarpımı (inner product) olsun. Bu durumda Öklid fonksiyonu, iki vektör arasındaki ilgili koordinat farkları karesinin karekökünü ölçer ve eşitlik 3.34 eşitliğinde verildiği gibi formüle edilir (Schoenharl and Madey 2008).

$$S_{Euclidean}(t_1, t_2) = [(t_1 - t_2).(t_1 - t_2)]^{1/2} \quad (3.34)$$

Hesaplanabilirlik açısından Öklid uzaklığından daha az maliyet gerektiren Manhattan fonksiyonu aşağıdaki gibi tanımlanır (Wang *et al.* 2007).

$$S_{Manhattan}(t_1, t_2) = |t_1 - t_2| \quad (3.35)$$

Doküman vektörlerinin elemanları arasındaki maksimum uzaklığı döndüren Chebyshev fonksiyonu ise şu şekildedir (Schoenharl and Madey 2008).

$$S_{Chebyshev}(t_1, t_2) = \max(|t_1 - t_2|) = \lim_{k\to\infty}(|t_1 - t_2|^k)^{1/k} \quad (3.36)$$

Kosinüs (cosine) benzerliği iki vektör arasındaki açının kosinüs değerini döndürür. İki vektör arasındaki açı azaldıkça benzerlik artmaktadır (Han *et al.* 2011).

$$S_{Cosine}(t_1, t_2) = \frac{t_1.t_2}{(t_1.t_1)^{1/2}.(t_2.t_2)^{1/2}} \quad (3.37)$$

Kosinüs benzerliği yöntemine benzer bir şekilde hesaplanan ancak doküman vektörlerinin özelleştirilmiş alt uzayını kullanan Pairwise adaptive yöntemi aşağıdaki gibi tanımlanır (D'hondt *et al.* 2010).

$$S_{Pair}(t_1, t_2) = \frac{t_{1,k}.t_{2,k}}{(t_{1,k}.t_{1,k})^{1/2}.(t_{2,k}.t_{2,k})^{1/2}} \quad (3.38)$$

Burada $t_{i,k}$, sırasıyla t_1 ve t_2 vektörlerinin en büyük özniteliklerinin birleşiminden oluşan *k* elemanlı bir alt seti temsil etmektedir. Bir diğer benzerlik fonksiyonu Extended Jaccard (EJ) şu şekilde formüle edilir (Strehl *et al.* 2000).

$$S_{EJ}(t_1, t_2) = \frac{t_1.t_2}{t_1.t_1 + t_2.t_2 - t_1.t_2} \quad (3.39)$$

Extended Jaccard ile benzer bir yöntem olan Dice Coefficient yöntemi şu şekilde tanımlanır.

$$S_{Dice}(t_1, t_2) = \frac{2(t_1.t_2)}{t_1.t_1 + t_2.t_2} \tag{3.40}$$

IT-Sim (Information Theoretic Similarity), doküman benzerliği hesabı için bilgi teorisinden uyarlanan bir yöntemdir ve 3.41 eşitliği ile tanımlanır (Lin 1998; Aslam and Frost 2003).

$$S_{IT}(t_1, t_2) = \frac{2\sum_{w_i} \min(p_{1i}, p_{2i}) log\pi(w_i)}{\sum_{w_i} p_{1i} log\pi(w_i) + \sum_{w_i} p_{2i} log\pi(w_i)} \tag{3.41}$$

Burada w_i: *i.* özniteliği, p_{ji}: *i.* özniteliğin t_j (j=1,2) doküman vektöründeki normalize edilmiş değerini ve $\pi(w_i)$ ise *i.* özniteliğin gözlendiği dokümanların oranını gösterir. Son olarak yukarıda açıklanan yöntemlerin yetersiz kaldığı veya dikkate almadığı durumlar değerlendirilerek önerilen SMTP (Similarity Measure for Text Processing) yöntemi aşağıdaki gibi tanımlanır (Lin *et al.* 2014).

$$N_*(t_{1j}, t_{2j}) = \begin{cases} 0.5\left(1 + exp\left\{-\left(\frac{t_{1j} - t_{2j}}{\sigma_j}\right)^2\right\}\right), t_{1j}.t_{2j} > 0 \\ 0, t_{1j} = 0 \; ve \; t_{2j} = 0 \\ -\lambda, diğer \; durumlar \end{cases} \tag{3.42}$$

$$N_{\cup}(t_{1j}, t_{2j}) = \begin{cases} 0, t_{1j} = 0 \; ve \; t_{2j} = 0 \\ 1, diğer \; durumlar \end{cases} \tag{3.43}$$

Burada σ_j: w_j özniteliğinin eğitim setindeki sıfırdan farklı tüm değerlerinin standart sapmasını ve -λ büyüklüğü ihmal edilen sıfırdan farklı öznitelik değerinin yerine kullanılan negatif sabit bir sayıyı temsil eder. $N_*(t_{1j}, t_{2j})$ ve $N_{\cup}(t_{1j}, t_{2j})$ sırasıyla 3.42 ve 3.43 eşitliklerinde verildiği gibi olmak üzere; $F(t_1, t_2)$ eşitlik 3.44 ile elde edilir.

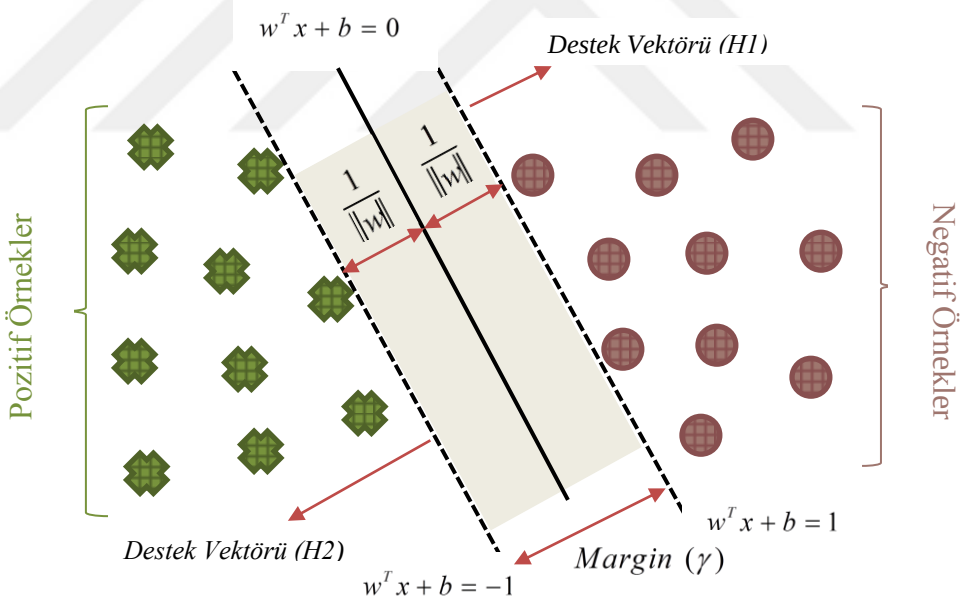
$$F(t_1, t_2) = \frac{\sum_{j=1}^{m} N_*(t_{1j}, t_{2j})}{\sum_{j=1}^{m} N_\cup(t_{1j}, t_{2j})} \tag{3.44}$$

Son olarak t_1 ve t_2 için SMTP benzerliği 3.45 eşitliğinde verildiği gibi formüle edilir.

$$S_{SMTP}(t_1, t_2) = \frac{F(t_1, t_2) + \lambda}{1 + \lambda} \tag{3.45}$$

3.2.3.d. Destek vektör makinesi

Destek Vektör Makinesi algoritması sınıflandırma problemlerinin çözümü için 1995 yılında ortaya atılmış bir MÖ yöntemidir (Cortes and Vapnik 1995).



Şekil 3.10. SVM ile iki sınıflı doğrusal sınıflandırma

Sınıflandırma problemlerinde doğrusal ve doğrusal olmayan olmak üzere iki farklı SVM türü mevcuttur (Burges 1998). SVM temelde iki sınıflı problemleri çözmekle ilgilenir. Burada temel amaç bu iki sınıfı birbirinden ayırt edebilecek ve her iki sınıfa olan uzaklığı maksimum olan bir aşırı düzlem (hyperplane) veya diğer adıyla doğrusal sınıflandırma fonksiyonu elde etmektir. İki sınıflı (negatif, pozitif vb. gibi) bir doğrusal

sınıflandırma problemi üzerinde SVM algoritmasının çalışma prensibi Şekil 3.10'da verilmiştir. Yani pozitif kategorideki örnekler +1, negatif kategorideki örnekler -1 ile temsil edilmiş ise sınıflandırıcı fonksiyon aşağıdaki gibi formüle edilir.

$$f(x) = w^T x - b \tag{3.46}$$

Amaç sınıflandırıcı fonksiyonunu x örneği pozitif ise $sign\big(f(x)\big) = +1$, negatif ise $sign\big(f(x)\big) = -1$ olacak şekilde elde etmektir. Şekil 3.10'dan anlaşılacağı üzere iki sınıfı birbirinden ayıran farklı ve çok sayıda vektör mevcuttur. SVM bu vektörlerden en iyi olanı bulmayı amaçlamaktadır. Sınıflandırıcının genelleme yeteneğinin artması için bu iki sınıf arasındaki mesafenin (margin) maksimum olması gerekmektedir. İki sınıf arasındaki bu mesafe şu şekilde maksimum olarak elde edilir. Pozitif ve negatif kategorilerden birer örneğin ayırıcı düzleme olan uzaklığı sırasıyla $wx^+ + b = +1$ ve $wx^- + b = -1$ olmak üzere margin değeri aşağıdaki gibi olur.

$$\gamma = \frac{w(x^+ - x^-)}{\|w\|} = \frac{2}{\|w\|} \tag{3.47}$$

Eşitlik 3.47'de γ değerinin maksimum olması, $y\{+1, -1\}$ kategorileri temsil etmek üzere;

$$\begin{aligned} &eğer\ y_i = +1\ ise\ w^T x - b \geq 1 \\ &eğer\ y_i = -1\ ise\ w^T x - b \leq -1 \end{aligned} \tag{3.48}$$

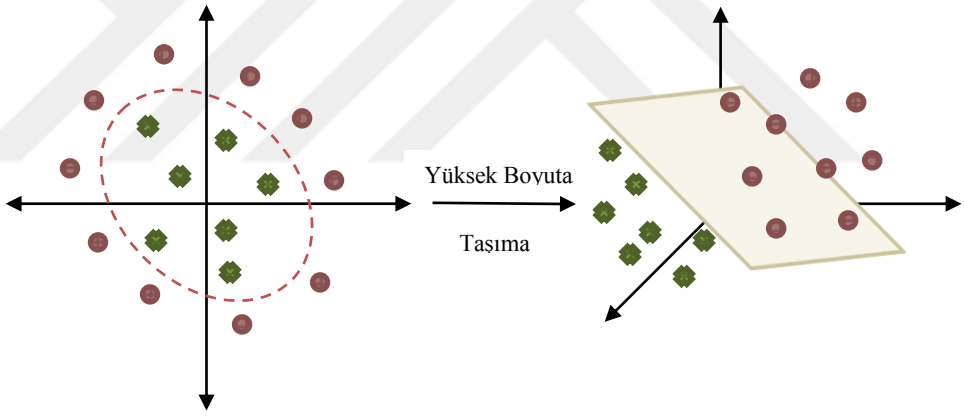
kısıtlarını sağlayacak şekilde $\frac{1}{2}\|w\|^2$ değerinin minimum olarak elde edilmesiyle gerçekleşmektedir. Burada $\|w\|$, ağırlık vektörü olarak adlandırılan w normal düzleminin normudur. Bu işlem optimizasyon problemi olarak ele alınır ve ℓ örnek sayısı olmak üzere w aşağıdaki eşitlik ile elde edilir.

$$y_i(w^T x_i - b) \geq 1, \forall_i = 1, \dots, \ell \tag{3.49}$$

Yukarıda verilen eşitlik ile iki sınıfı birbirinden en iyi şekilde ayırt edebilecek düzlem öğrenilmiş olur. Bu aşamadan sonra gelen her yeni örnek 3.50 eşitliği ile sınıflandırılır. Sonuç sıfırdan büyük ise kategori pozitif, küçük ise negatif olarak belirlenir.

$$f(x) = sign(w^T x_{yeni} + b) \tag{3.50}$$

Doğrusal olmayan SVM ise verinin doğrusal olarak sınıflandırılamadığı durumlarda kullanılır. Bu durumda Şekil 3.11'de gösterildiği gibi veri bir çekirdek fonksiyonu aracılığıyla daha yüksek boyutlu bir uzaya taşınır ve sınıflandırma bu şekilde yapılır (Karagülle 2008). Doğrusal olmayan SVM'de bu amaçla Doğrusal (Linear), Sigmoid, Polinomial, Radyal Tabanlı Fonksiyon (RBF: Radial Basis Function) gibi çekirdekler kullanılmaktadır (Gunn 1998).



Şekil 3.11. Verinin çekirdek fonksiyonu ile yüksek boyuta taşınması

SVM ikili (binary) sınıflandırma problemlerini çözmek geliştirilmiş olsa da çok kategorili sınıflandırma problemlerinde de başarıyla uygulanabilmektedir (Weston and Watkins 1998). Çok kategorili sınıflandırma işleminin uygulanabilmesi için bire-karşı-bir (one-against-one) ve bire-karşı-hepsi (one-against-all) gibi yöntemler önerilmiştir (Fradkin and Muchnik 2006). Bu tez kapsamında SVM uygulanması için LibSVM (Anonymous 2011c; Chang and Lin 2011) kütüphanesi kullanılmıştır. Çok kategorili sınıflandırma problemlerinde LibSVM bire-karşı-bir yöntemini kullanmaktadır (Meyer 2014).

3.2.3.e. Maksimum entropi

Sistemimizde Maksimum Entropi sınıflandırıcısı OpenNLP (Anonymous 2010) kütüphanesinde bulunan MaxEnt paketi ile uygulanmıştır. Lojistik regresyon olarak da bilinen (Faiz and Mercer 2013) Maksimum Entropi yöntemi DDİ uygulamalarında etkili olduğu kanıtlanmış alternatif bir yöntemdir. Bayes sınıflandırıcının aksine özniteliklerin birbirine bağımlı olduğunu varsayar. ME prensibinin temel amacı, veri setine ait tüm kısıtları sağlayacak şekilde örnek veriyi (eğitim verisi) oluşturan ve başlangıçta bilinmeyen en uygun (uniform) modelin oluşturulmasını sağlamaktır (Cuong *et al.* 2006). Bir dokümana ait herhangi bir model ile elde edilmiş öznitelikler $\{f_1, \dots, f_m\}$ ve $n_i(d)$ bu özniteliklerin dokümanda gözlenme frekansları (veya bir ağırlıklandırma yöntemi kullanılarak belirlenmiş ağırlığı) olsun. Bu durumda d dokümanı $\vec{d} \coloneqq (n_1(d), n_2(d), \dots, n_m(d))$ olacak şekilde temsil edilir. Maksimum entropi sınıflandırıcısı bir d test örneğini oluşturduğu $p(c|d)$ istatistiksel model ile c sınıfına atar.

$$p(c|d) = \frac{1}{Z(d)} exp\left(\sum_i \lambda_i F_{i,c}(d,c)\right) \tag{3.51}$$

Burada $Z(d) = \sum_c exp\left(\sum_i \lambda_i F_{i,c}(d,c)\right)$ normalizasyon fonksiyonunu (sabit bir değer) temsil eder. $F_{i,c}(d,c)$, maksimum entropi modelde öznitelik-sınıf fonksiyonu olarak adlandırılır ve 3.52 eşitliği ile tanımlanır.

$$F_{i,c}(d,c') = \begin{cases} 1, n_i(d) > 0 \; ve \; c' = c \\ 0, diğer \; durumlar \end{cases} \tag{3.52}$$

$$c_k = argmaxP(c_i|d) \tag{3.53}$$

Burada d: test örneğinin özniteliklerinin listesini (veya seyrek dizi), c' ise kategorisini temsil eder. Eğer test örneği, özniteliği içeriyor $n_i(d) > 0$ ve kategorisi $c' = c$ (olasılığı hesaplanan kategoride) ise fonksiyon 1 değerini döndürür. Ayrıca λ_i her bir f_i özniteliği için ağırlık değeridir. Bu değerin yüksek olması özniteliğin sınıf için güçlü bir gösterge

olduğunu gösterir (Pang *et al.* 2002). Oluşturulan bu modelde en büyük problem her bir öznitelik için modeldeki kısıtları sağlayabilecek λ_i ağırlık değerlerini tahmin etmektir. Bunun için farklı yöntemler kullanılmakla beraber sistemimizde GIS (Generalized Iterative Scaling) yöntemi kullanılmıştır (Darroch and Ratcliff 1972). Ağırlık değerleri tahmin edildikten sonra artık sınıflandırma için gerekli kısıtları öğrenmiş bir model oluşturulmuş olur. Daha sonra 3.53 eşitliği kullanılarak mevcut her bir sınıf ile test örneği arasındaki olasılık hesaplanır ve en yüksek olasılıklı sınıfa atama yapılır.

3.3. Prototip: OMESIS

Bu bölümde MS ve DA çalışmalarında kullanılan ve bir önceki bölümde anlatılan algoritmaların uygulanması ve sonuçların gösterilmesini içeren OMESIS'e ait grafik arayüzü anlatılmaktadır. Program uygulama sahası olarak metin tabanlı verileri işleme odaklı geliştirilmiş olsa da WEKA alt yapısını kullandığı için ARFF dosya formatındaki herhangi bir veri seti üzerinde sınıflandırma işlemi uygulamak mümkün olmaktadır. Uygulamada veri al, öznitelik seç, biçimlendir, sınıflandır ve analiz olmak üzere 5 temel adım vardır.

Programımız kullanıcı odaklı olarak geliştirilmiş ve işlemlerin arayüz aracılığı ile otomatik olarak gerçekleştirilmesi sağlanmıştır. Veri al aşamasında, sisteme yüklenecek verinin formatı ve bu verinin hangi önişlemlerden geçirileceğiyle ilgili seçeneklerin belirlenmesi gerekmektedir. Bu aşamada program kök dizinine kullanılabilecek hazır veri setleri eklenmiştir. Kullanıcı dilerse bu hazır veri setleri veya sisteme dışardan yükleyebileceği bir veri seti üzerinde MS ve DA tekniklerini kullanarak test yapabilecektir. Programın genel görünümü **Ek 1.1**'de verilmiştir. Uygulama ile gerçekleştirilecek sınıflandırma testlerinde verinin formatından dolayı karşılaşılabilecek zorlukları aşmak ve veri formatından bağımsız bir şekilde otomatik olarak sisteme veri yükleyebilmek için **Ek 1.2**'de verilen seçenek ekranı hazırlanmıştır.

Bu ekran (sisteme dışardan veri yüklenmek istendiğinde kullanıcıya gösterilir) aracılığı ile sisteme yüklenecek verinin formatı belirlenebilmektedir. Kullanıcı dilerse ARFF, kategorize edilmiş metin belgesi (her bir kategorinin bir klasör ve o kategorideki her bir içeriğin bu klasör altında ayrı ayrı .txt uzantılı dosyalarda olması gerekir) ve tab ile bölünmüş metin belgesi (örnekler tek bir .txt uzantılı dosyada kategori + tab + içerik şeklinde temsil edilir) formatlı verileri otomatik olarak sisteme yükleyebilmektedir. İkinci olarak veri al aşamasında verinin hangi önişleme adımlarından geçirileceğini belirlemek amacıyla **Ek 2.1**'de verilen önişleme seçeneklerini belirleme ekranı aracılığıyla veri okunur ve önişlemden geçirilir.

Bu ekranda hem MS hem de DA çalışmalarında kullanılan teknikler kullanıcıya birlikte sunulmuş, yapılacak çalışmaya göre seçeneklerin belirlenmesi kullanıcıya bırakılmıştır. Önişleme seçenekleri belirleme ekranı aracılığıyla, öznitelik çıkarım modeli (BoW, Ngram, SSTF) ve ağırlıklandırma yöntemi (BINARY, TF, IDF, TF-IDF, Entropi, STF, ANTF) belirlenir. Ayrıca minimum terim uzunluğu, terim frekans filtresi, normalizasyon, kök bulma, durak kelimeleri çıkarma, anlamdaş kelimeleri gruplama, olumsuzlama, özel terimlerin (his simgeleri, hashtag, URL, kullanıcı adı vb.) öznitelik olarak kabul edilip edilmeyeceği seçenekleri de yine bu ekrandan belirlenebilmektedir.

Veri formatı ve önişleme seçenekleri belirlendikten sonra veri okunmakta ve önişlemden geçirilen veriyle ilgili istatistikler görülebilmektedir. Ayrıca liste görünümünde veri setinde bulunan her bir örnek tutulmakta örneğin önişlemden önceki ve sonraki içerikleri arayüz aracılığıyla görülebilmektedir. Bu özellik kullanıcının önişlemden geçirilmiş örnekler üzerinde meydana gelen değişimleri görebilmesi açısından önemlidir. Önişlemden geçirilmiş (işlenmiş) içerik seçilen öznitelik çıkarma modeline bağlı olarak farklı formatlarda kullanıcıya gösterilmektedir. Böylece kullanıcı veri setinde bulunan örneklerden herhangi biri üzerinde detaylı inceleme yapabilmektedir. Bunun yanı sıra herhangi bir dokümandan elde edilen öznitelikler ve ağırlıkları görülebilmektedir. Bu işlemlerin gerçekleştirildiği ekran **Ek 2.2**'de verilmiştir. Boyut indirgeme (öznitelik seçme) ekranında farklı öznitelik seçme yöntemleri seçenek olarak kullanıcıya sunulmuştur. Kullanıcı bu yöntemler için gerekli

parametreleri girdikten sonra öznitelik seçme işlemi uygulayabilmekte boyutu düşürülen veri setini dilerse kayıt edebilmektedir. **Ek 3.1**'de gösterilen ekran sistemdeki veri formatı ARFF olduğunda aktif olmaktadır.

Programımızda metin belgesi formatındaki veriler üzerinde ARFF dönüşümü uygulayabilmek için biçimlendirme adımı da uygulanmıştır. **Ek 3.2**'de verilen biçimlendir adımında ise sadece eğitim verisi veya eğitim ve test verileri birlikte biçimlendirilebilmektedir. Biçimlendirmekten kasıt ARFF dönüşümü yapmaktır. Bu ekran üzerinden de kullanıcı dilerse test verisi okuyabilmekte (önişleme aşamasında belirlenen seçenekler uygulanır) ve dönüşüm uygulayabilmektedir. Dönüştürülen veriyi kullanıcı dilerse ARFF formatında kayıt edebilmektedir. Bu ekran sistemdeki veri formatı ARFF ise kullanıcıya gösterilmez.

Bu adımlardan geçtikten sonra veri sınıflandırma adımıyla sınıflandırılır. Bu ekranda sistemde mevcut olan farklı sınıflandırma algoritmaları ile sınıflandırma yapmak mümkündür. Sistemde sınıflandırma yapabilmek için veri setinin işlenip ARFF formatına dönüştürülmüş olması gerekir. Programımızda bütünlüğü sağlamak adına WEKA alt yapısını kullanmayan algoritmalar da bu yapıyı destekleyecek şekilde uygulanmıştır. **Ek 4.1**'de görüntüsü verilen sınıflandırma ekranında farklı yöntem ve sınıflandırıcılar ile sınıflandırma yapmak mümkündür. Kullanıcı dilerse sınıflandırıcılar için gerekli parametreleri kendisine gösterilen ekranlarla değiştirebilmektedir. Programımızın son adımı olan ve **Ek 4.2**'de verilen analiz aşamasında ise en son yapılan sınıflandırma testi ile ilgili karmaşıklık matrisi grafiksel olarak gösterilmektedir.

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Bu tez kapsamında MS ve DA alanlarında araştırmalar yapılmış ve geliştirilen OMESIS yazılımı ile Türkçe ve İngilizce veri setleri üzerinde deneyler gerçekleştirilmiştir. Gerçekleştirilen deneylerden elde edilen bulgular ulusal ve uluslararası konferanslarda bildiri olarak sunulmuştur. Bu bölümde bu tez kapsamında gerçekleştirilen çalışmalarda elde edilen deneysel sonuçlar kısaca açıklanmıştır. Tüm deneylerde işlenen metne bağlı olarak değişebilen teknikler dışında ortak olarak Şekil 3.2'de verilen sistem modeli kullanılmıştır. Bu tezin konusu olan Türkçe Twitter DA alanında aşağıdaki çalışmalar gerçekleştirilmiştir:

- Türkçe Twitter Mesajlarının Duygu Analizi (Coban vd 2015a)
- Türkçe Twitter Duygu Analizi için Benzerlik ve Uzaklık Metriklerinin Karşılaştırılması (Coban vd 2015b)
- Türkçe Twitter Mesajları için LDA ile Duygu Sınıflandırması (Coban ve Ozyer 2016a)

Yukarıda verilen çalışmaların dışında DA'da TA yönteminin etkisini inceleyebilmek amacıyla farklı TA yöntemleri (MS ve Bilgi Getirimi başta olmak üzere diğer alanlar için önerilmiş) Twitter DA'ya uygulanmış ve kapsamlı bir karşılaştırma yapılmıştır. Bu çalışmamız da "Twitter Duygu Analizinde Terim Ağırlıklandırma Yönteminin Etkisi" başlığıyla değerlendirilmek üzere Dokuz Eylül Üniversitesi Fen ve Mühendislik Dergisi'ne gönderilmiştir.

Bunun yanı sıra bu tezde uygulanan klasik MS teknikleri kullanılarak MS alanında da aşağıda verilen çalışmalar gerçekleştirilmiş ve kısaca açıklanmasına gerek görülmüştür:

- Metin Sınıflandırma Teknikleri ile İstenmeyen Kısa Mesajların Otomatik Olarak Tespit Edilmesi (Bozan vd 2015)
- Türkçe Şarkı Sözlerinden Müzik Türü Sınıflandırması (Coban ve Ozyer 2016b)

4.1. Türkçe Twitter Mesajlarının Duygu Analizi

Bu çalışmada, TTM veri seti üzerinde DA çalışılmış ve klasik MS yöntemleri ile analiz edilerek Twitter mesajları için duygu tespiti (pozitif veya negatif) gerçekleştirilmiştir. Deneysel sonuçlar SVM, NB, MNB ve *k*-NN algoritmalarıyla elde edilmiştir. BoW ve Ngram model ile elde edilen öznitelikler vektör uzay modeli ile temsil edilmiş ve sonuçlar üzerindeki etkisi incelenmiştir. Her bir Twitter mesajının içerdiği duygu iki (pozitif, negatif) veya daha fazla kategoride (çok iyi, iyi, tatmin edici, kötü, çok kötü vb.) olmak üzere derecelendirilebilir. Bu bağlamda çalışmamızda, DA her bir duygu derecesinin bir kategoriyi temsil ettiği bir MS işlemi olarak ele alınmıştır (Sebastiani 1999; Prabowo and Thelwall 2009). Twitter mesajları incelendiğinde kendine özgü belirgin bazı özel terimlere (hashtag{#}, kullanıcı adı{@}, URL{http://} ve his simgeleri{:), -), :(} gibi) sahip olduğu görülmektedir. Bu nedenle önişleme aşamasında bu tür terimlerin tespit edilmesi oldukça önem teşkil etmektedir. Bu amaçla çalışmamızda, önişleme aşamasında sırasıyla aşağıdaki işlemler uygulanmıştır:

- İçerik bakımından aynı olduğundan tekrar eden, retweet ve retweeted mesajlar veri setinden çıkarılmıştır.
- Mesaj kategorileri içerdiği his simgelerine bakılarak belirlendiğinden her iki duygu (olumlu, olumsuz) ile ilişkilendirilen his simgelerinin birlikte gözlendiği mesajlar elenmiştir.
- Twitter mesajlarında sıklıkla gözlenen ve yukarıda bahsedilen özel terimlerin yanı sıra iki ve daha az sayıda karakter içeren terimler çıkarılmıştır. Ayrıca terim frekans filtresi uygulanmış ve minimum terim frekansı 2 olarak alınmıştır.

Durak kelimelerin çıkarılması için Şekil 3.3'te verilen Türkçe durak kelime listesi kullanılmış, kök bulma işlemi ise Türkçe DDİ kütüphanesi olan Zemberek ile gerçekleştirilmiştir. TA aşamasında özniteliklerin vektör uzay model ile temsil edilen mesaj vektörlerindeki ağırlığının belirlenmesi için BINARY, TF ve TF-IDF yöntemleri kullanılmıştır. Ayrıca öznitelik boyutunun artmasını engellemek amacıyla kelime içerisinde tekrar eden harf sayısı bire indirgenmiştir. Elde edilen işlenmiş veri ARFF

dosya formatına dönüştürüldükten sonra KTÖS yöntemi ile öznitelikler seçilmiş ve boyut indirgenmiştir.

Çizelge 4.1. TTM verisi ile ilgili istatistikler

Özellik	**Önişlem**	
	Önce	Sonra
Ortalama Terim	7,5	5,1
Pozitif Etiketli Örnek	6887	6269
Negatif Etiketli Örnek	7890	7043
Toplam Örnek	14777	13312

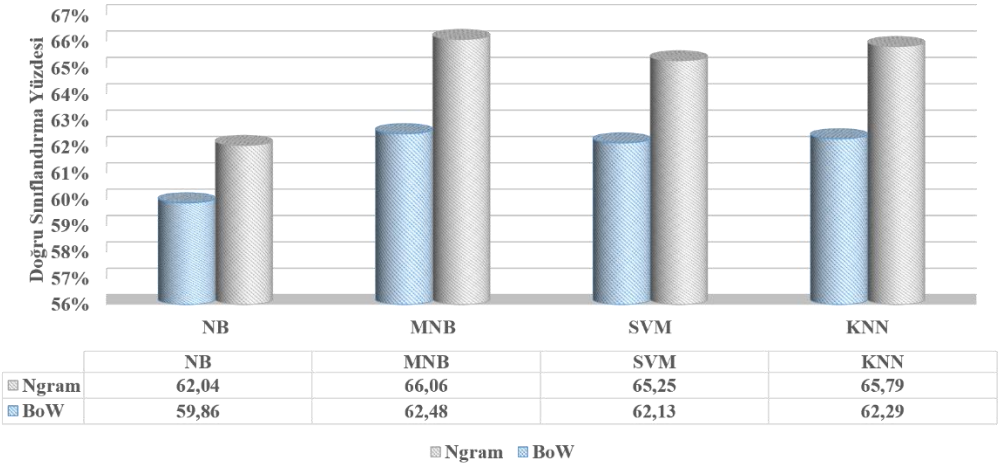
Çizelge 4.2. Önişlemden geçirilen TTM verisi için öznitelik istatistikleri

Elenen Öznitelik	**Sayısı**	**Yüzdesi (%)**
Yok (Tüm Terimler)	111316	100
(-) Hashtag	889	0,79
(-) URL	1913	1,71
(-) Kullanıcı Adı	7026	6,31
(-) His Simgeleri	14778	13,27
(-) Filtrelenen	18416	16,54
Toplam	**43022**	**38,62**

TTM verisinden tekrar eden, retweet ve retweeted mesajlar çıkarıldıktan sonra geriye 14777 mesaj kalmış, önişlemden geçirildikten sonra toplam mesaj sayısı 13312 olarak elde edilmiştir. Veri seti üzerinde önişlemin etkisini gösteren istatistikler ve elde edilen öznitelikler ile ilgili istatistiksel bilgiler sırasıyla Çizelge 4.1 ve Çizelge 4.2'de verilmiştir. Öznitelikler elde edilirken kök bulma, durak kelimelerin ve tekrarlanan harflerin çıkarılması işlemleri sadece BoW modelde uygulanmıştır. Bu nedenle Çizelge 4.1'de verilen ortalama terim sayısı sadece BoW model için geçerlidir.

Önişleme aşamasında his simgeleri dahil olmak üzere özel terimler elenerek sınıflandırıcı sadece önişlenmiş mesaj içeriğinden çıkarılan anlamlı kelime veya Ngram öznitelikleri ile eğitilmiştir. Ngram modelde, karakter seviye (kelime seviyeden daha başarılı olduğundan) öznitelikler (bigram, trigram ve four-gram olmak üzere üç farklı seviyede) kullanılmıştır. Sınıflandırma sonuçları ise 10-kat çapraz geçerleme yöntemi ile elde edilmiştir. Deneysel sonuçlar incelendiğinde, karakter seviye Ngram modelin

BoW modelden daha başarılı (en başarılı Ngram öznitelikleri trigram'lar olmuştur) olduğu tespit edilmiştir. En başarılı ağırlıklandırma yöntemi ise BINARY olmuştur. Bunun yanı sıra uygulanan öznitelik seçme işlemi ile 17720 Ngram ve 7569 BoW benzersiz özniteliklerinden sırasıyla 144 ve 122 tane öznitelik elde edilmiştir. Sınıflandırıcılar arasında her iki model için de en başarılı yöntem MNB olmuştur. SVM sınıflandırıcısı için doğrusal çekirdek kullanılmış, *k*-NN sınıflandırmasında en yakın komşu sayısı *k*=1 alınmıştır.



	NB	MNB	SVM	KNN
Ngram	62,04	66,06	65,25	65,79
BoW	59,86	62,48	62,13	62,29

Şekil 4.1. Sınıflandırma yöntemlerinin BoW ve Ngram modeldeki başarıları

Farklı sınıflandırma yöntemleri kullanılarak BoW ve Ngram modelde elde edilen sonuçlar Şekil 4.1'de gösterilmiştir. Bunun yanı sıra his simgeleri kullanılarak gerçekleştirilen duygu sınıflandırmasının (etiketleme) başarısını gözlemleyebilmek amacıyla her iki kategori için (pozitif, negatif) sadece tek bir kategoride ve ortak olarak her ikisinde en sık gözlenen kelimeler tespit edilmiştir. Köküne indirgenmiş BoW özniteliklerinden ortak olarak gözlenen 30 ve sadece bir kategoride gözlenen 20 kelime (her iki kategori için) sırasıyla Şekil 4.2 ve Şekil 4.3'te verilmiştir. Bu çalışmada klasik MS teknikleri kullanılarak TTM verisi üzerinde DA gerçekleştirilmiş ve önceki çalışmalarda temel yöntemler kullanılarak elde edilen başarı oranı Türkçe için yakalanmıştır.

Şekil 4.2. Her iki kategoride en sık gözlenen kelimeler

(a) (b)

Şekil 4.3. Pozitif (a) ve negatif (b) kategorilerde en sık gözlenen kelimeler

Bu araştırmadan elde edilen bulgulara göre; duygu sınıflandırması aşamasında his simgelerinin kullanılması, bu simgelerin önişlem aşamasında mesaj içeriklerinden çıkarılması ve Twitter mesajlarının herhangi bir konu gözetilmeksizin rastgele toplanması DA başarısını düşürmüştür. Ayrıca Şekil 4.2'den de anlaşılacağı üzere pozitif kategoride ayırt edici olması beklenen kelimelerin ("iyi, güzel, günaydın, yeni" gibi) negatif kategorili mesajlarda da çok sık gözleniyor olması BoW modelde başarının düşmesine neden olmuştur.

4.2. Türkçe Twitter Duygu Analizi için Benzerlik ve Uzaklık Metriklerinin Karşılaştırılması

Bu çalışmada, Türkçe Twitter mesajlarının duygu analizi için *k*-NN yöntemi ile kullanılan uzaklık ve benzerlik fonksiyonlarının karşılaştırması yapılmıştır. Sistem modeli olarak Şekil 3.2'de verilen model kullanılmış, deneyler R8 ve TTM olmak üzere iki farklı veri seti üzerinde gerçekleştirilmiştir. Benzerlik fonksiyonlarının farklı veri setlerinde sınıflandırma başarısına yaptığı katkı bakımından aynı davranışları gösterip göstermediğini incelemek ve kıyaslama yapabilmek amacıyla iki farklı veri seti kullanılmıştır. Çalışmanın temel amacı benzerlik ve uzaklık fonksiyonlarının (metrik) genişletilmiş karşılaştırmasını yaparak SMTP algoritmasının diğer metriklerden daha iyi sonuç verip vermediğini incelemek ve bu çalışmayı DA için uygulamaktır. Çalışmamızda, literatürde yapılan çalışmalardan farklı olarak şu katkılar sağlanmıştır:

- Literatür taraması yapılmış ve DA alanında Türkçe bir veri seti üzerinde benzerlik fonksiyonları için performans karşılaştırmasını esas alan akademik bir çalışma olmadığı tespit edilmiştir. Bu bakımdan çalışmamız DA alanında Türkçe için belirtilen amaçla yapılmış ilk çalışma olma özelliğini taşımaktadır.
- Referans alınan çalışmada karşılaştırması yapılan metriklere Manhattan, Chebyshev ve Dice metrikleri de eklenmiş, ağırlıklandırma aşamasında dört farklı yöntem kullanılarak karşılaştırma sonuçları genişletilmiştir. Deneyler sınıflandırma sonuçlarına olan etkisini inceleyebilmek amacıyla öznitelik seçme işlemi uygulanarak elde edilmiştir.
- DA çalışmalarında Ngram öznitelikleri genellikle kelime seviyesinde çıkarılırken çalışmamızda karakter seviyesinde çıkarılmıştır.

Çalışmamızda kullandığımız sistem modeli gereği, her iki veri seti dil bağımlı olmak üzere önişlemeye tabi tutulmuştur. TTM verisi Türkçe mesajlardan oluştuğu için daha önceki çalışmamızda (Coban vd 2015a) uygulanan önişleme adımlarından geçirilmiştir. R8 verisi İngilizce olduğu için önişleme aşamasının durak kelime çıkarma ve kök bulma adımlarında farklı işlemler uygulanmıştır. Durak kelimelerin çıkarılması için Şekil

3.3'te İngilizce için verilen durak kelime listesi kullanılmış, kök bulma işlemi için Porter Stemmer (Porter 1980) algoritması kullanılmıştır. Öznitelik çıkarım aşamasında, BoW ve Ngram (bigram ve trigram) modellerinde öznitelik çıkarılmıştır.

BoW modelde özniteliklerin içerdiği tekrarlanan harfler İngilizce için ikiye, Türkçe için bire düşürülmüştür. Ağırlıklandırma aşamasında BINARY, TF, TF-IDF ve Entropi yöntemleri ile ağırlıklandırma yapılmıştır. ARFF dönüşümü uygulandıktan sonra öznitelik seçme işlemi için KTÖS yöntemi kullanılmıştır. Sınıflandırma aşamasında 3.2.3 başlığı altında açıklanan *k*-NN algoritması ile farklı benzerlik ve uzaklık fonksiyonları kullanılmış sonuçlar 10-kat çapraz geçerleme yöntemi ile elde edilmiştir. Önişleme aşamasında her iki veri seti için de terim frekans filtresi iki alınmış ve minimum terim sayısı ikiden az olan doküman veya mesajlar veri setinden çıkarılmıştır. Önişlem önce ve sonrasında her iki veri setine ait karakteristik özellikler (örnek sayıları) sırasıyla Çizelge 4.3 ve Çizelge 4.4'te verilmiştir.

Çizelge 4.3. Önişlem önce ve sonrasında TTM verisi karakteristiği

Kategori	**Önişleme**	
	Önce	Sonra
Pozitif	10000	6029
Negatif	10000	6757
Toplam	**20000**	**12786**

Çizelge 4.4. Önişlem önce ve sonrasında R8 verisi karakteristiği

Önişlem önce ve sonrası			
Kategori	Örnek Sayısı	Kategori	Örnek Sayısı
earn	2480	acq	1596
trade	251	ship	108
grain	41	crude	253
interest	190	money-fx	206

Çizelge 4.4'ten anlaşılacağı üzere R8 verisinin karakteristiğinde önişlemden sonra bir değişiklik olmamıştır. Uygulanan önişlemin her iki veri seti üzerindeki etkisi ise Çizelge 4.5'te verilmiştir.

Çizelge 4.5. TTM ve R8 verileri üzerinde önişlemin etkisi

Elenen Öznitelik	**Öznitelik Sayısı**		**Yüzdesi (%)**	
	R8	TTM	R8	TTM
Yok (Tüm Terimler)	577453	141787	100	100
(-) Hashtag	-	1090	-	0,76
(-) URL	-	2188	-	1,54
(-) Kullanıcı Adı	-	9049	-	6,38
(-) His Simgesi	-	16372	-	11,54
(-) Filtrelenen	312656	33915	54,1	23,91
Toplam	**312656**	**62614**	**54,1**	**44,16**

Çizelge 4.6. TTM ve R8 verileri için çıkarılan öznitelik sayıları

Benzersiz öznitelik sayısı		**R8**	**TTM**
BoW		14910	4060
Ngram	Bigram	687	973
	Trigram	8364	10332

Çizelge 4.7. TTM ve R8 için BoW ve Ngram modelde seçilen öznitelik sayıları

Veri / Model	**Ağırlıklandırma Yöntemi**			
	BINARY	TF	TF-IDF	Entropi
R8-BoW	34	42	43	34
TTM-BoW	93	94	94	93
R8-Bigram	29	39	39	29
R8-Trigram	59	66	66	59
TTM-Bigram	21	30	30	21
TTM-Trigram	85	85	84	84

Öznitelik çıkarım aşamasında farklı modellerde her iki veri seti için çıkarılan özniteliklere ilişkin bilgiler Çizelge 4.6'da, farklı ağırlıklandırma yöntemi ve veri seti kombinasyonları için seçilen benzersiz (unique) öznitelik sayısına ilişkin bilgiler ise Çizelge 4.7'de verilmiştir. Deneysel sonuçlar geliştirdiğimiz OMESIS yazılımı ile alınmış, farklı metriklerle *k*-NN sınıflandırıcısının *k* parametresi için [1-15] aralığındaki farklı değerlerde toplam 1728 sınıflandırma testi yapılmıştır. Sınıflandırma aşamasında SMTP metriği için λ parametresi 1 alınmıştır. Sonuçlar her iki veri seti için iki farklı modelde elde edilmiş olup, TTM verisi için Ngram modelde elde edilen ve Çizelge 4.8 ile Çizelge 4.9'da verilen sonuçlardan elde edilen bulgular şöyledir:

- Euclidean ve Manhattan metrikleri bigram ve trigram modelde en başarılı metrik olmuştur. Trigram tüm metriklerde, bigram modelden daha başarılı olmuş ve her iki modelde en başarılı sonuçlar TF ağırlıklandırmasıyla alınmıştır (birkaç istisna hariç).

R8 verisi için Ngram modelde elde edilen ve Çizelge 4.10 ile Çizelge 4.11'de verilen sonuçlardan elde edilen bulgular ise şunlardır:

- Trigram tüm metriklerde bigram modelden daha başarılı olmuştur (birkaç istisna hariç) ve her iki modelde en başarılı sonuçlar TF + IT-Sim kombinasyonu ile elde edilmiştir.
- Bigram modelde, BINARY ve Entropi ağırlıklandırmalarında en başarılı sonuçlar Euclidean ve Manhattan metrikleri ile elde edilmiştir.
- Trigram modelde TF-IDF ve Entropi ağırlıklandırmasında en başarılı metrik SMTP iken, TF ağırlıklandırmasında en başarılı metrik IT-Sim olmuştur. Her iki modelde en başarılı sonuçlar TF + IT-Sim kombinasyonu ile elde edilmiş, SMTP Trigram modelde sadece TF-IDF ağırlıklandırması için en iyi metrik olmuştur.

TTM verisi için BoW modelde elde edilen ve Çizelge 4.12'de verilen sonuçlardan elde edilen bulgular şöyledir:

- SMTP ve IT-Sim sonuçların hiç birisinde en iyi metrik olamamış, en başarılı sonuçlar Euclidean ve Manhattan yöntemleri ile alınmıştır. En başarılı ağırlıklandırma yöntemleri ise BINARY ve Entropi yöntemleri olmuştur.

R8 verisi için BoW modelde elde edilen ve Çizelge 4.13'te verilen sonuçlardan elde edilen bulgular ise şunlardır:

- SMTP metriği sadece Entropi ağırlıklandırması ile en iyi metrik olmuş ($k \geq 3$) ancak TF ve TF-IDF ağırlıklandırmalarında en başarılı metrik IT-Sim olmuştur. En başarılı sonuçlar TF + IT-Sim kombinasyonu ile elde edilmiştir.

Çizelge 4.8. TTM verisi için Bigram modelde *k*-NN algoritması kullanılarak farklı benzerlik ve ağırlıklandırma yöntemleriyle elde edilen sınıflandırma sonuçları (%)

Model: Bigram		***k=1***	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***	***k=13***	***k=15***
Binary	Euclidean	**59,84**	**59,96**	**59,92**	**59,92**	**59,86**	**59,88**	**59,88**	**59,90**
	Manhattan	**59,84**	**59,96**	**59,92**	**59,92**	**59,86**	**59,88**	**59,88**	**59,90**
	Chebyshev	59,77	59,68	59,40	59,23	59,17	59,15	59,09	59,02
	Cosine	54,71	55,46	55,67	56,99	57,99	57,93	57,58	57,62
	Dice	54,72	55,46	55,67	56,99	57,19	57,93	57,88	57,62
	E. Jaccard	54,72	55,46	55,67	56,99	57,19	57,93	57,88	57,62
	Pairwise	54,02	54,39	55,31	55,48	55,46	56,14	55,31	55,98
	IT-Sim	54,82	55,50	55,73	57,03	57,18	57,91	57,57	57,68
	SMTP	54,72	55,46	55,67	56,99	57,19	57,93	57,58	57,62
Entropi	Euclidean	**59,84**	**59,96**	**59,92**	**59,92**	**59,86**	**59,88**	**59,88**	**59,90**
	Manhattan	**59,84**	**59,96**	**59,92**	**59,92**	**59,86**	**59,88**	**59,88**	**59,90**
	Chebyshev	59,77	59,68	59,40	59,23	59,17	59,15	59,09	59,02
	Cosine	54,84	55,52	55,69	57,03	57,19	57,94	57,61	57,68
	Dice	54,84	55,52	55,69	57,03	57,19	57,94	57,61	57,68
	E. Jaccard	54,84	55,52	55,69	57,03	57,19	57,94	57,61	57,68
	Pairwise	53,89	53,99	53,67	55,42	55,38	55,72	55,62	55,29
	IT-Sim	54,85	55,51	55,69	57,08	57,18	57,93	57,62	57,78
	SMTP	54,72	55,46	55,67	56,99	57,19	57,93	57,58	57,62
TF	Euclidean	58,57	59,47	**60,07**	60,29	60,46	**60,72**	**60,91**	**61,07**
	Manhattan	**58,62**	**59,53**	59,78	60,36	60,31	60,70	60,80	61,05
	Chebyshev	**58,62**	59,31	59,94	**60,38**	**60,62**	60,70	60,81	60,91
	Cosine	56,47	57,58	57,83	58,70	59,33	59,22	59,18	59,40
	Dice	56,72	57,84	58,29	59,00	59,11	59,12	59,43	59,93
	E. Jaccard	56,72	57,84	58,29	59,00	59,11	59,12	59,43	59,93
	Pairwise	56,35	56,64	57,26	57,38	57,35	57,22	56,31	56,72
	IT-Sim	56,57	57,80	57,75	59,02	59,51	59,18	59,38	59,82
	SMTP	56,55	57,60	58,23	59,01	59,21	59,42	59,38	59,40
TF-IDF	Euclidean	58,55	59,26	**59,96**	60,29	**60,60**	60,71	**60,97**	61,10
	Manhattan	**58,69**	**59,44**	59,75	60,35	60,39	**60,78**	60,92	**61,13**
	Chebyshev	58,56	58,87	59,90	**60,37**	60,44	60,57	60,65	60,68
	Cosine	56,60	57,83	58,20	58,97	59,67	59,44	59,67	59,84
	Dice	57,03	57,90	58,46	59,48	59,90	59,54	59,77	59,75
	E. Jaccard	57,03	57,90	58,46	59,48	59,90	59,54	59,77	59,75
	Pairwise	54,14	54,60	55,82	57,00	56,99	57,61	57,30	57,33
	IT-Sim	56,82	57,74	57,80	58,58	59,22	59,09	59,26	59,75
	SMTP	56,56	57,58	58,29	58,99	59,21	59,43	59,33	59,35

Çizelge 4.9. TTM verisi için Trigram modelde *k*-NN algoritması kullanılarak farklı benzerlik ve ağırlıklandırma yöntemleriyle elde edilen sınıflandırma sonuçları (%)

Model: Trigram		***k=1***	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***	***k=13***	***k=15***
Binary	Euclidean	**62,19**	**62,17**	**62,19**	**62,34**	**62,47**	**62,45**	62,41	62,38
	Manhattan	**62,19**	**62,17**	**62,19**	**62,34**	**62,47**	**62,45**	62,41	62,38
	Chebyshev	61,65	60,92	60,54	60,36	60,26	60,02	59,76	59,50
	Cosine	58,01	59,32	60,02	60,47	60,91	61,52	62,09	61,84
	Dice	58,00	59,33	60,02	60,47	60,93	61,52	62,10	61,86
	E. Jaccard	58,00	59,33	60,02	60,47	60,93	61,52	62,10	61,86
	Pairwise	55,26	56,02	57,04	58,09	58,61	58,97	59,26	59,66
	IT-Sim	58,46	59,56	60,58	61,06	61,42	61,84	**62,42**	**62,44**
	SMTP	58,00	59,33	60,02	60,47	60,93	61,52	62,10	61,86
Entropi	Euclidean	**62,19**	**62,17**	**62,19**	**62,34**	**62,47**	**62,45**	62,41	62,38
	Manhattan	**62,19**	**62,17**	**62,19**	**62,34**	**62,47**	**62,45**	62,41	62,38
	Chebyshev	61,65	60,92	60,54	60,36	60,26	60,02	59,76	59,50
	Cosine	58,50	59,69	60,50	60,96	61,13	61,86	62,36	62,26
	Dice	58,51	59,68	60,46	60,96	61,15	61,85	62,34	62,32
	E. Jaccard	58,51	59,68	60,46	60,96	61,15	61,85	62,34	62,32
	Pairwise	58,16	58,86	59,54	60,12	60,77	61,21	61,42	60,85
	IT-Sim	58,63	59,66	60,47	60,90	61,27	61,94	**62,53**	**62,42**
	SMTP	58,00	59,33	60,02	60,47	60,93	61,52	62,10	61,86
TF	Euclidean	62,06	62,30	62,56	**62,79**	**63,04**	**63,14**	**63,10**	**63,20**
	Manhattan	**62,20**	62,41	**62,61**	62,76	62,86	62,96	63,09	63,09
	Chebyshev	62,08	**62,59**	62,53	62,56	62,50	62,57	62,63	62,53
	Cosine	58,34	59,30	60,08	60,96	61,07	62,17	61,68	61,64
	Dice	58,47	59,28	60,14	60,98	61,06	62,12	61,92	61,84
	E. Jaccard	58,47	59,28	60,14	60,98	61,06	62,12	61,92	61,84
	Pairwise	55,07	56,09	57,40	58,32	59,09	59,45	59,47	59,28
	IT-Sim	58,86	59,91	60,58	61,42	61,86	62,61	62,41	62,55
	SMTP	58,34	59,24	60,32	60,80	61,00	62,05	61,88	61,82
TF-IDF	Euclidean	62,01	62,17	62,50	62,53	**62,88**	62,92	**62,94**	**63,03**
	Manhattan	**62,03**	62,23	62,34	**62,56**	62,71	62,68	62,89	62,95
	Chebyshev	**62,03**	**62,45**	**62,53**	62,45	62,48	62,55	62,65	62,68
	Cosine	59,34	60,20	61,00	61,68	62,02	62,95	62,49	62,53
	Dice	59,52	60,36	61,38	61,88	62,21	**62,99**	62,68	62,78
	E. Jaccard	59,52	60,36	61,38	61,88	62,21	**62,99**	62,68	62,78
	Pairwise	57,92	59,47	59,51	60,85	61,34	62,24	61,80	61,92
	IT-Sim	59,08	60,18	60,90	61,41	61,59	62,49	62,32	62,53
	SMTP	58,33	59,25	60,31	60,84	60,98	62,05	61,90	61,82

Çizelge 4.10. R8 verisi için Bigram modelde *k*-NN algoritması kullanılarak farklı benzerlik ve ağırlıklandırma yöntemleriyle elde edilen sınıflandırma sonuçları (%)

Model: Bigram		***k=1***	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***	***k=13***	***k=15***
Binary	Euclidean	**76,59**	**77,50**	**78,37**	**78,50**	**78,70**	**78,90**	**78,68**	**78,72**
	Manhattan	**76,59**	**77,50**	**78,37**	**78,50**	**78,70**	**78,90**	**78,68**	**78,72**
	Chebyshev	61,73	55,49	54,03	53,61	53,19	53,05	52,88	52,65
	Cosine	75,09	76,44	77,01	77,93	77,84	78,24	78,30	78,50
	Dice	75,09	76,42	76,93	77,73	77,79	78,17	78,12	78,50
	E. Jaccard	75,09	76,42	76,93	77,73	77,79	78,17	78,12	78,50
	Pairwise	63,84	64,13	64,50	66,07	67,27	69,00	70,64	71,55
	IT-Sim	74,31	75,62	76,02	77,22	77,57	77,64	77,55	77,75
	SMTP	75,09	76,42	76,93	77,73	77,79	78,17	78,12	78,50
Entropi	Euclidean	**76,59**	**77,50**	**78,37**	**78,50**	**78,70**	**78,90**	**78,68**	**78,72**
	Manhattan	**76,59**	**77,50**	**78,37**	**78,50**	**78,70**	**78,90**	**78,68**	**78,72**
	Chebyshev	61,73	55,49	54,03	53,61	53,19	53,05	52,88	52,65
	Cosine	74,58	75,75	75,97	75,97	76,97	77,21	77,32	77,48
	Dice	74,47	75,71	75,98	76,90	76,84	77,10	77,33	77,15
	E. Jaccard	74,47	75,71	75,98	76,90	76,84	77,10	77,33	77,15
	Pairwise	61,76	67,38	71,61	72,79	73,58	73,81	74,05	74,27
	IT-Sim	73,74	74,87	75,15	76,33	76,17	76,77	76,99	76,64
	SMTP	75,09	76,42	76,93	77,73	77,79	78,17	78,12	78,50
TF	Euclidean	81.20	82.29	82,88	83,00	83,22	82,84	82,82	83,00
	Manhattan	**82,46**	82,88	83,71	83,62	84,01	83,88	84,01	**84,32**
	Chebyshev	78,15	78,14	78,61	78,08	78,10	77,77	77,39	77,04
	Cosine	80,56	81,05	81,60	82,07	81,95	82,00	81,91	82,46
	Dice	80,27	80,56	81,47	81,49	81,85	81,62	81,93	81,60
	E. Jaccard	80,27	80,56	81,47	81,49	81,85	81,62	81,93	81,60
	Pairwise	53,19	51,68	53,07	58,21	65,70	72,32	75,40	77,17
	IT-Sim	82,33	**83,26**	**83,97**	**84,10**	**84,48**	**84,52**	**84,32**	84,22
	SMTP	78,63	78,63	80,69	80,92	81,42	81,23	81,45	81,49
TF-IDF	Euclidean	81,22	82,29	82,88	83,02	83,28	82,86	82,80	82,97
	Manhattan	**82,46**	**82.88**	**83,73**	**83,62**	**83,99**	**83,84**	**84,02**	**84,32**
	Chebyshev	78,08	78,17	78,52	78,17	78,19	78,17	77,64	77,26
	Cosine	80,65	81,75	82,53	82,73	83,00	82,82	82,84	82,73
	Dice	80,78	81,42	82,20	82,11	82,24	82,38	82,58	82,55
	E. Jaccard	80,78	81,42	82,20	82,11	82,24	82,38	82,58	82,55
	Pairwise	74,82	78,10	81,40	81,78	82,53	82,49	82,37	82,42
	IT-Sim	80,09	81,34	81,82	82,33	82,53	82,13	81,95	82,11
	SMTP	78,65	79,85	80,69	80,91	81,42	81,23	81,45	81,49

Çizelge 4.11. R8 verisi için Trigram modelde *k*-NN algoritması kullanılarak farklı benzerlik ve ağırlıklandırma yöntemleriyle elde edilen sınıflandırma sonuçları (%)

Model: Trigram		***k=1***	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***	***k=13***	***k=15***
Binary	Euclidean	**88,82**	88,40	88,45	88,22	87,96	87,96	87,78	87,73
	Manhattan	**88,82**	88,40	88,45	88,22	87,96	87,96	87,78	87,73
	Chebyshev	58,50	53,78	53,16	52,72	52,45	52,30	52,30	52,30
	Cosine	88,76	**89,66**	**89,73**	**89,77**	89,69	90,02	**89,93**	**89,84**
	Dice	88,58	89,51	**89,73**	89,73	**89,82**	**89,73**	89,82	89,75
	E. Jaccard	88,58	89,51	**89,73**	89,73	**89,82**	**89,73**	89,82	89,75
	Pairwise	67,45	73,34	74,34	75,02	76,59	77,83	78,96	83,09
	IT-Sim	88,11	88,80	88,89	89,02	89,33	89,62	89,40	89,44
	SMTP	88,58	89,51	**89,73**	89,73	**89,82**	**89,73**	89,82	89,75
Entropi	Euclidean	**88,82**	88,40	88,45	88,22	87,96	87,96	87,78	87,73
	Manhattan	**88,82**	88,40	88,45	88,22	87,96	87,96	87,78	87,73
	Chebyshev	58,50	53,78	53,16	52,72	52,45	52,30	52,30	52,30
	Cosine	87,62	87,85	88,25	88,16	88,24	88,36	88,36	88,45
	Dice	87,45	87,67	88,18	88,18	87,91	88,38	88,27	88,18
	E. Jaccard	87,45	87,67	88,18	88,18	87,91	88,38	88,27	88,18
	Pairwise	70,73	78,12	81,85	84,74	85,30	85,43	85,26	85,37
	IT-Sim	86,54	87,23	87,62	87,83	87,98	87,82	88,24	88,00
	SMTP	88,58	**89,51**	**89,73**	**89,73**	**89,82**	**89,73**	**89,82**	**89,75**
TF	Euclidean	90,26	90,35	90,24	90,11	90,17	90,26	90,39	90,44
	Manhattan	90,00	89,97	90,42	90,42	90.53	90,39	90,17	90,28
	Chebyshev	87,16	86,21	85,45	85,06	84,88	84,43	84,17	83,70
	Cosine	90,06	90,48	90,73	90,68	90,70	90,55	90,66	90,55
	Dice	90,04	90,50	90,55	90,68	90,86	90,75	90,72	90,37
	E. Jaccard	90,04	90,50	90,55	90,68	90,86	90,75	90,72	90,37
	Pairwise	74,23	78,96	80,40	81,38	83,77	88,58	89,38	89,79
	IT-Sim	**91,23**	**91,79**	**92,10**	**92,10**	**92,08**	**92,14**	**92,12**	**92,21**
	SMTP	89,64	90,66	90,61	90,72	90,84	90,92	90,90	90,82
TF-IDF	Euclidean	90,24	90,31	90,22	90,13	90,15	90,35	90,42	90,48
	Manhattan	**90,48**	89.95	90,35	90,44	90,51	90,50	90,17	90,22
	Chebyshev	56,33	86,87	86,03	85,63	85,76	85,30	85,15	84,86
	Cosine	89,64	90,10	90,37	89,91	90,15	90,10	90,20	90,02
	Dice	89,77	90,06	90,20	90,20	90,48	90,48	90,46	90,57
	E. Jaccard	89,77	90,06	90,20	90,20	90,48	90,48	90,46	90,57
	Pairwise	82,04	87,16	88,38	88,64	89,48	89,66	89,69	89,55
	IT-Sim	90,20	90,46	90,17	90,22	90,55	90,82	90,72	90,48
	SMTP	89,64	**90,66**	**90,61**	**90,72**	**90,84**	**90,92**	**90,90**	**90,82**

Çizelge 4.12. TTM verisi için BoW modelde *k*-NN algoritması kullanılarak farklı benzerlik ve ağırlıklandırma yöntemleriyle elde edilen sınıflandırma sonuçları (%)

Model: BoW		***k=1***	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***	***k=13***	***k=15***
Binary	Euclidean	**61,48**	**60,76**	60,60	**60,58**	**60,48**	**60,41**	**60,21**	**59,87**
	Manhattan	**61,48**	**60,76**	60,60	**60,58**	**60,48**	**60,41**	**60,21**	**59,87**
	Chebyshev	61,34	60,66	**60,66**	60,37	60,26	59,98	59,76	59,41
	Cosine	55,36	58,36	59,14	58,84	59,00	58,61	58,67	58,50
	Dice	55,36	58,36	59,14	58,84	59,00	58,61	58,67	58,50
	E. Jaccard	55,36	58,36	59,14	58,84	59,00	58,61	58,67	58,50
	Pairwise	55,14	57,96	59,22	58,80	59,01	58,65	58,81	58,68
	IT-Sim	55,40	58,42	59,18	58,93	59,11	58,83	58,91	58,65
	SMTP	55,36	58,36	59,14	58,84	59,00	58,61	58,67	58,50
Entropi	Euclidean	**61,48**	**60,76**	**60,60**	**60,58**	**60,48**	**60,41**	**60,21**	**59,87**
	Manhattan	**61,48**	**60,76**	**60,60**	**60,58**	**60,48**	**60,41**	**60,21**	**59,87**
	Chebyshev	61,34	60,66	60,40	60,37	60,26	59,98	59,76	59,41
	Cosine	55,41	58,47	59,18	58,88	59,14	58,72	58,77	58,60
	Dice	55,41	58,47	59,18	58,88	59,14	58,72	58,77	58,60
	E. Jaccard	55,41	58,47	59,18	58,88	59,14	58,72	58,77	58,60
	Pairwise	55,13	57,96	58,99	58,59	58,80	58,44	58,60	58,68
	IT-Sim	55,39	58,45	59,17	58,92	59,11	58,83	58,91	58,68
	SMTP	55,36	58,36	59,14	58,84	59,00	58,61	58,67	58,50
TF	Euclidean	61,15	60,61	60,46	60,46	**60,47**	**60,33**	**60,20**	**59,96**
	Manhattan	61,16	**60,62**	**60,47**	**60,48**	60,45	60,32	60,13	59,93
	Chebyshev	**61,17**	60,57	60,40	60,37	60,28	60,09	59,89	59,54
	Cosine	55,67	58,23	58,68	58,36	58,83	58,54	57,98	58,41
	Dice	55,51	58,35	58,63	58,25	58,89	58,59	57,93	58,23
	E. Jaccard	55,51	58,35	58,63	58,25	58,89	58,59	57,93	58,23
	Pairwise	54,80	57,76	58,59	58,23	58,32	58,32	58,16	58,42
	IT-Sim	55,82	58,46	58,89	58,57	59,14	58,90	58,26	58,75
	SMTP	55,58	58,42	58,71	58,28	58,97	58,75	58,11	58,47
TF-IDF	Euclidean	61,09	60,60	60,43	**60,44**	**60,44**	**60,30**	**60,16**	**59,92**
	Manhattan	61,09	60,62	60,44	**60,44**	60,40	60,26	60,09	59,84
	Chebyshev	**61,15**	**60,64**	**60,47**	**60,44**	60,40	60,28	60,09	59,76
	Cosine	55,76	58,42	58,78	58,45	59,01	58,74	58,11	58,57
	Dice	55,70	58,56	58,76	58,30	59,07	58,80	58,06	58,41
	E. Jaccard	55,70	58,56	58,76	58,30	59,07	58,80	58,06	58,41
	Pairwise	54,97	57,88	58,66	58,66	58,73	58,35	57,86	58,40
	IT-Sim	55,80	58,35	58,83	58,55	59,10	58,92	58,29	58,83
	SMTP	55,57	58,41	58,71	58,28	58,97	58,75	58,11	58,47

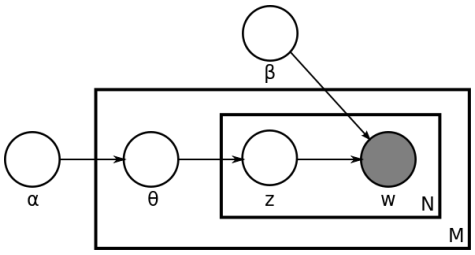
Çizelge 4.13. R8 verisi için BoW modelde *k*-NN algoritması kullanılarak farklı benzerlik ve ağırlıklandırma yöntemleriyle elde edilen sınıflandırma sonuçları (%)

Model: BoW		***k=1***	***k=3***	***k=5***	***k=7***	***k=9***	***k=11***	***k=13***	***k=15***
Binary	Euclidean	**90,31**	**90,22**	89,84	89,66	89,49	89,35	89,26	88,86
	Manhattan	**90,31**	**90,22**	89,84	89,66	89.49	89.35	89,26	88,86
	Chebyshev	78,12	70,22	66,43	64,28	61,95	59,28	57,66	56,57
	Cosine	88,29	89,31	89,99	**90,50**	**90,44**	**90,33**	90,10	90,10
	Dice	88,31	89,33	89,99	90,46	90,42	90,26	90,04	90,08
	E. Jaccard	88,31	89,33	89,99	90,46	90,42	90,26	90,04	90,08
	Pairwise	76,31	77,39	78,24	77,81	78,21	80,63	80,96	81,23
	IT-Sim	88,47	89,79	**90,08**	90,11	90,30	90,15	**90,46**	**90,44**
	SMTP	88,31	89,33	89,99	90,46	90,42	90,26	90,04	90,08
Entropi	Euclidean	**90,31**	**90,22**	89,84	89,66	89,49	89,35	89,26	88,86
	Manhattan	**90,31**	**90,22**	89,84	89,66	89,49	89,35	89,26	88,86
	Chebyshev	78,12	70,22	66,43	64,28	61,95	59,28	57,66	56,57
	Cosine	88,05	88,80	89,11	89,06	89,22	89,18	89,17	89,17
	Dice	88,04	88,87	89,24	89,04	89,27	89,35	89,18	89,22
	E. Jaccard	88,04	88,87	89,24	89,04	89,27	89,35	89,18	89,22
	Pairwise	77,90	78,52	79,27	80,60	82,40	83,04	83,00	83,19
	IT-Sim	88,09	89,15	89,40	89,29	89,42	89,60	89,58	89,79
	SMTP	88,31	89,33	**89,99**	**90,46**	**90,42**	**90,26**	**90,04**	**90,08**
TF	Euclidean	90.59	90.82	91,13	90,84	91,04	90,86	90,55	90,72
	Manhattan	**91.04**	91,23	91,24	91,06	91,70	91,34	91,43	91,34
	Chebyshev	90.02	89,42	89,27	88,93	88,69	87,87	87,42	86,89
	Cosine	89,11	90,44	90,57	90,57	90,81	90,93	90,79	90,81
	Dice	88,84	90,19	90,55	90,22	90,41	90,30	90,48	90,50
	E. Jaccard	88,84	90,19	90,55	90,22	90,41	90,30	90,48	90,50
	Pairwise	68,24	73,65	73,40	74,93	77,26	77,68	77,93	77,92
	IT-Sim	90,97	**92,21**	**92,26**	**92,37**	**92,52**	**92,41**	**92,56**	**92,67**
	SMTP	88,95	90,53	90,84	90,93	91,01	91,13	90,81	90,90
TF-IDF	Euclidean	90,55	90,13	90,24	89,84	89,89	89,99	89,75	89,68
	Manhattan	90,79	90,53	90,79	90,82	90,72	90,31	90,13	89,86
	Chebyshev	89,22	88,86	88,71	88,42	87,60	87,21	86,54	86,19
	Cosine	90,70	91,37	91,35	91,32	91,26	91,19	91,24	91,06
	Dice	90,37	91,21	91,41	91,48	91,46	91,26	91,04	91,21
	E. Jaccard	90,37	91,21	91,41	91,48	91,46	91,26	91,04	91,21
	Pairwise	74,84	82,73	83,91	84,50	85,26	85,97	86,87	87,69
	IT-Sim	**90,97**	**91,77**	**92,05**	**91,97**	**91,99**	**92,06**	**91,75**	**91,66**
	SMTP	89,62	90,48	90,92	91,03	91,12	91,10	91,13	91,06

Dice ve Extended Jaccard metrikleri ise bütün testlerde aynı sonuçları vermiştir. Tüm bu veriler ışığında referans aldığımız çalışmanın aksine TF-IDF ağırlıklandırmasında genellikle IT-Sim metriğinin daha başarılı olduğu görülmüştür. R8 için TF ve TF-IDF yöntemlerinin, TTM verisi için ise Boolean ve Entropi yöntemlerinin daha etkili olduğu tespit edilmiştir. Ayrıca R8 için en başarılı sonuçlar BoW modelde, TTM için Trigram modelde elde edilmiştir. SMTP metriğinin genellikle Entropi ve TF-IDF yöntemlerinde en etkili metrik olduğu gözlenmiştir. Bunun yanı sıra R8 verisi üzerinde öznitelik seçme işlemi uygulanmadan da sınıflandırma testleri yapılmış referans çalışmada elde edilen sonuçlarla örtüşen sonuçlar alınmıştır. Dolayısıyla R8 verisi için sunduğumuz sonuçlarda, öznitelik seçme işleminin sonuçları yaklaşık olarak yüzde [1,5-2] aralığında düşürmesinden dolayı fark oluştuğu tespit edilmiştir.

4.3. Türkçe Twitter Mesajları için LDA ile Duygu Sınıflandırması

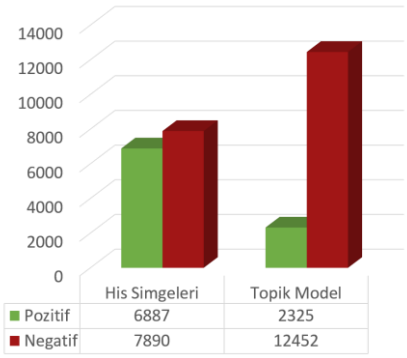
Bu çalışmada, TTM verisi üzerinde önceki çalışmamızda (Coban vd 2015a) kullanılan etiketleme (his simgelerini kullanarak) yönteminin yerine topik bilgisine dayalı etiketleme yönteminin kullanılması önerilmiştir. Böylece etiketleme (duygu sınıflandırması) işlemi ile sınıflandırma başarısının doğru orantılı olduğu düşünülerek DA başarısının artırılması amaçlanmıştır. Yöntem olarak ise yine bu çalışmada kullanılan önişleme teknikleri uygulanmış; önişlemden sonra veri seti topik bilgisine dayalı duygu sınıflandırmasından geçirilmiştir. Bu aşamada kullanılan topik model Şekil 4.4'te notasyonu verilen Gizli Dirichlet Ataması (LDA: Latent Dirichlet Allocation) yöntemiyle oluşturulmuştur (Blei *et al.* 2003).



Şekil 4.4. Gizli Dirichlet Ataması

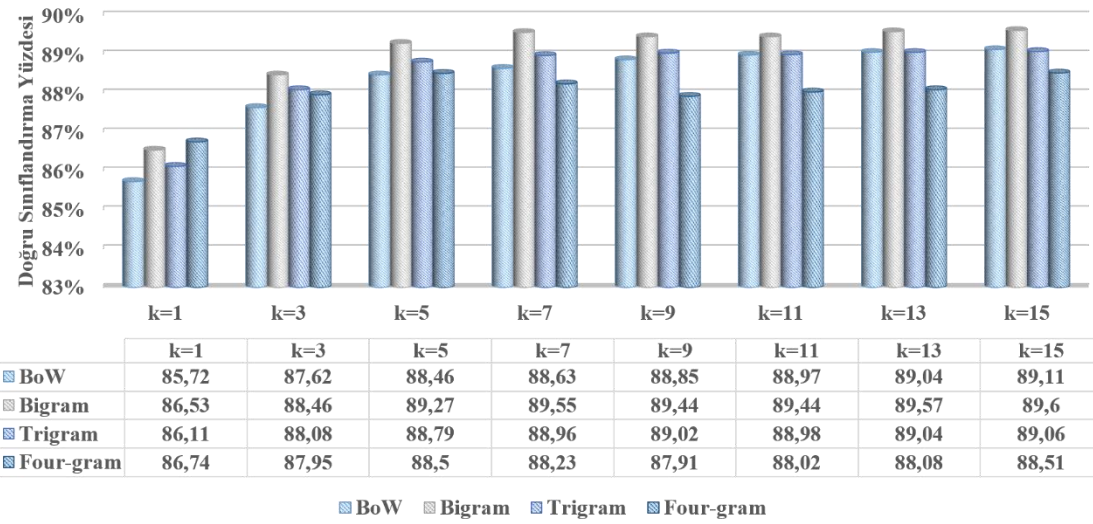
Gizli Dirichlet Ataması (GDA) yöntemi ise MALLET (Anonymous 2002) paketi ile uygulanmıştır. Duygu sınıflandırmasından sonra BoW ve karakter seviye Ngram (bigram, trigram ve four-gram) modellerde öznitelik çıkarılmıştır. TA aşamasında ise ANTF yöntemi kullanılmıştır. Sistemin eğitim ve test aşamasında MS ve DA alanlarında yaygın olarak kullanılan k-NN, SVM, NB, MNB ve ME sınıflandırıcıları kullanılmıştır. k-NN algortimasında en iyi k değerini belirleyebilmek adına k parametresinin [1-15] aralığındaki değerleri için sonuçlar elde edilmiştir. En yakın komşuların tespitinde ise Öklid uzaklığı kullanılmıştır.

Önişlemden sonra veri setinden 14777 mesaj elde edilmiştir. Duygu sınıflandırması aşamasında GDA yöntemi için ön tanımlı parametreler kullanılmış, topik ve iterasyon sayıları sırasıyla 2 ve 2000 olarak alınmıştır. Deneysel sonuçlar 10-kat çapraz geçerleme modeliyle elde edilmiş, performans değerlendirmesinde doğruluk metriği kullanılmıştır. Duygu sınıflandırmasından sonra TTM verisi için elde edilen kategori-örnek dağılımı ise Şekil 4.5'te verilmiştir. Bu durum etiketleme aşamasında his simgeleri dikkate alındığında negatif kategorili mesajların çoğunlukla pozitif (yanlış bir şekilde) olarak etiketlendiğini göstermiştir.



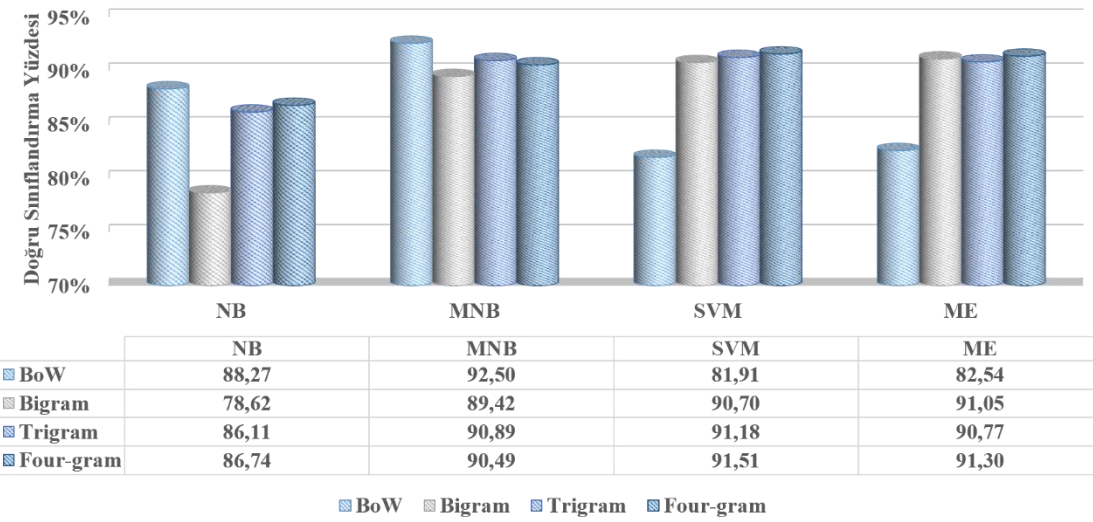
Şekil 4.5. Duygu sınıflandırmasından sonra TTM verisi için kategori-örnek dağılımı

Her iki öznitelik modeli için *k*-NN ve diğer sınıflandırıcılar ile elde edilen sonuçlar ise sırasıyla Şekil 4.6 ve Şekil 4.7'de verilmiştir.



	k=1	k=3	k=5	k=7	k=9	k=11	k=13	k=15
BoW	85,72	87,62	88,46	88,63	88,85	88,97	89,04	89,11
Bigram	86,53	88,46	89,27	89,55	89,44	89,44	89,57	89,6
Trigram	86,11	88,08	88,79	88,96	89,02	88,98	89,04	89,06
Four-gram	86,74	87,95	88,5	88,23	87,91	88,02	88,08	88,51

Şekil 4.6. BoW ve Ngram modelde *k*-NN için sınıflandırma başarıları



	NB	MNB	SVM	ME
BoW	88,27	92,50	81,91	82,54
Bigram	78,62	89,42	90,70	91,05
Trigram	86,11	90,89	91,18	90,77
Four-gram	86,74	90,49	91,51	91,30

Şekil 4.7. BoW ve Ngram modelde NB, MNB, SVM ve ME için sınıflandırma başarıları

Deneysel sonuçlara göre, *k*-NN için bigram (en başarılı sonuçlar k=15 için alınmıştır), diğer sınıflandırıcılar için ise BoW model daha başarılı olmuştur. Ngram modelde karakter düzeyi ile sınıflandırma başarısı doğru orantılı olmuştur. En başarılı sınıflandırıcı ise MNB olmuş ve DA başarısı en yüksek %92,50 olarak elde edilmiştir.

Çalışmamızın amacı daha etkin bir etiketleme yöntemi kullanarak DA başarısını artırmak olmuş ve önceki çalışmamıza kıyasla yaklaşık olarak %26 oranında daha iyi başarı elde edilmiştir. Bu artışın sebebi ise beklendiği gibi duygu sınıflandırmasında kullanılan yöntemden dolayı mesajların çok daha yüksek oranda doğru bir şekilde etiketlenmesi olmuştur. Topik bilgisine dayalı etiketleme yönteminin en önemli avantajı ise dil işleme ile ilgili işlemler uygulanmadan yüksek başarı elde edilebilmesi olmuştur. Elde edilen sonuçlar DA'da etiketleme işleminin sonuçlar üzerinde doğrudan etkili olduğunu doğrulamıştır. Bu nedenle topik bilgisinin yanı sıra dil işleme ile ilgili yöntemler de uygulanarak başarı oranını daha da artırmak mümkün olabilir. Ayrıca bu yöntem metin tabanlı farklı çalışma alanlarında iki veya daha fazla kategorili veri kümeleri üzerinde de oldukça başarılı sonuçlar üretebilir.

4.4. Metin Sınıflandırma Teknikleri ile İstenmeyen Kısa Mesajların Otomatik Olarak Tespit Edilmesi

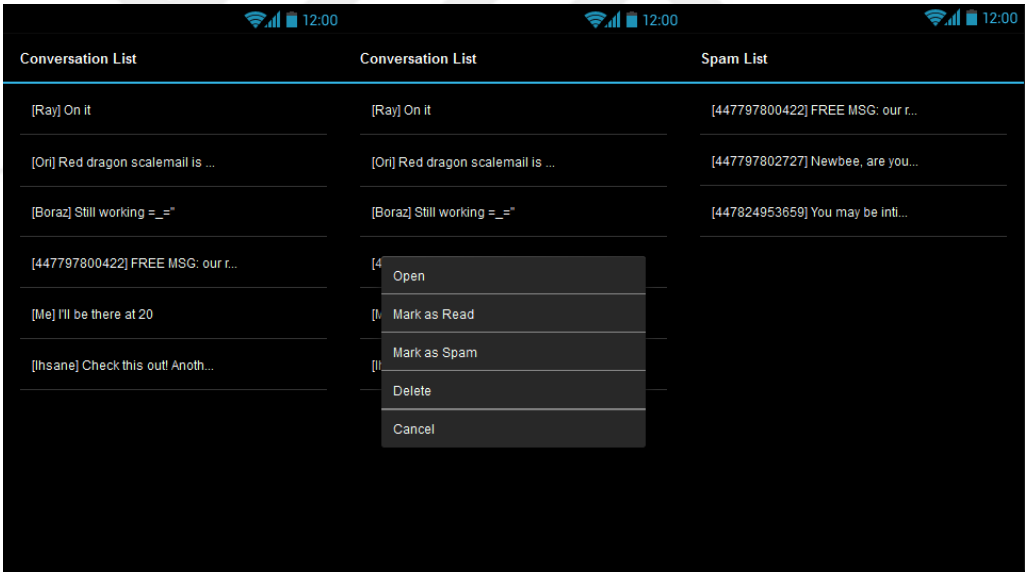
Bu çalışmada, istenmeyen kısa mesajların otomatik olarak tespit edilmesi ve filtrelenmesi amaçlı deneyler gerçekleştirilmiştir. Bunun yanı sıra istemci odaklı olması öngörülen ve MS tekniklerinin kullanıldığı (uzman sistem modelini temel alan) bir sistem önerilmiştir. Performans ve yeni kuralların eklenebilmesi gibi avantajlarından dolayı sistemin, uzman sistem tabanlı olması öngörülmüştür. Çalışmamızda önerilen sistem; uzman sistem ve otomatik sınıflandırma işlemi için gerekli eğitim aşaması olmak üzere iki adımdan oluşmaktadır.

Temel amaç kısa mesajları analiz edip kendini sürekli güncelleyerek (kullanıcıdan öğrenerek) otomatik olarak istenmeyen mesajları filtreleyebilecek mobil bir uygulamanın geliştirilmesi olmuştur. Bu bağlamda öncelikle tasarlanan uzman sistemin gelen mesajın spam olup olmadığına karar vermesi, bu kararı veremediği durumlarda ise mesajın MS teknikleri ile işlenerek önceden eğitilmiş bir sınıflandırıcı ile sınıflandırılması öngörülmüştür. Bu nedenlerden dolayı çalışmamızın ikinci aşamasında klasik MS teknikleri kullanılmış, sınıflandırıcı olarak ise performans ve uygulanabilirlik açısından avantajlı olan Bayes algoritması kullanılmıştır. Ancak en iyi performansı

gösteren sınıflandırıcıyı tespit etmek için NB, MNB, *k*-NN ve SVM algoritmaları ile de deneyler yapılmış ve karşılaştırmalar yapılmıştır. Deneyler sonuçlar elde edilirken sınıflandırıcıların eğitimi Android platformunda gerçekleştirilemediği için bilgisayar ortamında eğitim yapılmış ve sonuçlar elde edilmiştir. Önerilen sistemin temel iki adımı ise sırasıyla aşağıda kısaca açıklanmıştır.

4.4.1. Uzman sistem aşaması

Sistemimizin ilk aşaması olan uzman sistemde mobil cihaza gelen kısa mesajlar işlenmektedir. Bu adımda filtrelenemeyen mesajlar ise sınıflandırılmak üzere diğer aşamaya gönderilmektedir.

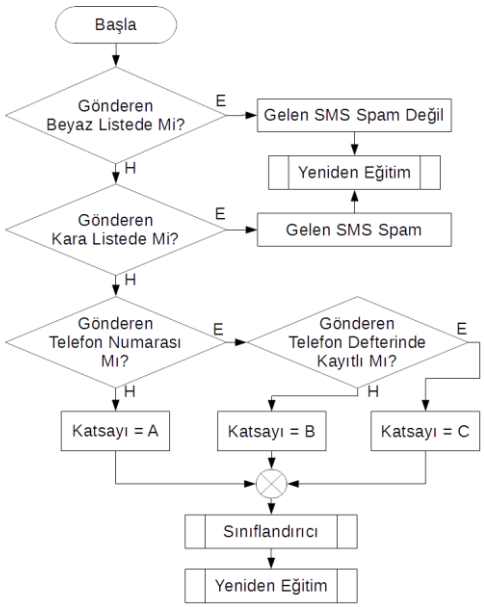


(a) (b) (c)

Şekil 4.8. Geliştirilen Android yazılımı; mesajlaşma ekranı (a), mesaj seçenekleri menüsü (b), spam mesaj kutusu (c)

Bu aşamadan sonra sınıflandırıcıdan mesajın spam olma ihtimali elde edilmekte ve mesajın filtrelenmesi ile ilgili karar son olarak uzman sistem bünyesinde verilmektedir. Şekil 4.8'de önerdiğimiz sistemi kullanan ve Android işletim sistemi için geliştirilen prototip uygulamanın ekran görüntüleri verilmiştir. Sistemin bu uygulama üzerindeki iç

işleyişinde (Şekil 4.9) mesaj kontrolü ise; uzman sistem tarafından tutulan ve sürekli güncellenen kara ve beyaz listelerde gönderen bilgisinin gözlenip gözlenmediğinin kontrolü ile başlamaktadır. Bu bilgi beyaz listede mevcut ise mesaj gelen kutusuna eklenir ve yeni mesaj bildirimi yapılır. Aksi durumda ise bilgi kara listede aranır ve eşleşme olması durumunda mesaj filtrelenir ve yeni mesaj bildirimi yapılmaz. Gönderen bilgisinin her iki listede de gözlenmemesi durumunda ise mesaj sınıflandırıcıya gönderilir. Son adımda, mesaj yine bu bilgiye bağlı olarak belirlediğimiz üç farklı kat sayı kullanılarak (sınıflandırıcıdan gelen spam olma ihtimali ile birlikte) uzman sistem tarafından kategorize edilir.



Şekil 4.9. Uzman sistem akış diyagramı

Kara ve beyaz listelerde eşleşme olmaması durumunda kullanılacak üç farklı oran ise sırasıyla 0,2 (gönderen rehberde kayıtlı), 0,7 (rehbere kayıtlı olmayan bir telefon numarası) ve 1,2 (gönderen telefon numarası değil) olacak şekilde belirlenmiştir. Ancak bu değerler istenmesi durumunda kullanıcı tarafından değiştirilebilmektedir.

4.4.2. Sınıflandırma aşaması

Sistemin bu aşamasında uzman sistem tarafından filtrelenemeyen mesajlar metinsel içeriği analiz edilerek sınıflandırılmakta ve geriye mesajın spam olma ihtimali döndürülmektedir. Klasik MS teknikleri ile işlenen mesajlar sırasıyla önişleme, terim ağırlıklandırma, öznitelik seçme ve sınıflandırma aşamalarından geçirilmektedir. Çalışmamızın bu adımında mesajların otomatik olarak sınıflandırılması için MS teknikleri kullanılmış ve en iyi performansı gösteren sınıflandırıcı ve öznitelik modelini belirleyebilmek için deneyler yapılmıştır. Sistemin eğitim ve test aşamalarında SpamSMSCollection veri seti ve Şekil 3.2'de verilen sistem modeli kullanılmıştır. Önişleme aşamasında her türlü anlamsız karakterden temizlenen mesaj içeriğinde tespit edilen his simgeleri özel karşılıkları ile kodlanarak korunmuştur. Daha sonra durak kelime çıkarımı ve gövdeleme işlemleri uygulanmıştır (Porter 1980). Bunun yanı sıra minimum terim frekansı ve terim uzunluğu filtreleri uygulanmış; frekans ve uzunluk için minimum değer 2 olarak alınmıştır.

TA aşamasında BINARY, TF, IDF ve TF-IDF yöntemleri kullanılmıştır. Daha sonra veri üzerinde ARFF dosya formatına dönüştürme işlemi uygulanmıştır. Öznitelik seçme işleminin sonuçlar üzerindeki etkisini incelemek amacıyla deneysel sonuçlar tüm öznitelikler ve seçilmiş öznitelikler (elde edilen benzersiz öznitelikler üzerinde KTÖS yöntemiyle) kullanılarak iki farklı şekilde elde edilmiştir. Bunun yanı sıra normalizasyon (N) işleminin etkisini inceleyebilmek amacıyla da global normalizasyon işlemi uygulanmıştır. Sınıflandırma esnasında literatürde MS çalışmalarında yaygın olarak kullanılan *k*-NN, NB, MNB ve SVM algoritmaları kullanılmıştır.

Çizelge 4.14. Uygulanan önişlemin veri seti üzerindeki etkisi

Özellik	**Önişlem**	
	Önce	Sonra
Ortalama Terim	11,3	10
Normal (ham)	4825	4765
İstenmeyen (spam)	749	745
Toplam Örnek	5574	5510

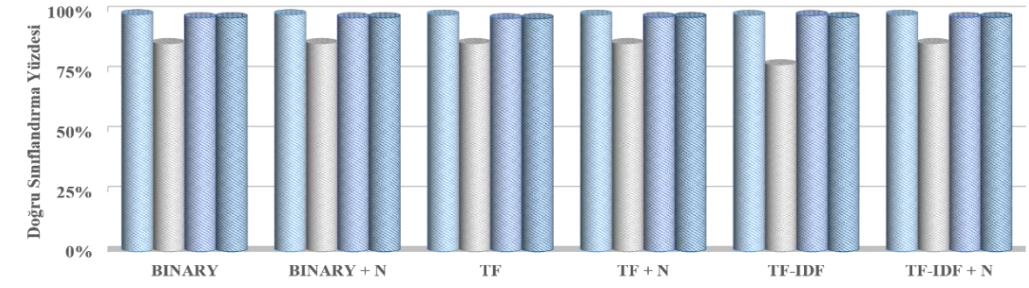
Çizelge 4.15. Önişlemden sonra elde edilen öznitelik istatistikleri

Elenen Öznitelik	**Sayısı**	**Yüzdesi (%)**
Yok (Tüm Terimler)	63445	100
His Simgeleri	354	0,56
(-) Filtrelenen	4075	6,42
Toplam	**4075**	**6,42**

Veri seti önişlemden geçirildikten sonra örnek ve ortalama terim sayılarındaki değişim ile öznitelik istatistikleri sırasıyla Çizelge 4.14 ve Çizelge 4.15'te verilmiştir. Uygulanan minimum terim frekansı filtresinden dolayı önişlemden sonra veri setinden 64 mesaj çıkarılmıştır. Öznitelikler incelendiğinde ise minimum terim uzunluğu filtresi, gövdeleme ve durak kelimelerin çıkarılması gibi nedenlerden dolayı öznitelik sayısının %6,42 oranında azaldığı tespit edilmiştir. Öznitelik çıkarım aşamasında model olarak BoW modeli kullanılmış ve önişlemden sonra tüm veri setinden 6622 adet benzersiz öznitelik elde edilmiştir. Sınıflandırma sonuçları öznitelik seçme işleminin uygulanma durumuna bağlı olmak üzere iki başlık altında incelenmiştir. Sistemin bu adımıyla ilgili deneyler OMESIS yazılımı kullanılarak gerçekleştirilmiştir.

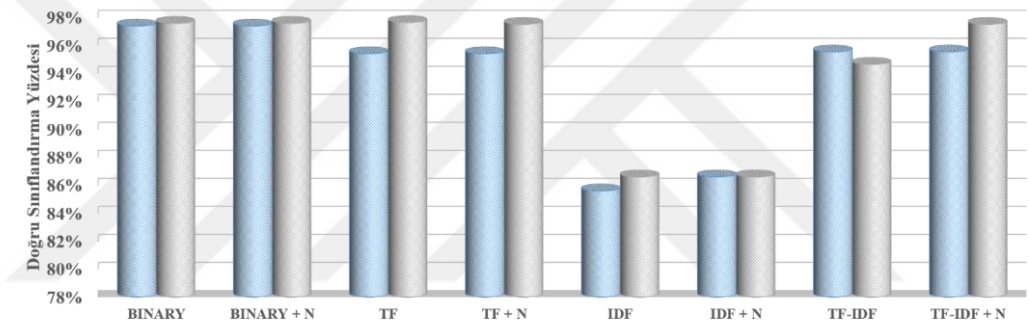
4.4.3. Öznitelik seçme işlemi uygulanmadan elde edilen sonuçlar

Bu başlık altında sınıflandırma algoritması ve ağırlıklandırma yönteminin farklı olduğu durumlar ile normalizasyon işleminin kombinasyonu için elde edilen sonuçlar verilmiştir. Sonuçlar öznitelik seçme işlemi uygulanmadan elde edilmiştir. Şekil 4.10'da SVM, Şekil 4.11'de NB ve MNB algoritmaları için dört farklı ağırlıklandırma yöntemi ile elde edilen sonuçlar verilmiştir. Şekil 4.12'de *k*-NN algoritması ile elde edilen sonuçlarda ise en başarılı sonucu verdiği için sadece TF-IDF yöntemi için elde edilen sonuçlar verilmiştir. Ayrıca tüm sonuçlara, normalizasyon işlemi uygulanarak elde edilen sonuçlar da eklenmiştir.



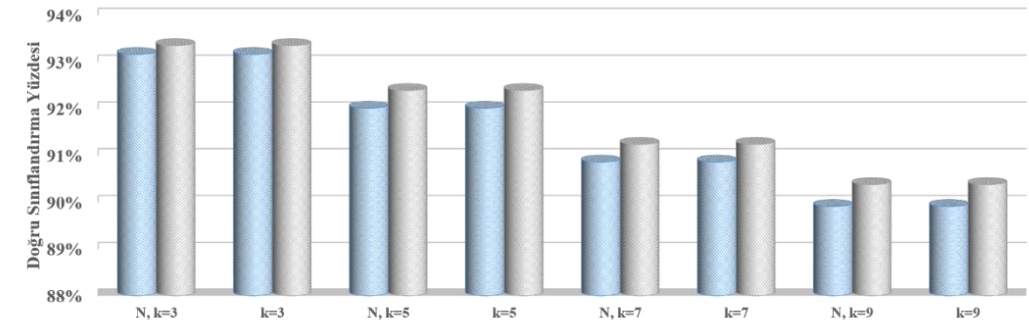
	BINARY	BINARY + N	TF	TF + N	TF-IDF	TF-IDF + N
Linear	98,61	98,61	98,43	98,42	98,34	98,42
Polynomial	86,59	86,59	86,59	86,59	77,79	86,59
RBF	97,32	97,32	96,96	97,46	98,24	97,43
Sigmoid	97,32	97,32	96,93	97,46	97,45	97,45

Şekil 4.10. Tüm öznitelikler kullanıldığında SVM ile elde edilen sonuçlar



	BINARY	BINARY + N	TF	TF + N	IDF	IDF + N	TF-IDF	TF-IDF + N
NB	97,36	97,36	95,38	95,38	85,59	86,59	95,51	95,51
MNB	97,55	97,55	97,61	97,48	86,59	86,59	94,63	97,48

Şekil 4.11. Tüm öznitelikler kullanıldığında NB ve MNB ile elde edilen sonuçlar



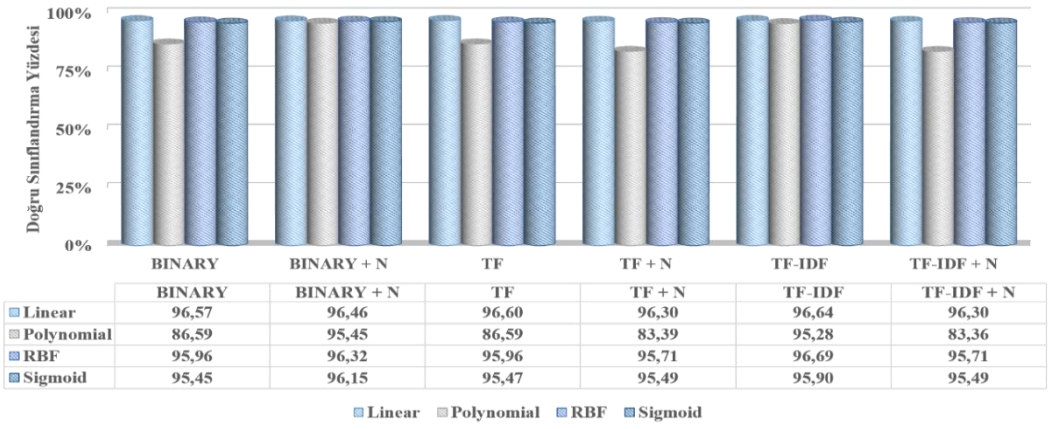
	N, k=3	k=3	N, k=5	k=5	N, k=7	k=7	N, k=9	k=9
Euclidean	93,16	93,16	92,01	92,01	90,86	90,86	89,91	89,91
Manhattan	93,35	93,35	92,39	92,39	91,24	91,24	90,38	90,38

Şekil 4.12. Tüm öznitelikler kullanıldığında *k*-NN ile elde edilen sonuçlar

Yukarıdaki sonuçlardan; uygulanan yöntemlerle veri seti üzerinde oldukça başarılı sonuçlar elde edildiği, verinin çok seyrek olmasından dolayı uygulanan farklı ağırlıklandırma yöntemi ve normalizasyon işleminin bu veri seti için sonuçlar üzerinde çok büyük bir farklılığa neden olmadığı tespit edilmiştir. En iyi sonuçlar ise SVM algoritmasıyla doğrusal çekirdek kullanıldığında elde edilmiştir. Ayrıca bu veriler ışığında veri setindeki örneklerin ayırt edilebilirlik özelliğinin oldukça iyi olduğu ve kategori sayısının iki (ham, spam) olmasının da sınıflandırma başarısına olumlu katkı yaptığı düşünülmektedir, zira kategori sayısı arttıkça sınıflandırıcının yanlış atama yapma olasılığı da artacaktır. Yukarıdaki sonuçlara ek olarak öznitelik seçme işleminin sonuçlara etkisini incelemek amacıyla KTÖS algoritmasıyla öznitelikler seçilmiş ve sınıflandırma yapılmıştır.

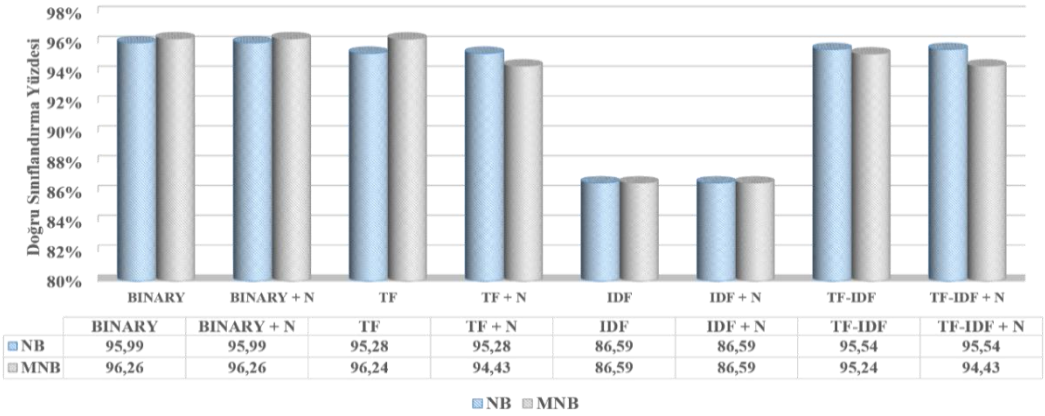
4.4.4. Öznitelik seçme işlemi uygulanarak elde edilen sonuçlar

Bu başlık altında öznitelik seçme işlemi uygulandıktan sonra elde edilen sınıflandırma sonuçları (Şekil 4.13, Şekil 4.14 ve Şekil 4.15) verilmiştir. Uygulanan normalizasyon işleminin ise seçilen öznitelik sayısına etkisi olmadığı tespit edilmiştir.



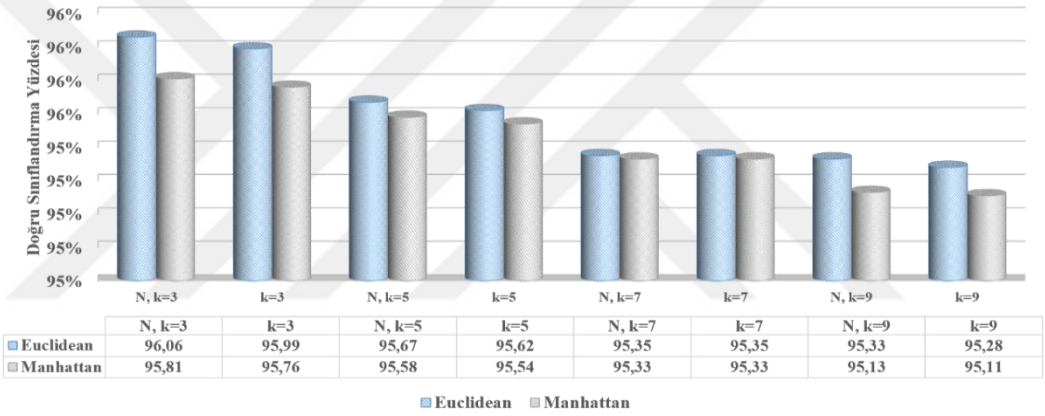
	BINARY	BINARY + N	TF	TF + N	TF-IDF	TF-IDF + N
Linear	96,57	96,46	96,60	96,30	96,64	96,30
Polynomial	86,59	95,45	86,59	83,39	95,28	83,36
RBF	95,96	96,32	95,96	95,71	96,69	95,71
Sigmoid	95,45	96,15	95,47	95,49	95,90	95,49

Şekil 4.13. Seçilmiş öznitelik kullanıldığında SVM ile elde edilen sonuçlar



	BINARY	BINARY + N	TF	TF + N	IDF	IDF + N	TF-IDF	TF-IDF + N
NB	95,99	95,99	95,28	95,28	86,59	86,59	95,54	95,54
MNB	96,26	96,26	96,24	94,43	86,59	86,59	95,24	94,43

Şekil 4.14. Seçilmiş öznitelik kullanıldığında NB ve MNB ile elde edilen sonuçlar



	N, k=3	k=3	N, k=5	k=5	N, k=7	k=7	N, k=9	k=9
Euclidean	96,06	95,99	95,67	95,62	95,35	95,35	95,33	95,28
Manhattan	95,81	95,76	95,58	95,54	95,33	95,33	95,13	95,11

Şekil 4.15. Seçilmiş öznitelik kullanıldığında *k*-NN ile elde edilen sonuçlar

Tüm bu bilgiler ışığında, yapılan deneylerde SVM sınıflandırıcının en başarılı sınıflandırıcı olduğu ve en iyi sonuçların doğrusal çekirdek ile elde edildiği tespit edilmiştir. Öznitelik seçme işleminin SVM, NB ve MNB sınıflandırıcıları için başarıyı düşürdüğü ancak *k*-NN sınıflandırıcısı için yaklaşık %3 oranında artırdığı tespit edilmiştir. En başarılı sonuçlar ise tüm öznitelikler kullanıldığında elde edilmiştir. Deneysel sonuçlar incelendiğinde, çalışmamızda uygulanan önişlemlerden dolayı daha yüksek başarı (veri setinin alındığı çalışma ile kıyaslandığında) ile istenmeyen mesajların tespit edildiği görülmüştür.

4.5. Türkçe Şarkı Sözlerinden Müzik Türü Sınıflandırması

Bu çalışmada, bir metin sınıflandırma problemi olarak ele alınan müzik türü sınıflandırması Türkçe şarkı sözleri üzerinde uygulanmıştır. Bu amaçla bu alanda yaygın olarak kullanılan SSTF modelin yanı sıra BoW ve Ngram model ile öznitelik elde edilmiştir. TA aşamasında farklı ağırlıklandırma yöntemleri (ANTF, STF, TF, TF-IDF) kullanılmıştır. Önişleme aşamasında tüm öznitelik modellerinde ortak olarak metin içeriği üzerinde küçük harf dönüşümü uygulanmış ve öznitelik olarak kabul edilmeyen her türlü karakter temizlenmiştir. Ayrıca segmentler arasında tek satır boşluk olması sağlanmış ve nakaratlar tek segmente indirgenmemiştir. Ngram ve BoW modelde terim uzunluğu ve terim frekans filtreleri uygulanmamıştır. Her üç modelde anlamsal işlemler Zemberek ile gerçekleştirilmiştir. SSTF modelde ihtiyaç duyulan durak kelime oranı için ise Şekil 3.3'te verilen Türkçe durak kelime listesi kullanılmıştır. Ngram modelde öznitelikler karakter düzeyinde olmak üzere üç farklı şekilde (bigram, trigram ve four-gram) elde edilmiştir.

SSTF modelde bulunan toplam 42 adet öznitelik ise üç gruba (SSTFI, SSTFII ve SSTFIII) ayrılmıştır. Bu öznitelikler yapısal (SSTFI), istatistiksel (SSTFII) ve sözcük türü etiketleri (SSTFIII) özniteliklerini içermektedir. SSTFI ve SSTFII gruplarının içerdiği öznitelikler Çizelge 4.16'da verilmiştir. SSTFIII grubunda ise edat, isim, sıfat, zamir, bağlaç, fiil + sıfat, fiil + isim ve isim + sıfat frekansları öznitelik olarak kullanılmıştır. SSTFII grubuna uyak şablon (AA, ABB, ABAB, ABBA) öznitelikleri de (SSTFU) eklenmiştir. Ayrıca her üç öznitelik modelinde, en iyi öznitelik grubu ile ağırlıklandırma yöntemi kombinasyonunun da sonuçlar üzerindeki etkisi incelenmiştir. Türkçe'de kelimeler yazıldığı gibi seslendirildiği için çalışmamızda uyak tespitinde hece benzerliği yeterli koşul kabul edilmiştir. Uyak tespitinde aralarında ses benzerliği bulunan kelimelerin redif ya da uyak olup olmadığı kontrol edilmemiş (redifler de kafiye kabul edilmiştir), yarım kafiye (tek ses benzerliği) dikkate alınmamıştır.

Çizelge 4.16. SSTFI ve SSTFII grubu öznitelikleri

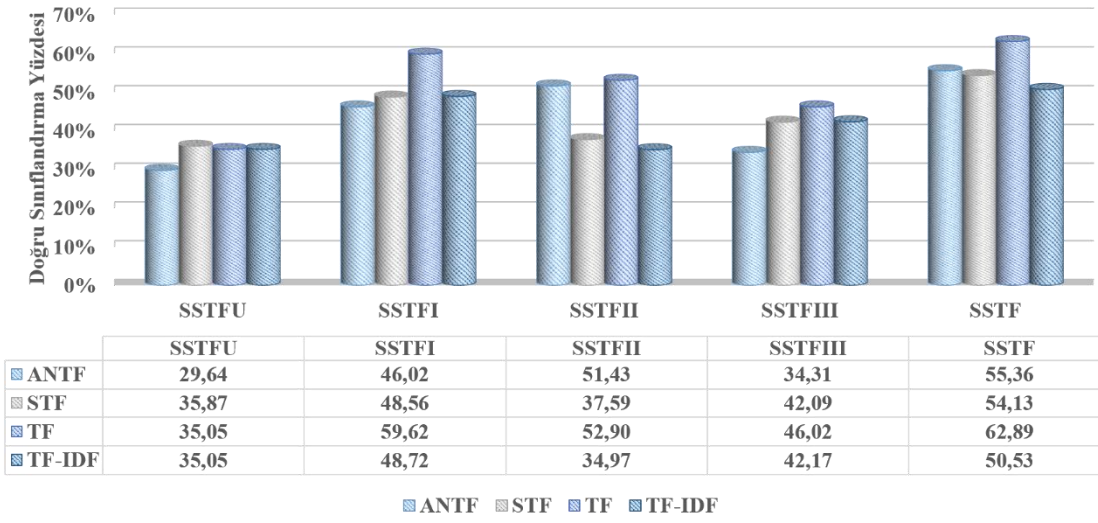
	Öznitelikler			
SSTFI	Cümle Başına Düşen Kelime Oranı	Durak Kelime Oranı	Ortalama Satır Uzunluğu	Segment Başına Düşen Satır Sayısı
	Çapraz Uyak Sayısı (ABAB | BABA)	Toplam Satır Sayısı	Ortalama Kelime Uzunluğu	Düz Uyak Sayısı (AA | BB)
	Segment Başına Düşen Ortalama Satır	Boş Satır Sayısı	Ortalama Cümle Uzunluğu	Sarmal Uyak Sayısı (ABBA | BAAB)
	İkili Düz Uyak Sayısı (AABB | BBAA)	Segment Sayısı	Cümle Sayısı	Benzersiz Uyak Kelimeleri Sayısı
SSTFII	Kelime Başına Düşen Hece Oranı	Kelime Sayısı	Satır Başına Düşen Kelime Oranı	(-) Tire İşareti Sayısı
	Kelime Başına Düşen Karakter Oranı	Sayısal Değer Sayısı	(“) Çift Tırnak Sayısı	(…) Üç Nokta Sayısı
	Kelime Başına Düşen Karakter Varyansı	Kelime Zenginliği Oranı	(:) İki Nokta Sayısı	(!) Ünlem İşareti Sayısı
	Satır Başına Düşen Benzersiz Kelime Oranı	Benzersiz Kelime Sayısı	(;) Noktalı Virgül Sayısı	(,) Virgül Sayısı
	Satır Başına Düşen Kelime Varyansı	Noktalama İşaretleri Oranı	(?) Soru İşareti Sayısı	(*) Yıldız Sayısı

BoW modelde durak kelime çıkarımı ve kök bulma işlemlerinin etkisini inceleyebilmek için öznitelikler üç farklı şekilde elde edilmiştir. Her modelde en iyi öznitelik grubu alınarak birleştirilmiş (birleşik model) ve sonuçlar üzerindeki etkisi incelenmiştir. Birleştirilmiş modelde öznitelikler tekrar ağırlıklandırılmamıştır. Deneylerde kullanılan öznitelik grupları ve kodları ise Çizelge 4.17’de verilmiştir. Sınıflandırma aşamasında 10-kat çapraz geçerleme modeli ve SVM algortiması kullanılmıştır. Sınıflandırıcının performans değerlendirmesinde ise doğruluk metriği kullanılmıştır.

Çizelge 4.17. Deneylerde kullanılan öznitelikler ve kodları

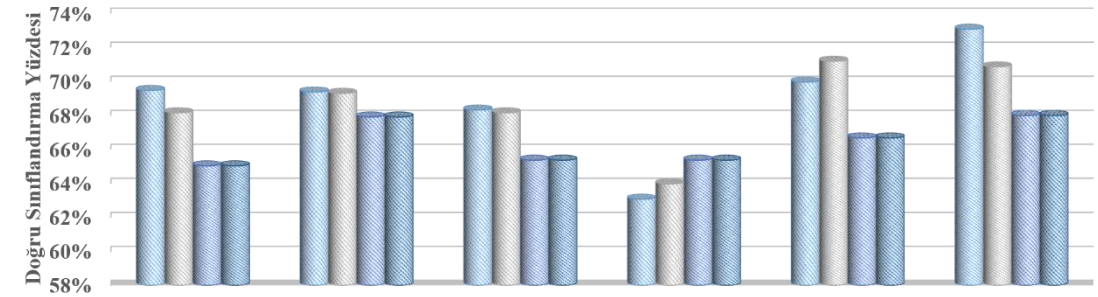
Öznitelik Grubu / Modeli		**Kodu**
SSTF	Yapısal Öznitelikler	SSTFI
	İstatistikler Öznitelikler	SSTFII
	Sözcük Türü Frekansları (POS Tags)	SSTFIII
	Uyak Şablonları (AA, AABB, ABAB, ABBA)	SSTFU
	SSTFI + SSTFII + SSTFIII	SSTF
BoW	BoW + Kök Bulma	BoWS
	BoW + Durak Kelime Çıkarma	BoWRS
	BoW + Kök Bulma + Durak Kelime Çıkarma	BoW
Ngram	Bigram	NGB
	Trigram	NGT
	Four-gram	NGF
Birleşik	BoW + Ngram	BWNG
	SSTF + BoW	SSBW
	Ngram + SSTF	NGSS

Farklı öznitelik grubu ve ağırlıklandırma yöntemleri ile elde edilen deneysel sonuçlar ise sırasıyla Şekil 4.16 ve Şekil 4.17'de verilmiştir.



	SSTFU	SSTFI	SSTFII	SSTFIII	SSTF
ANTF	29,64	46,02	51,43	34,31	55,36
STF	35,87	48,56	37,59	42,09	54,13
TF	35,05	59,62	52,90	46,02	62,89
TF-IDF	35,05	48,72	34,97	42,17	50,53

Şekil 4.16. SSTF öznitelikleri için farklı ağırlıklandırma yöntemleri ile elde edilen sınıflandırma başarıları (%)



	BoWS	BoWRS	BoW	NGB	NGT	NGF
ANTF	69,45	69,36	68,30	63,06	69,94	73,05
STF	68,14	69,28	68,14	63,96	71,17	70,84
TF	65,02	67,89	65,35	65,35	66,66	67,97
TF-IDF	65,02	67,89	65,35	65,35	66,66	67,97

Şekil 4.17. BoW ve Ngram öznitelikleri için farklı ağırlıklandırma yöntemleri ile elde edilen sınıflandırma başarıları

Deneysel sonuçlara göre, SSTF modelde en başarılı grup SSTFI olmuştur. SSTFU öznitelikleri ise tek başına kullanıldığında yeterli başarıyı sağlayamamıştır. Bu modelde en yüksek başarı ise tüm öznitelikler (SSTF) kullanıldığında elde edilmiştir. BoW modelde kök bulma ve durak kelimelerin çıkarılması işlemleri birlikte uygulandığında genellikle sınıflandırma başarısı düşmüştür.

Çizelge 4.18. Birleştirilmiş öznitelik grupları ile elde edilen sınıflandırma başarıları (%)

Öznitelik Grubu	**Sınıflandırma Başarısı**
SSTF + BoW	69,45
BoW + Ngram	**73,79**
Ngram + SSTF	73,62

SSTF modelde TF, BoW modelde ise ANTF en başarılı ağırlıklandırma yöntemi olmuştur. BoW modelde en iyi sonuçlar ise sadece kök bulma işlemi uygulandığında elde edilmiştir. Ngram modelde karakter seviyesi arttıkça sınıflandırma başarısı artmış, karakter düzeyine göre en başarılı ağırlıklandırma yöntemi değişmiştir. Öznitelik modeli bazında değerlendirildiğinde ise en başarılı model Ngram model olmuştur. Bunun yanı sıra Çizelge 4.18'de verilen sonuçlar incelendiğinde birleştirilmiş öznitelik

modelinin sınıflandırma başarısını artırdığı tespit edilmiştir. Tüm deneylerde elde edilen en yüksek sınıflandırma başarısı ise birleşik modelde %73,79 olmuştur.

Çalışmamızda klasik MS teknikleri ile Türkçe şarkı sözleri üzerinde otomatik müzik türü sınıflandırması çalışılmış ve bu alanda İngilizce için yapılan çalışmalardaki başarı yakalanmıştır. SSTF model diğer modellerden daha başarısız olsa da öznitelik sayısı bakımından performans avantajına sahiptir. Bu nedenle SSTF modelden vazgeçmek yerine gelecek çalışmalarda bu modelde kullanılan öznitelik kümesine yeni özniteliklerin eklenmesi düşünülmektedir. Ayrıca bu çalışmada uyak tespitinde kullanılan algoritmanın tüm Türkçe uyak şablonlarını tespit edecek şekilde geliştirilmesi düşünülmektedir.

5. SONUÇ

Bu tezde MS ve DA teknikleri ile sosyal medya ortamında otomatik olarak duygu tespiti konusu çalışılmıştır. Bu amaçla birbiriyle bağlantılı olan her iki çalışma alanında yapılan çalışmalar incelenmiştir. Literatür incelemesi sonucunda özellikle DA alanında Türkçe için yapılan çalışmaların yetersiz olduğu görülmüştür. Bu nedenle bu alandaki açığı kapatabilmek adına sosyal medyadan elde edilen Twitter gönderileri üzerinde DA çalışılmıştır.

Çalışmamızda literatürde çoğunlukla İngilizce veriler üzerinde uygulanan metot ve yöntemlerin yanı sıra Türkçe'ye özgü işlemler gerektiren teknikler uygulanmıştır. Hem MS hem de DA çalışmalarında kullanılan veri, metin tabanlı olduğu için bu tezde DA deneylerinin yanı sıra MS alanında da deneyler yapılmıştır. Ayrıca MS ve DA süreçlerini otomatikleştirme adına OMESIS yazılımı geliştirilmiştir. Geliştirilen yazılımın, Türkçe ve İngilizce veriler üzerinde DDİ ve MÖ yöntemleri kullanılarak metin tabanlı herhangi bir sınıflandırma çalışmasında kullanılabilmesi sağlanmıştır. Bu tez ile her iki alanda da gerçekleştirilen deneyler ulusal ve uluslararası konferanslarda bildiri olarak sunulmuştur.

Bu tez kapsamında; MS alanında istenmeyen mesajların otomatik olarak tespit edilmesi ve şarkı sözünden müzik türü tanıma konularında deneyler yapılmıştır. Ancak ağırlıklı olarak Twitter ortamından elde edilen Türkçe mesajlar üzerinde DA deneyleri gerçekleştirilmiştir. Genel olarak ise bu çalışmada metin tabanlı verilerin otomatik olarak sınıflandırılması sürecinde önişleme ve boyut indirgeme aşamalarına odaklanılmıştır. Ancak birincil hedef Türkçe Twitter mesajları için DA'nın gerçekleştirilmesi ve başarı oranının artırılması olmuştur. Bunun yanı sıra Twitter DA için *k*-NN algoritması ile kullanılan uzaklık ve benzerlik fonksiyonlarının genişletilmiş bir performans karşılaştırması gerçekleştirilmiştir.

Bu tez kapsamında gerçekleştirilen deneylerden elde edilen sonuçların kısa özeti ise aşağıda verilen genelleme ve çıkarımlar ile MS ve DA alanlarında çalışan araştırmacıların bilgisine sunulmuştur:

- MS ve DA süreç bakımından diğer örüntü tanıma problemleri ile benzerdir. Aradaki fark işlenen verinin metin tabanlı olması ve önişleme aşamasında uygulanması gereken tekniklerin farklı olmasından ileri gelmektedir.
- MS ile kıyaslandığında DA, karakter sınırı ve kullanıcıların gramer kurallarına uymadan mesaj yazıyor olmaları gibi nedenlerden dolayı daha etkin tekniklerin uygulanmasını gerektirmektedir ve bu durum anlamlı bilgi çıkarmayı zorlaştırmaktadır.
- Her iki çalışma alanında da elde edilen başarı önişleme tekniklerinin başarısıyla doğru orantılıdır.
- MS ve DA çalışmalarında en büyük problemlerden birisi öznitelik uzay boyutunun seyrek (sparse) yapıda olması ve çok yüksek boyutlara ulaşmasıdır. Bu nedenle etkili bir boyut indirgeme yöntemine genellikle ihtiyaç duyulmaktadır.
- Sınıflandırma başarısını etkileyen en önemli faktörlerden birisi de metinsel içerikten öznitelik elde edilirken kullanılan öznitelik modelidir.
- Ngram öznitelik modeli karakter ve kelime seviye olarak uygulanmakta ve genellikle karakter seviye daha başarılı olmaktadır. Karakter düzeyinde ise karakter sayısı ile sınıflandırma başarısı doğru orantılı olmaktadır.
- BoW model metnin diline bağlı önişleme teknikleri gerektirdiğinden bu modelde doğru ve anlamlı özniteliklerin elde edilmesi çok önemlidir.
- SSTF model yapılandırılmış veya yarı-yapılandırılmış metinlerden öznitelik çıkarmak için kullanılmakta ve öznitelik sayısı az olduğundan performans bakımından avantajlı olmaktadır.
- İçerik bakımından uzun olan metinler üzerinde genellikle BoW model daha başarılıyken, kısa metinler üzerinde Ngram model daha başarılı olmaktadır.
- Öznitelik modelleri arasında elde edilen öznitelik sayısı bakımından kıyaslama yapıldığında ise genellikle SSTF < BoW < Ngram sıralaması (karakter seviye

Ngram modelde bigram ve trigram öznitelikleri kullanıldığında öznitelik sayısı genellikle BoW modelden daha az olur) elde edilmektedir.

- Sınıflandırma yöntemleri arasında genellikle en başarılı yöntem SVM olmaktadır. Hız bakımından değerlendirildiğinde ise en hızlı yöntem MNB ve en yavaş yöntem *k*-NN algoritması olmaktadır.
- Araştırma bulguları bölümünde değinilmemiş olsa da henüz yayınlanmamış olan "Twitter Duygu Analizinde Terim Ağırlıklandırma Yönteminin Etkisi" başlıklı çalışmamızdan elde edilen bulgulara göre; TA aşamasında, TF ve TF-IDF gibi geleneksel yöntemlerin aksine sınıf bilgisini de dikkate alan ağırlıklandırma yöntemleri daha başarılı olmaktadır.
- Bu tezde uygulanan boyut indirgeme yöntemlerinden birisi olan KTÖS yöntemi öznitelikleri seçerken, seçilecek öznitelik sayısı ile ilgili kullanıcı müdahalesi gerektirmez. Bu nedenle deneylerde boyut indirgeme aşamasında KTÖS yöntemi tercih edilmiş ancak genellikle sınıflandırma başarısını düşürmüştür.

Yukarıda verilen genellemeler özetlenecek olursa sonuç olarak; Türkçe için MS ve DA alanlarında yapılan çalışma sayısı oldukça sınırlıdır. Ayrıca bu alanlarda kullanılabilecek herkese açık Türkçe veri setlerinin olmayışı en büyük problemlerden birisidir. Bu tez kapsamında, oluşturulan Twitter veri kümesi üzerinde iki kategorili DA çalışılmış ve en yüksek %92,50 oranında başarı sağlanmıştır. Ancak bu tezde uygulanan yöntemler çok kategorili DA problemi için de uygulanabilir. Ayrıca oldukça popüler çalışma alanlarından birisi olan DA problemine derin öğrenme (DL: Deep Learning) teknikleri de uygulanarak başarı oranı artırılabilir. Bunun yanı sıra hem DA hem de MS alanlarında öznitelik boyutunun çok yüksek olması performans bakımından aşılması gereken önemli engellerden birisidir. Bu nedenle MS ve DA problemleri üzerinde paralelleştirme veya dağıtık hesaplama gibi teknikler uygulanarak performans problemi ortadan kaldırılabilir.

KAYNAKLAR

Adalı, E., 2012. Doğal Dil İşleme (Natural Language Processing). Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 6(6).

Afrin, F., Nahar, I., 2015. Incremental learning based intelligent job search system (Doctoral dissertation, BRAC University).

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R., 2011, June. Sentiment analysis of twitter data. In Proceedings of the Workshop on Languages in Social Media (pp. 30-38). Association for Computational Linguistics.

Akın, A. A., Akın, M. D., 2007. Zemberek, an open source NLP framework for Turkic Languages. Structure, 10, 1-5.

Almeida, T. A., Hidalgo, J. M. G., Yamakami, A., 2011. Contributions to the study of SMS spam filtering: new collection and results. In Proceedings of the 11th ACM symposium on Document engineering (pp. 259-262). ACM.

Alpaydin, E., 2014. Introduction to machine learning. MIT press.

Amasyalı, M. F., Balcı, S., Mete, E., Varlı, E. N., 2012. Türkçe Metinlerin Sınıflandırılmasında Metin Temsil Yöntemlerinin Performans Karşılaştırılması/A Comparison of Text Representation Methods for Turkish Text Classification. EMO Bilimsel Dergi, 2(4).

Anonymous, 1987a. http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html (20/07/2015)

Anonymous, 1987b. http://web.ist.utl.pt/~acardoso/datasets/ (20/07/2015)

Anonymous, 2001. http://www.cs.cmu.edu/~webkb/ (20/07/2015)

Anonymous, 2002. http://mallet.cs.umass.edu/topics.php (08/08/2015)

Anonymous, 2006. http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection (25/07/2015)

Anonymous, 2007. https://code.google.com/p/zemberek/ (30/07/2015)

Anonymous, 2009. http://www.cs.waikato.ac.nz/ml/weka/ (01/08/2015)

Anonymous, 2010. https://opennlp.apache.org/ (03/08/2015)

Anonymous, 2011a, http://analyticstraining.com/2011/sentiment-analysis/ (20/07/2015)

Anonymous, 2011b. http://lucene.apache.org/ (24/07/2015)

Anonymous, 2011c. https://www.csie.ntu.edu.tw/~cjlin/libsvm/ (02/08/2015)

Anonymous, 2015a. https://en.wikipedia.org/wiki/Twitter (23/07/2015)

Anonymous, 2015b. https://apiwiki.twitter.com/ (23/07/2015)

Anonymous, 2015c. https://dev.twitter.com/overview/documentation (23/07/2015)

Argamon, S., Whitelaw, C., Chase, P., Hota, S. R., Garg, N., Levitan, S., 2007. Stylistic text classification using functional lexical features. Journal of the American Society for Information Science and Technology, 58(6), 802-822.

Aslam, J. A., Frost, M., 2003. An information-theoretic measure for document similarity. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (pp. 449-450). ACM.

Ayodele, T. O., 2010. Machine learning overview. INTECH Open Access Publisher.

Baharudin, B., Lee, L. H., Khan, K., 2010. A review of machine learning algorithms for text-documents classification. Journal of advances in information technology, 1(1), 4-20.

Baker, K., 2005. Singular value decomposition tutorial. The Ohio State University, 2005, 1-24.

Blei D. M., Ng A. Y., and Jordan M. I., 2003. Latent Dirichlet allocation. the Journal of machine Learning research, 3, 993-1022.

Boiy, E., Hens, P., Deschacht, K., Moens, M. F., 2007. June. Automatic sentiment analysis in on-line text. In ELPUB (pp. 349-360).

Bollen, J., Mao, H., Zeng, X., 2011. Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1-8.

Bozan, Y. S., Coban, O., Ozyer, G. T., Ozyer, B., 2015. SMS spam filtering based on text classification and expert system. In Signal Processing and Communications Applications Conference (SIU), 2015 23th (pp. 2345-2348). IEEE.

Burges, C. J., 1998. A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2), 121-167.

Brücher, H., Knolmayer, G., Mittermayer, M. A., 2002. Document classification methods for organizing explicit knowledge. Institut für Wirtschaftsinformatik der Universität Bern.

Caruana, R., Niculescu-Mizil, A., 2006. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning (pp. 161-168). ACM.

Cavnar, W. B., Trenkle, J. M., 1994. N-gram-based text categorization. Ann Arbor MI, 48113(2), 161-175.

Chang, C. C., Lin, C. J., 2011. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3), 27.

Chao, W. L., 2011. Machine Learning Tutorial.

Chowdhury, G. G., 2003. Natural language processing. Annual review of information science and technology, 37(1), 51-89.

Coban, O., Ozyer, B., Ozyer, G. T., 2015. Sentiment analysis for Turkish Twitter feeds. In Signal Processing and Communications Applications Conference (SIU), 2015 23th (pp. 2388-2391). IEEE.

Coban, O., Ozyer, B., Ozyer, G. T. A Comparison of Similarity Metrics for Sentiment Analysis on Turkish Twitter Feeds. In International Conference on Smart City/SocialCom/SustainCom (SmartCity), 2015 (pp. 333-338). IEEE.

Coban, O., Ozyer, G. T., 2016. Sentiment classification for Turkish Twitter feeds using LDA. In Signal Processing and Communications Applications Conference (SIU), 2016 24th. IEEE.

Coban, O., Ozyer, G. T., 2016. Music genre classification from Turkish Lyrics. In Signal Processing and Communications Applications Conference (SIU), 2016 24th. IEEE.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), 273-297.

Chisholm, E., Kolda, T. G., 1999. New term weighting formulas for the vector space method in information retrieval. Computer Science and Mathematics Division, Oak Ridge National Laboratory.

Cuong, N. V., Ha, N. T. T. L., Thuy, Q., Hieu, P. X., 2006. A Maximum Entropy Model for Text Classification. In The International Conference on Internet Information Retrieval 2006 (pp. 134-139).

Dadvar, M., Hauff, C., De Jong, F. M. G., 2011. Scope of negation detection in sentiment analysis.

Dalal, M. K., Zaveri, M. A., 2011. Automatic text classification: a technical review. International Journal of Computer Applications, 28(2), 37-40.

Darroch, J. N., Ratcliff, D., 1972. Generalized iterative scaling for log-linear models. The annals of mathematical statistics, 1470-1480.

Dasgupta, A., Drineas, P., Harb, B., Josifovski, V., Mahoney, M. W., 2007. Feature selection methods for text classification. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 230-239). ACM.

Dash, M., Liu, H., 2008. Dimensionality reduction. In Wiley Encyclopedia of Computer Science and Engineering.

Davidov, D., Tsur, O., Rappoport, A., 2010, August. Enhanced sentiment learning using twitter hashtags and smileys. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters (pp. 241-249). Association for Computational Linguistics.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., Harshman, R. A., 1990. Indexing by latent semantic analysis. JAsIs, 41(6), 391-407.

D'hondt, J., Vertommen, J., Verhaegen, P. A., Cattrysse, D., Duflou, J. R., 2010. Pairwise-adaptive dissimilarity measure for document clustering. Information Sciences, 180(12), 2341-2358.

Dietterich, T. G., 1997. Machine-learning research. AI magazine, 18(4), 97.

Diri, B., Amasyalı, M. F., 2003. Automatic Author Detection for Turkish Texts. In Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP) (pp. 138-141).

Doğan, S., Diri, B., 2010. Türkçe Dokümanlar İçin N-gram Tabanlı Yeni Bir Sınıflandırma (Ng-ind): Yazar, Tür ve Cinsiyet. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 3(3).

Faiz, S. I., Mercer, R. E., 2013. Identifying explicit discourse connectives in text. In Advances in Artificial Intelligence (pp. 64-76). Springer Berlin Heidelberg.

Forman, G., 2003. An extensive empirical study of feature selection metrics for text classification. The Journal of machine learning research, 3, 1289-1305.

Fradkin, D., Muchnik, I., 2006. Support vector machines for classification. Discrete methods in epidemiology, 70, 13-20.

Galavotti, L., Sebastiani, F., Simi, M., 2000. Feature selection and negative evidence in automated text categorization. In Proceedings of KDD.

Gentleman, R., Huber, W., Carey, V. J., 2008. Supervised machine learning. In Bioconductor Case Studies (pp. 121-136). Springer New York.

Go, A., Huang, L., Bhayani, R., 2009. Twitter sentiment analysis. Entropy, 17.

Go, A., Bhayani, R., Huang, L., 2009. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1, 12.

Güran, A., Akyokuş, S., Bayazıt, N. G., Gürbüz, M. Z., 2009. Turkish text categorization using N-gram words. In Proceedings of the International

Symposium on Innovations in Intelligent Systems and Applications (INISTA 2009) (pp. 369-373).

Grimmer, J., Stewart, B. M., 2013. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. Political Analysis, mps028.

Gunn, S. R., 1998. Support vector machines for classification and regression. ISIS technical report, 14.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. The Journal of Machine Learning Research, 3, 1157-1182.

Hall, M. A., 1999. Correlation-based feature selection for machine learning (Doctoral dissertation, The University of Waikato).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H., 2009. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.

Han, J., Kamber, M., Pei, J., 2011. Data mining: concepts and techniques: concepts and techniques. Elsevier.

Hotho, A., Nürnberger, A., Paaß, G., 2005. A Brief Survey of Text Mining. In Ldv Forum (Vol. 20, No. 1, pp. 19-62).

Ikonomakis, M., Kotsiantis, S., Tampakas, V., 2005. Text classification using machine learning techniques. WSEAS Transactions on Computers, 4(8), 966-974.

Jiang, S., Pang, G., Wu, M., Kuang, L., 2012. An improved K-nearest-neighbor algorithm for text categorization. Expert Systems with Applications, 39(1), 1503-1509.

Kanaris, I., Kanaris, K., Houvardas, I., Stamatatos, E., 2007. Words versus Character N-Ggrams for Anti-Spam Filtering. International Journal on Artificial Intelligence Tools.

Karagülle, F., 2008. Destek vektör makinelerini kullanarak yüz bulma.

Kaşıkçı, T., Gökçen, H., 2014. Metin Madenciliği İle E-Ticaret Sitelerinin Belirlenmesi. International Journal of Informatics Technologies, 7(1).

Kaya, M., Fidan, G., Toroslu, I. H., 2012. December. Sentiment analysis of turkish political news. In Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01 (pp. 174-180). IEEE Computer Society.

Kibriya, A. M., Frank, E., Pfahringer, B., Holmes, G., 2005. Multinomial naive bayes for text categorization revisited. In AI 2004: Advances in Artificial Intelligence (pp. 488-499). Springer Berlin Heidelberg.

Kim, S. B., Han, K. S., Rim, H. C., Myaeng, S. H., 2006. Some effective techniques for naive bayes text classification. Knowledge and Data Engineering, IEEE Transactions on, 18(11), 1457-1466.

Ko, Y., 2012. August. A study of term weighting schemes using class information for text classification. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval (pp. 1029-1030). ACM.

Kohavi, R., 1995. August. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).

Korde, V., Mahender, C. N., 2012. Text classification and classifiers: A survey. International Journal of Artificial Intelligence & Applications (IJAIA), 3(2), 85-99.

Kotsiantis, S. B., Zaharakis, I., Pintelas, P., 2007. Supervised machine learning: A review of classification techniques.

Kouloumpis, E., Wilson, T., Moore, J., 2011. Twitter sentiment analysis: The good the bad and the omg!. Icwsm, 11, 538-541.

Küçük, D., Steinberger, R., 2014. Experiments to Improve Named Entity Recognition on Turkish Tweets. arXiv preprint arXiv:1410.8668.

Lewis, D. D., Ringuette, M., 1994. A comparison of two learning algorithms for text categorization. In Third annual symposium on document analysis and information retrieval (Vol. 33, pp. 81-93).

Lewis, D. D., Yang, Y., Rose, T. G., Li, F., 2004. Rcv1: A new benchmark collection for text categorization research. The Journal of Machine Learning Research, 5, 361-397.

Li, Y. H., Jain, A. K., 1998. Classification of text documents. The Computer Journal, 41(8), 537-546.

Lin, D., 1998. An information-theoretic definition of similarity. In ICML (Vol. 98, pp. 296-304).

Lin, Y. S., Jiang, J. Y., Lee, S. J., 2014. A similarity measure for text classification and clustering. Knowledge and Data Engineering, IEEE Transactions on, 26(7), 1575-1590.

Liu, H., Sun, J., Liu, L., Zhang, H., 2009. Feature selection with dynamic mutual information. Pattern Recognition, 42(7), 1330-1339.

Liu, B., Zhang, L., 2012. A survey of opinion mining and sentiment analysis. In Mining text data (pp. 415-463). Springer US.

Manning, C. D., Raghavan, P., Schütze, H., 2008. Introduction to information retrieval (Vol. 1, p. 496). Cambridge: Cambridge university press.

Maron, M. E., 1961. Automatic indexing: an experimental inquiry. Journal of the ACM (JACM), 8(3), 404-417.

Martineau, J., Finin, T., 2009. May. Delta TFIDF: An Improved Feature Space for Sentiment Analysis. In ICWSM.

Mayer, R., Neumayer, R., Rauber, A., 2008. October. Combination of audio and lyrics features for genre classification in digital audio collections. In Proceedings of the 16th ACM international conference on Multimedia (pp. 159-168). ACM.

McCallum, A., Nigam, K., 1998. A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, pp. 41-48).

Meyer, D., 2014. Support vector machines. The Interface to libsvm in package e1071. WIEN, FH Technikum.

Michelson, M., Macskassy, S. A., 2010. October. Discovering users' topics of interest on twitter: a first look. In Proceedings of the fourth workshop on Analytics for noisy unstructured text data (pp. 73-80). ACM.

Ng, H. T., Goh, W. B., Low, K. L., 1997. Feature selection, perceptron learning, and a usability case study for text categorization. In ACM SIGIR Forum (Vol. 31, No. SI, pp. 67-73). ACM.

Nigam, K., McCallum, A. K., Thrun, S., Mitchell, T., 2000. Text classification from labeled and unlabeled documents using EM. Machine learning, 39(2-3), 103-134.

Oflazer, K., Bozşahin H.C., 2006. Türkçe Doğal Dil İşleme. Ç.Ü. Türkoloji-Makale Bilgi Sistemi, http://turkoloji.cu.edu.tr/DILBILIM/turkce_dogal_dil_isleme.pdf (09.07.2015)

Ozgür, A., 2004. Supervised and unsupervised machine learning techniques for text document categorization (Doctoral dissertation, Bogaziçi University).

Pak, A., Paroubek, P., 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In LREC (Vol. 10, pp. 1320-1326).

Pang, B., Lee, L., Vaithyanathan, S., 2002. Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

Patra, A., Singh, D., 2013. A Survey Report on Text Classification with Different Term Weighing Methods and Comparison between Classification Algorithms. International Journal of Computer Applications, 75(7).

Polettini, N., 2004. The vector space model in information retrieval-term weighting problem. Entropy, 1-9.

Porter, M. F., 1980. An algorithm for suffix stripping. Program, 14(3), 130-137.

Prabowo, R., Thelwall, M., 2009. Sentiment analysis: A combined approach. Journal of Informetrics, 3(2), 143-157.

Rennie, J. D., Shih, L., Teevan, J., Karger, D. R., 2003. Tackling the poor assumptions of naive bayes text classifiers. In ICML (Vol. 3, pp. 616-623).

Saad, M. K., 2010. The impact of text preprocessing and term weighting on Arabic text classification (Doctoral dissertation, The Islamic University-Gaza).

Salton, G., Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5), 513-523.

Salton, G., Wong, A., & Yang, C. S., 1975. A vector space model for automatic indexing. Communications of the ACM, 18(11), 613-620.

Schapire, R. E., 2003. The boosting approach to machine learning: An overview. In Nonlinear estimation and classification (pp. 149-171). Springer New York.

Schoenharl, T. W., Madey, G., 2008. Evaluation of measurement techniques for the validation of agent-based simulations against streaming data. In Computational Science–ICCS 2008 (pp. 6-15). Springer Berlin Heidelberg.

Sebastiani, F., 1999. A tutorial on automated text categorisation. In Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence (pp. 7-35). Buenos Aires, AR.

Sebastiani, F., 2002. Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), 1-47.

Sebastiani, F., "Text categorization", Alessandro Zanasi (ed.) Text Mining and its Applications, WIT Press, Southampton, UK, pp. 109-129, 2005.

Sommer, S., Schieber, A., Hilbert, A., Heinrich, K., 2011. Analyzing customer sentiments in microblogs–A topic-model-based approach for Twitter datasets. In Proceedings of the Americas Conference on Information Systems (AMCIS).

Srividhya, V., Anitha, R., 2010. Evaluating preprocessing techniques in text categorization. International journal of computer science and application, 47(11).

Stamatatos, E., Fakotakis, N., Kokkinakis, G., 2000. Automatic text categorization in terms of genre and author. Computational linguistics, 26(4), 471-495.

Strehl, A., Ghosh, J., Mooney, R., 2000. Impact of similarity measures on web-page clustering. In Workshop on Artificial Intelligence for Web Search (AAAI 2000) (pp. 58-64).

Tan, A. H., 1999, April. Text mining: The state of the art and the challenges. In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases (Vol. 8, pp. 65-70).

Tan, S., 2006. An effective refinement strategy for KNN text classifier. Expert Systems with Applications, 30(2), 290-298.

Tang, H., Tan, S., Cheng, X., 2009. A survey on sentiment detection of reviews. Expert Systems with Applications, 36(7), 10760-10773.

Tantuğ, A. C., 2012. Metin Sınıflandırma (Text Classification). Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 6(6).

Torunoğlu, D., Çakırman, E., Ganiz, M. C., Akyokuş, S., Gürbüz, M. Z., 2011. June. Analysis of preprocessing methods on classification of Turkish texts. In Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on (pp. 112-117). IEEE.

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

Türkmenoğlu, C., Tantuğ, A. C., 2014. Sentiment Analysis in Turkish Media. ICML (International Conference on Machine Learning), Beijing

Uysal, A. K., Gunal, S., 2012. A novel probabilistic feature selection method for text classification. Knowledge-Based Systems, 36, 226-235.

Wang, J., Neskovic, P., Cooper, L. N., 2007. Improving nearest neighbor rule with a simple adaptive distance measure. Pattern Recognition Letters, 28(2), 207-213.

Weston, J., Watkins, C., 1998. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, May.

Wiemer-Hastings, P., 2004. Latent Semantic Analysis.

Xu, Y., Jones, G. J., Li, J., Wang, B., Sun, C., 2007. A study on mutual information-based feature selection for text categorization. Journal of Computational Information Systems, 3(3), 1007-1012.

Yang, Y., 1999. An evaluation of statistical approaches to text categorization. Information retrieval, 1(1-2), 69-90.

Yang, Y., Liu, X., 1999. A re-examination of text categorization methods. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 42-49). ACM.

Yang, Y., Pedersen, J. O., 1997. A comparative study on feature selection in text categorization. In ICML (Vol. 97, pp. 412-420).

Yıldız, H. K., Gençtav, M., Usta, N., Diri, B., Amasyalı, M. F., 2007. A new feature extraction method for text classification. In Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th (pp. 1-4). IEEE.

Zhang, H., Li, D., 2007. Naïve Bayes text classifier. In Granular Computing, 2007. GRC 2007. IEEE International Conference on (pp. 708-708). IEEE.

Zheng, Z., Wu, X., Srihari, R., 2004. Feature selection for text categorization on imbalanced data. ACM Sigkdd Explorations Newsletter, 6(1), 80-89.

ÖZGEÇMİŞ

Önder ÇOBAN 1988 yılında Erzurum'da doğdu. İlk, orta ve lise öğrenimini Erzurum'da tamamladı. 2009 yılında Süleyman Demirel Üniversitesi Yabancı Diller Yüksek Okulu'nda İngilizce dil eğitimini, 2013 yılında Süleyman Demirel Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde lisans eğitimini tamamladı. 2014 yılından itibaren Atatürk Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünde araştırma görevlisi olarak görev yapmaktadır.