T.C.

GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

LİSANS BİTİRME PROJESİ ARA RAPORU

AKILLI E-POSTA ASİSTANI

Kenan SAVAŞ

Danışman

Prof. Dr. İbrahim SOĞUKPINAR

Haziran, 2016

Gebze, KOCAELİ

T.C.

GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

LİSANS BİTİRME PROJESİ ARA RAPORU

AKILLI E-POSTA ASİSTANI

Kenan SAVAŞ

Danışman

Prof. Dr. İbrahim SOĞUKPINAR

Haziran, 2016

Bu çalışma .... / .... / ……. tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümünde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı	Prof. Dr. İbrahim SOĞUKPINAR	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı	Doç. Dr. Didem GÖZÜPEK	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı	Doç. Dr. Hasari ÇELEBİ	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

ÖNSÖZ

Hızla büyüyen bir problem olarak karsımıza çıkan istenmeyen elektronik postalar ve çığ etkisi gibi bizi bilgilendiren, ancak içeriği itibari ile bizimle alakalı olmayan pek çok e-postanın altında ezilen kullanıcıların bu kayıp zaman ve emeklerine çzöüm bulmak üzere güvenilir filtreler gelistirmek zorunluluk haline gelmistir. Şimdiye kadar geliştirilen filtrelerin çoğu el ile kullanılan tekniklere dayalı anahtar kelimeler kullanan bir filtreleme yapısında olup, bu yöntem sürekli değisme eğiliminde olan istenmeyen elektronik postaların karakterlerini belirlemede ve onları filtrelemede yetersiz kalmaktadır.

Bu çalısmada, elektronik posta filtrelemede statik olarak kullanılan yöntemlere alternatif olarak son zamanlarda kullanımı hızla artan metin madenciliği yöntemlerinden biri olan naive bayes ile sınıflamaya dayanan metin madenciliği yöntemi kullanılarak yukarıda da bahsedilen problemlere çözüm getirilmiştir.

Çalısmamın geliştirilmesinde değerli katkısı olan Prof. Dr. İbrahim SOĞUKPINAR hocam ile yanına gittiğinde yoğun çalışma temposu içinde olmasına rağmen ilgisini esirgemeyen Arş. Gör. Arzu KAKIŞIM hocama teşekkürlerimi bir borç bilirim.

Tez çalışmamda beni daima yalnız bırakmayıp bu çalışmaya özendiren ve ilgisini eksik etmeyen değerli eşim Halide Hanım'a da teşekkür ediyorum.

Kenan SAVAŞ

İstanbul, 2016

İÇİNDEKİLER

ŞEKİL LİSTESİ

TABLO LİSTESİ

KISALTMA LİSTESİ

E-Mail **:** E-Posta, Elektronik posta

GTÜ **:** Gebze Teknik Üniversitesi

HTML **:** Hyper Text Markup Language (Akış metin işaretleme dili)

HTTP **:** Güvensiz internet bağlantısı

HTTPS **:** Güvenli internet bağlantısı

SPAM **:** İstenmeyen e-posta

UML **:** Unified Modeling Language (Birleşik Modelleme Dili)

URL **:** İnternet bağlantı linki

SEMBOL LİSTESİ

E **:** Şifreleme Algoritması

D **:** Deşifreleme Algoritması

M **:** Şifrelenecek Metin (Plaintext)

C **:** Şifrelenmiş Metin (Ciphertext)

ÖZET

Bu çalışma sadece Türkçe içerikli istenmeyen elektronik postaların otomatik olarak filtrelenmesi problemine metin madenciliğine dayalı bir uygulama ile çözüm bulmak amacıyla yapılmıştır. Öncelikle, literatürde veri ve ardından metin madenciliği ile ilgili çalışmalar yapılmış ve yapılan çalışmaların özeti sunulmuştur. Daha sonra istenmeyen elektronik postaların karakteristikleri hakkında bilgi verildikten sonra mevcut istenmeyen elektronik posta engelleme yöntemleri literatürden örnekler ile açıklanmıştır. İstenmeyen elektronik postaların engellenmesi bir otomatik öğrenme (belge sınıflandırma) işlemi olarak ele alınmış ve bu alanda yaygın olarak kullanılan tfidf değerlerine dayalı naive bayes sınıflama yöntemi tanıtılmış ve çalışmada kullanılmıştır. E-posta içerikleri kelimelerine ayrılırken Türkçe dil yapısına uygun bir kullanım kolaylığı sunan açık kaynak kod olan Zemberek kütüphanesinden yararlanılmıştır. Son olarak üretilen çözümler doğrultusunda Java yazılım geliştirme ortamında, bu amaç doğrultusunda bir yazılım geliştirilmiştir. Çeşitli kullanıcılardan elde edilen istenmeyen ve normal elektronik posta mesajlarından oluşan örneklemler üzerinde yazılım çalıştırılmış ve elde edilen sonuçlar %85 üzerinde başarılı sınıflamanın yapıldığı kanıtlar niteliktedir.

Anahtar Kelimeler: spam elektronik postalar, istenmeyen elektronik postaların filtrelenmesi, belge sınıflandırma, naive bayes sınıflandırma

ABSTRACT

In this work, it is aimed that spam e-mails which are only in Turkish language could be filtered automatically based technics of the text mining consept. Firstly, the workings has been analized and their abstracts has been presented in the literatüre. After that, information about spam e-mails are given and then, the filtering methods exist in the literatüre with examples are presented. Preventing consept in this work has been accepted as a document categorization usng text mining methods. In this work, categorization methods using naive bayessian categorication based on tfidf calculations is introduced and used in the work. The contexts of the e-mails are tokenized using open source Zemberek library that is proper to the structure of Turkish language. As a last, a software has been developed using Java programming paltform using these technics. The samples from various users of spame-mails are used to test the software developed here, and the rsults of this shows that greater than 85% successful filtering is realized.

Keywords: spam e-mails, spam filtering, text mining, naive bayessian categorization

1. GİRİŞ

Günümüz dünyasında insanoğlunun tüm işlemleri ve yaşantısı neredeyse kontrol altında tutulmaktadır. İnsanlar her yerde kendileri hakkında tanımlayıcı öğeler bırakmaktadırlar. Bilgisayar ve iletişim teknolojilerindeki son gelişmeler verinin çok hızlı bir şekilde depolamasına, işlenmesine ve bilgiye dönüştürmesine imkân sağlamaktadır [1]. Bilgisayar kullanımının yaygınlaşması ve internetin hızla gelişmesi sonucunda elektronik posta hayatımızda giderek önemi artan bir haberleşme ortamı haline gelmiştir. Çok hızlı ve en ekonomik haberleşme biçimi olduğu için kullanıcılarının sayısı da kısa zamanda büyük ölçüde artmıştır [1]. Ayrıca, internete erişim kolaylığı, optik okuyucular, yüksek hızlı ağlar ve pahalı olmayan, yüksek miktardaki bilgi depolama imkanları gibi yeni teknolojik gelişmeler sonucu, online metin, makalelere, e-maillere, teknik raporlara vb. erişilebilirlik konusunda büyük bir artış yaşandı [2].

Elektronik posta, sadece iki insan arasındaki iletişimi sağlayan bir ortam olarak değil, elektronik ticaret yönetimi için de iyi bir ortam olarak popüler olmuştur [1].

Bilgisayarın hayatımıza girmesiyle birlikte birçok işin kolaylaştığı, işlemlerin daha hızlı ve doğru yapıldığı bir gerçektir. Her alanda kullanılan bilgisayar, depolanan veri miktarını da arttırmıştır. Kişiler, işletmeler ve çeşitli kuruluşlar kendileriyle ilgili her türlü veriyi dosya olarak veya veritabanlarında saklamaktadırlar. Fakat bu verilerin fayda sağlaması için işlenmesi gerekmektedir. İşlenmeyen veriler veritabanı boyutunu arttırmaktan başka bir işe yaramazlar ve işlenmediği sürece anlamsız bir yığından öteye gidemezler [3]. Teknolojik alanındaki bu hızlı gelişmeler ve veri toplamındaki hızlı artış toplanan verilerden nasıl faydalanılacağı ve bu verinin daha anlamlı hala getirileceği problemini ortaya çıkarmıştır [4].

Veri ile bilgi birbirinden ayrı ayrı kavramlardır. Veri en alt düzeyde bulunur. O, basit bir gözlem değeri içerir. İkinci aşamada bilgi bulunur. Daha yaygın kullanıma sahiptir. O veriye dayalı bir gözlemdir. Anlamlı veri de denilebilir. Veri ile veri elemanları arasındaki ilişkiler verileri anlamlı hale getirmektedir. Örneğin yaş ve ağırlık bilgileri ikisi de sayısal verilerdir lakin bunların aldığı değerler farklı anlamlara gelmektedir. Bilgi, veriye göre daha çok istenir. Bu bilgileri bulmak için doğru testler yapmak ve doğru teknikler kullanmak gerekmektedir [5].

Veri ve bilginin yönetilmesi konusu eski bir konudur. Bu ihtiyaç en azından kütüphanecilik kadar eski bir konu sayılabilir. Bu konuya, bilginin saklanması, bulunması ve gösterilmesi gözüyle bakabiliriz. Burada bizim konsantre olacağımız konu bu verilerin içinden önemli bilgileri bulmaktır [5].

Veri madenciliği, önceden bilinmeyen ve potansiyel olarak faydalı olabilecek, veri içindeki gizli bilgilerin çıkarılmasıdır [6]. Veri madenciliği yapısal veriler üzerinde çalışır. Fakat metin dosyaları yapısal olmayan verilerdir. Bu tür verilerin işlenebilmesi için yapısal hale dönüştürülmesi gerekir. Metin madenciliği bu problemlere çözüm olarak sunulan, metin formatındaki verileri kullanarak içerisindeki bilgileri gün ışığına çıkaran ve özellikle 2000'li yıllardan sonra ilginin giderek arttığı önemli bir alandır [7].

Geleneksel istatistikî tekniklerle büyük boyuttaki veriyi çözmek kolay değildir. Bu sebeple verileri işlemek ve çözümlemek için özel tekniklere ihtiyaç duyulmuştur. Geleneksel istatistik

teknikleri metin verilerinden bilgi çıkarımında etkisiz kalmış ve bunun sonucu olarak da metin madenciliği çalışmaları hızla yayılmıştır [4].

Metin madenciliği, özel amaçlar için metinden bazı bilgiler çıkarmak adına, metinin analiz edilmesi [5], metin dokümanlarının bir veri tabanı içinden kullanıcı isteklerine olan benzerliklerine göre sıralanması işlemidir [8].

Metin madenciliğinin çalışma alanı sadece bilgisayarımızda saklanılan dosyalardan ibaret değildir. E-mailler, bloglar, kişisel sayfalar, haber siteleri gibi internet ortamında bulunan verilerin de işlenmesi metin madenciliği teknikleriyle gerçekleştirilir. Metin madenciliğinin amacı bu tür verilerin, veri madenciliği tekniklerinin uygulanabileceği yapısal forma dönüştürülmesidir [9].

Metinlerin analiz işlemlerinden bir tanesi sınıflandırmadır. Metin sınıflandırma, önceden belirlenmiş sınıflara dokümanların atanması işlemidir [10].Metin sınıflandırması ise önceden belirlenmiş kategorilere göre, doğal dil metinlerinin sınıflandırılmasıdır [11]. Sonsuz sayıda doğal dil girdilerini, küçük bir kategoriler kümesine indirgemek, metin tabanlı bilgileri işleyen hesaplama sistemlerinin temel stratejisidir[2].

Metin kategorizasyonu iki açıdan önemli olmuştur. Bilgiyi bulma açısından düşünüldüğünde, internet gibi hızla gelişen metin tabanlı bilgi kaynakları sayesinde bilgiyi işleme ihtiyacı da artmıştır. Metin kategorizasyonunun buradaki kullanımı, doküman filtreleme, konuya özel işleme mekanizmalarını sağlama ve böylece bilgiyi edinme şeklindedir [2].

Makine öğrenmesi (machine learning = ML) açısından düşünüldüğünde de yapılan son araştırmalar, veri madenciliği gibi yöntemler üzerine olmuştur. Makine öğrenmesine ihtiyaç duyulmasının nedeni, elle kategorizasyonun pahalı ve zaman tüketen bir iş oluşudur ki, ayrıca elle sınıflandırmada, sınıflandırmayı yapan uzmanların vermiş oldukları kararlara bağlı olarak sonuçlar da değişmektedir [12].

Son zamanlarda Metin Kategorizasyonu, çok çeşitli uygulama alanları bulmuştur. Bu uygulama alanları;

- Metin bulup getirme ve kütüphane organizasyonu gibi alanlarda destek sağlayan, "dokümanlara kategori ataması yapmak",
- Bu kategorileri insanların atadığı uygulamalarda "kategori atamasında yardımcı rol oynamak",
- Mesajları, haberleri ve diğer "metin akımı (stream) halindeki bilgileri alıcılara ulaştırmak",
- Doğal dil işleme sistemlerinin bir parçası olarak; "ilgisiz metinleri ve metin parçalarını filtrelemek", "metinleri, kategori bazlı işleme mekanizmalarına yönlendirmek" veya "sınırlı şekillerde bilgi edinimini sağlamak", "sözcük analizi işlerinde yardımcı olmak (sözcük belirsizliğini giderme gibi)" vb. [2].

Metin kategorizasyonu yaparken iki temel adım vardır. İlki, performansın değerlendirileceği kategorizasyon algoritmasını seçmek; ikincisi ise, algoritmanın üzerinde uygulanacağı örnek veri kümesini seçmek [2].

Araştırmacılar, eğitim dokümanları kümesinin yeterince bilgi içermesi gerektiğini, ancak bunun yanında zaman performansını sağlamak adına büyük miktarda veri içermemesi gerektiğini

ortaya koymuşlardır. Zaman performansını sağlayıcı nitelikte eğitim dokümanı bulundurmak, büyük veriler için kategorizasyon problemlerini çözmede önemli olmaktadır [2].

Günümüze kadar araştırmacılar, genel olarak İngilizce veya İngilizce'ye yapısal olarak benzeyen dillerle ilgilenmişlerdir. Böyle dillerde kelimelere eklenen az sayıda ek mevcuttur. Bu tür dillerde metin kategorizasyonu yapılırken ekler doğrudan atılır ve ekler üzerinde herhangi bir yapısal analiz yapılmaz. Bu da kelimelerin kolay işlenmesini ve köklerinin kolayca bulunmasını sağlar [2].

Türkçe gibi bitişken (bükümlü) dillerde ise kelimeler, en küçük anlamlı parçasının sınırlarına dair bir belirti göstermez, üstelik bu parçalar, morfolojik ve fonolojik şartlara bağlı olarak şekil alırlar. Türkçe'de bir kelimenin son ekine bir tane daha ekleyerek, nispeten uzun kelimeler elde edilebilir, üstelik, sadece bir tek Türkçe kelimeden çok miktarda değişik anlamlı kelimeler oluşturulabilir. Bu karmaşık morfolojik yapı yüzünden, Türkçe; İngilizce'den ve benzeri dillerden daha farklı metin işleme teknikleri gerektirir. Bu nedenle, bütün kelimelerin küçük harfe çevrilmesi ve noktalama işaretlerinin kaldırılması dışında; joker kelimeler ile anahtar kelimelerin oluşturulması gibi bazı ön hazırlıklar yapılması gerekmektedir [12].

Metin sınıflandırmada boyut azaltmanın etkileri ve özellik seçimi konusunun işlendiği çalışmada, tf-idf olarak ağırlıklandırılmış sözcükler kullanılarak, iki farklı veri seti ile çalışılmıştır. Sınıflandırmada kullanılacak sözcük sayısı, orijinal metin sayısının %90,00'ından %10,00'una kadar kademeli olarak düşürülerek sonuçlar gözlenmiştir. Toplam sözcüklerin %10,00'u ile yapılan sınıflandırmada, başarının birinci veri setinde %6,25, ikinci veri setinde ise %11,39 arttığı gözlemlenmiştir [13].

Metinlerin sınıflandırılmasında olduğu gibi web sayfalarının sınıflandırılmasında da metin madenciliği yöntemleri kullanılmaktadır. Çünkü web sayfaları da metin içerikli veri barındırmaktadır ve bu verilerin, veri madenciliği tekniklerinin uygulanabileceği yapıya dönüştürülmesi işlemi metin madenciliği ile gerçekleştirilmektedir [9].

Çeşitli sınıflandırma algoritmaları uygulanarak yapılan web sayfası sınıflandırması çalışmasında, sınıflarda sıkça geçen kelimeler için pozitif, seyrek geçen kelimeler içinde negatif ağırlık kullanarak özellik seçimi uygulanmıştır. Sonuçlara göre, en yüksek sınıflandırma başarısı %91,12'dir. Fakat çalışmada kullanılan özellik seçimiyle k-NN uygulandığında, %90,62 olan sınıflandırma %90,80'e çıkmıştır [14].

Bir taraftan internet üzerinde ürünlerinin (çoğunlukla güvenilmeyen, hileli ürünler) ticari reklâmını yapmak veya yasal olmayan duyurularda bulunmak isteyenler, diğer taraftan da elektronik posta adres listelerinin sayısının artması geçtiğimiz birkaç yıl içerisinde "istenmeyen elektronik posta" (spam) kavramının ortaya çıkmasına sebep olmuştur. Bugün çok ileri boyutlara ulaşan ve kullanıcılar açısından zaman kaybı, işletmeler açısından ise verimlilik kaybına sebep olan istenmeyen elektronik posta probleminin önüne geçebilmek amacıyla bu tür elektronik postaları otomatik olarak filtreleyebilen metotlar bir gereklilik haline gelmiştir. İstenmeyen elektronik posta problemini tamamen çözebilmiş bir teknik veya tekniklerin birleşmesinden oluşan bir çözüm mevcut değildir. İstenmeyen elektronik posta trafiği çok yoğun kuruluşlar için yapılacak en iyi şey birkaç tekniği birleştirmek olacaktır [1].

Spam internet alanında önemli bir sorundur. Spamın tam olarak bir tanımı olmamasına rağmen (diğer adıyla önemsiz posta olarak da bilinir), aynı zamanda spam olmayanlar gerçek posta olarak adlandırılır. Kısa popüler tanımı spam için "istenmeyen toplu e-posta" (unsolicited bulk email, UBE) gibi karakterize edilir [15] ya da bazen ticari olarak (UCE) kelimesi kullanılır. TREC Spam Track firmasına göre, spam "kullanıcı ile hiçbir geçerli ilişkisi olmayan bir

gönderici tarafından, doğrudan veya dolaylı olarak, gelişigüzel gönderilen istenmeyen e-postadır" [16].

Spam ayrıca antispam sitelerinde aşağıdaki gibi tanımlanır [17]: Spam yoksa alması seçmelisiniz değil kişilere mesajı almak istemeyen kişilere zorla aynı mesajı toplucva tüm kişilere birden göndermektir. Çoğu spam mesajlar kuşku duyulan çeşitli ürünler için ticari reklam amaçlı olup hızlı şekilde tasarıma sahip bir formatta oluşturulmuş zengin metin biçimine sahiptir. Spam göndermek gönderen için maliyeti çok azdır. Maliyetlerin çoğu gönderici yerine alıcı veya taşıyıcılar tarafından karşılanır [18].

Spam mesajlarının iki ana türü vardır ve bunların internet kullanıcıları üzürindende farklı etkileri vardır: grup (usenet) spam ve e-posta spam. İptal edilebilir grup spam 20 veya daha fazla haber gruplarına gönderilen tek mesajdır. Grup spamın hedefi "gözlemciler", yani haber gruplarını okuyan nadirde olsa ama nadiren ya da hiç postalanmamış veya verilmemiş uzak adreslere de olabilir. Grup spamlar haber gruplarını kullanan kullanıcılara istenmeyen reklamları veya ilişkili postaların baskıcı etkisi ile kişisel bilgilerini çalar. Ayrıca, grup spam yönetici veya sahiplerine sistemlerinde kabul ettikleri konuları yönetme yeteneği katar [18].

E-posta spamlar bireysel kullanıcıları doğrudan posta mesajları ile hedefler. E-posta spam listeleri genellikle grup ilanları taranarak, internet e-posta listeleri çalınarak veya adres bilgileri Web'de aranmak suretiyle oluşturulur. E-posta spamler tipik olarak kullanıcıların paralarını harcamalarını neden olurlar. Birçok kişinin ve özellikle sunulan telefon hizmetleri sırasındaki ölçümler, veri analizi ve sayaç faaliyetleri sırasında kullanıcıların postaları tabiri caizse alınır veya okunur. Spamlar servis sağlayıcılara ek masrafa neden olur. Bu durumun en doğal sonucu olarak çevrimiçi hizmetler için ISS'ler spam gönderilmesi sebebiyle oluşan masrafları abonelere doğrudan iletir [18].

E-posta spamların özellikle kötü bir çeşidi kamu ya da özel e-posta tartışma forumlarındaki posta listelerine spam gönderir. Spamcıların mümkün olduğunca çoğu posta listelerine otomatik kayıt olma araçları kullanması ve adres listelerini alabileceği ve amaçlarına uygun doğrudan hedef olarak posta listelerini kullanabilecekleri nedeniyle birçok e-posta listeleri abonelerinin aktivitelerini sınırlandırır [18].

Spam e-postalar alınan e-postaların yaklaşık %80'ini oluşturması nedeniyle istenmeyen posta önemli bir sorundur. Spamlar finansal, depolama alanı, hesaplama gücü ve e-postaları silerken geçen üretken zaman gibi pek çok faktörde kayıp ve problemlere neden olur. Spam e-postalar bunların yanında meşru olmayan reklamlar gibi yasal sorunlara da neden olmaktadır. Ferris Araştırma Analiz Bilgilendirme Servisi spam nedeniyle dünya çapında toplam mali kayıpların 2005 yılında 50 milyar $ olduğunu tahmin etmiştir [19].

Spam e-postaların olumsuz etkileri nedeniyle istenmeyen e-postaların filtrelenmesi güncel bir konudur. Son zamanlarda "Şirketlerde Anti Spam Stratejileri" üzerine yapılan çalışmalarda [11], filtreleme en yaygın olarak kullanılan yöntemdir ve yakın gelecekte en sık kullanılan yöntem olmaya devam edecektir. Makine öğrenmesi tekniklerine dayalı spam filtreleme önemli bir pratik uygulama olacağı tahmin edilmektedir. Bu durum insan müdahalesi olmadan yeni tip spamların belirlenmesini sağlayacaktır [18].

Spam tespiti ve filtrelemesi için birçok yaklaşım vardır. Spamcıların yeni spam postaları için kendini geliştiren uygulamaları filtre kurallarını ihlal etmektedir. Bu nedenle öğrenme dayalı adaptif algılama spamlar ile başa çıkmada önemli bir rol oynamaktadır [18].

Öğrenme tabanlı adaptif algılama sistemlerinin kombinasyonu spam e-postaları daha iyi filtremektedir. Bu çalışmanın temel amacı bulanık kural tabanının genetik algoritmaya dayalı olarak ayarlanması ile Genetik Algoritma ile Adaptif Neuro-Fuzzy Çıkarım Sistemi kombinasyonunu kullanarak düşük hata oranı üretmeğe dayalı bir çözüm sunmaktır [18].

Spam tespiti ve filtrelemesinde kullanılan tekniklerin en yenilerinden birisi olan Bayes Filtreleme artık istenmeyen elektronik posta engelleme yazılımlarının bileşenleri arasında son derece başarılı sonuçlar üreterek yerini almıştır [1]. Teoride metin sınıflandırma metotlarından Yalın Bayes sınıflandırmanın elektronik postalara uygulanmasıyla geliştirilmiş şekli olan Bayes Filtreleme yöntemi kullanılmaktadır [1].

1.1. Proje Tanımı

Bu tez çalışmasının amacı, metin madenciliği yöntemi ile alınan e-posta mesajlarının otomatik olarak alınıp sınıflandırılmasıdır. Çalışma kapsamında veri madenciliği ile metin madenciliği hakkında bilgiler verilecek ve elektronik posta sisteminin genel yapısı ve mevcut istenmeyen elektronik posta engelleme teknikleri tanıtılacaktır. Bu çalışmada istenmeyen elektronik postaların filtrelenmesi bir belge (metin) sınıflandırma problemi bağlamında ele alınacaktır. Çalışmada 10 farklı kişiye ait 50'şer adet spam e-postanın ve 100'er adet normal e-postanın sınıflandırılması amaçlanmıştır. İşlemleri gerçekleştirmek için Java ile Netbeans IDE ortamında yazılım geliştirilmesi amaçlanmıştır.

Bu çalışmanın amacı özellikle tf-idf yöntemini kullanarak Türkçe metinlerde bulunan verileri belirli başlıklar altında kümeleyerek gerekli bilgiyi elde etmektir. Metin madenciliği alanında farklı yöntemler kullanılmasına rağmen bu yöntemin seçilmesinin sebebi yaygın olarak ve kolay bir şekilde kullanılabilir oluşudur. Bu yöntem ile yazılı belgeler arasındaki anahtar kelimelerin çıkarılarak aralarındaki örüntülerin/ilişkilerin bulunması hedeflenmektedir. Bu çalışmada 10 farklı veri seti üzerinde uygulama yapılıp deneysel sonuçların elde edinilmesi düşünülmüştür. Geliştirilen prototip yazılımla kullanılan veri setlerine ait e-posta metinleri kategorizasyon işlemi için çeşitli işlemlere tabi tutulacak, kullanılan yöntem veya yöntemlerin Türkçe istenmeyen ve normal elektronik posta mesajlarından oluşturulmuş bir derleme farklı yollarla uygulanarak ve performans değerlendirmesi, karşılaştırması incelenmiş ve sonuçlar yorumlanmıştır.

1.2. Projenin Neden Ve Amaçları

Bu proje ile kullanıcıların geliştirilecek yazılımı kullanarak e-postalarını daha güvenle okumaları ve istenmeyen e-postalar ile daha sonra okumayı düşündükleri ya da normal olarak gördükleri e-postaları arasında otomatik ve kullanıcıya özgü sınıflandırma yaparak gereksiz içeriğe sahip olan postaları okuyarak bu yazılımı kullanmadan önce geçirdikleri verimsiz zaman yerine kullanıcılara yaşam faaliyetlerini daha efektif bir şekilde ve zamanı daha etkin kullanabilmesi amaçlanmıştır.

2. TEMEL BİLGİLER VE İLGİLİ ÇALIŞMALAR

Elektronik postaların kategorilere ayrılması bir belge(metin) sınıflandırma problemi olarak ele alınmış ve belge sınıflandırma algoritmalarından kümeleme algoritmaları altında ele alınabilecek bir filtreleme tekniği kullanılarak geliştirilen bir prototip yazılımla, çeşitli kullanıcılardan toplanmış istenmeyen ve normal elektronik posta mesajlarından oluşan bir örneklem üzerinde test edilecektir. Bu işlem yapılırken örneklem, talim kümesi ve test kümesi olarak iki kısma ayrılacaktır. Geliştirilen prototip filtre önce talim kümesi üzerinde öğrenme işlemini gerçekleştirecek sonra da test kümesi üzerinde filtreleme işlemi yapacaktır. Elde edilen sonuçların etkinliği doğru kategoriye ayrılmış olup olmama durumuna ve ileride katılması düşünülebilecek yöntemlerle değerlendirilecektir.

2.1. Veri Madenciliği

Gelişen teknolojiyle birlikte bilgisayarlar gündelik hayatımıza daha sık girmektedir. Bu sayede günlük hayatımızda yaptığımız her işlem bilgisayarlarda depolanmaktadır. Örneğin; gün içerisinde bir mağazaya, alışveriş merkezine girerken veya çıkarken, bir banka içerisinde beklerken yaptığımız her işlem güvenlik sebebiyle kameralar tarafından düzenli olarak kayıt altına alınmakta ve veritabanlarında saklanmaktadır. Bu şekilde yapılan kayıtların ardından veritabanlarında bir veri yığını oluşacaktır. Bu kayıtlar arasından istenilen bir bilgi çekileceği zaman, hızlı olarak istenilen sonuçlar alınamayabilir. Oluşabilecek bu veri yığınları içerisindeki önemli bilgilerin süzülmesi, çıkartılması gerekmektedir. Düzenli olarak kaydedilen tüm bu veriler, veritabanlarında çıkarılmayı bekleyen değerli bir maden gibi durmaktadır [20]. Bu değerli maden ayıklanıp, işlendikten sonra anlamlandırılarak işe yarar bilgi haline gelmektedir.

1960'lardan beri veritabanı teknolojilerindeki gelişmelere paralel olarak bilgisayar donanımı ve iletişim teknolojilerindeki baş döndürücü ilerlemeler sayesinde son derece güçlü bilgisayarların, veri toplama ekipmanlarının ve veri depolama ortamlarının kullanılabilir olması ve yaygınlaşması mümkün olmuş, bu da işlem yönetimi (transaction management), bilgi erişim (information retrieval) ve veri analizi için kullanılan çok sayıda ve son derece büyük veri deposunun ortaya çıkmasına olanak sağlamıştır [21].

Bilgiye ulaşma sürecinde büyük verilerin hepsinin aynı anda kullanımıyla istenilen bilginin elde edilmesi uzun zaman alacak ve ekonomik açıdan da masrafı fazla olacaktır. Bu durum, büyük veritabanları içerisinden talep edilen bilgilerin, düşük masraflı ve hızlı bir şekilde çekilmesi gerekliliğini karşımıza çıkarmaktadır [22]. Büyük veri yığınları üzerinde çalışmak üzere güçlü, verimli ve ölçeklenebilir araçlara ve tekniklere ihtiyaç duyulmaktadır. Yapılan araştırmalar sonucu elde edilen teknikler ile çok miktarda veriyi son derece az bir çabayla ve son derece düşük bir maliyetle toplamak ve depolamak mümkün olmaktadır [23]. Bu ihtiyacı karşılamaya yönelik olarak Veri Madenciliği (Data Mining) olarak adlandırılan disiplinler arası bir uzmanlık alanı ortaya çıkmıştır.

Veri madenciliği, bilgiye erişim sürecinde; makine öğrenmesi, veritabanı veya veri ambarı yönetimi, matematiksel ve istatistiksel teknikleri kullanarak önceden tahmin edilemeyecek olan bilgiye ulaşabilmektedir [22].

Uzun yıllardır başarı ile kullanılan veritabanı yönetim araçları ve istatistiksel analiz teknikleri, büyük veri yığınları üzerinde etkin ve hızlı sonuç gerektiren çalışmalarda yetersiz kalmaktadır. Veritabanlarında Bilgi Keşfi (Knowledge Discovery in Databases - KDD) olarak da adlandırılan veri madenciliği, veritabanları ve veri ambarları gibi çeşitli veri depolarında saklanmakta olan büyük miktardaki verinin işlenerek içindeki geçerli, daha önceden bilinmeyen, potansiyel olarak kullanışlı, yararlı ve değerli olabilecek bilginin çıkartılması sürecidir [24]. Veri madenciliği, veritabanı teknolojileri, istatistik, yapay zeka, örüntü tanıma, yüksek başarımlı hesaplama, veri görselleştirme, bilgi erişim, imge ve sinyal işleme, uzaysal ve zamansal veri analizi gibi çeşitli disiplinlerden tekniklerin bütünleştirildiği bir uygulama alanıdır [3].

Son yıllarda insanların yaptıkları her bankacılık işlemi, her alışveriş, neredeyse insanların her adımı uzaktan algılayıcılar, uydular tarafından kontrol altında tutulmakta ve yapılan birçok işlemin kolayca veritabanlarında saklanılabilir olduğu bilinmektedir. Bu şekilde bilgilerin kayıt altında tutulması veritabanlarının inanılmaz boyutlarda artmasına neden olmaktadır. Yalnızca uydu ve diğer uzay araçlarından elde edilen görüntülerin saatte 50 gigabyte düzeyinde olması, bu artışın boyutlarını daha açık bir şekilde göstermektedir. Hacimlerdeki bu büyük artış sebebiyle depolanan kayıtlar ve olaylar sonucu toplanan bu verilerden nasıl yararlanılacağı konularında araştırmalar yapılmıştır. Bu araştırmalar sonrasında "veritabanlarında bilginin keşfi (knowledge discovery in databases) karşımıza çıkmaktadır. Bilginin keşfi süreci çeşitli basamaklardan oluşmaktadır. Bu süreç içerisinde en önemli basamak veri madenciliği aşamasıdır. Bu önem sebebiyle "veritabanlarında bilginin keşfi" sürecine birçok araştırmacı "veri madenciliği" demektedir [25].

2.2. Metin Madenciliği

Veri madenciliği üzerine yapılan çalışmalar genelde veri ambarlarında ve ilişkisel veritabanı gibi yapısal veriler üzerine kurulmuştur. Erişilebilir ve kullanılabilir durumdaki verinin önemli bir bölümü metin veritabanlarında bulunmaktadır. Bu veritabanları çeşitli kaynaklardan oluşan (e-posta, araştırma bildirileri, haberler, sayısal kütüphaneler, kitaplar, makaleler, web sayfaları vs.) geniş döküman koleksiyonlarından oluşmaktadır. Elektronik ortamdaki bilgi miktarındaki artış nedeniyle metin veritabanlarının boyutları da hızla artmaktadır [26]. Tahminlere göre iş dünyasıyla ilgili bilginin %85'i metin formatında saklanmaktadır [27].

Metin Madenciliği (Text Mining) ise, ilginç, yararlı ve henüz keşfedilmemiş bilginin, metin halindeki veriden, bilgi işlem metotları ile elde edilmesi olarak tanımlanabilir [28]. **Bilgi Patlaması** (information explosion / information overload) adlı soruna çözüm bulmayı amaçlayan bir araştırma alanıdır. Bu sorunu çözmeye çalışırken; veri madenciliği, metin madenciliği, yapay zeka, istatistik, doğal dil işleme (NLP Natural Language Processing), bilgi yönetimi (Knowledge Management) ve bilgi erişim (IR Information Retrieval) tekniklerini kullanır. Metin Madenciliği, döküman koleksiyonlarının önişlemeden geçirilmesi, ara sonuçların saklanması, ara sonuçları analiz edebilmek için çeşitli tekniklerin kullanılması ve elde edilen sonuçların görselleştirilmesi gibi aşamalardan oluşmaktadır [26].

Metin madenciliği çeşitli amaçlar için yapılabilir. Bunlardan bazıları, yazıyı özetleme, sınıflandırma, kategorize etme, kelime sıklığı ve kelimeler arası ilişki gibi istatistiki özelliklerini belirleme, duygu veya sentiment analizi yapma, bilgi çıkarımı gibi uygulamalardır. Bir örnek verilecek olursa, popüler bir araç olan Twitter yazı verileri üzerinde yapılan duygu (sentiment) analizi anlatılabilir. Mesela bir X markası reklam kampanyası yaptığında ve Twitter üzerinde belirli bir zaman diliminde o marka hakkında yapılan yazışmaların ne oranda olumlu ve olumsuz olduğunu duygu analizi ile öngörebilir. Bunu gerçeklemek için kullanılabilecek algoritma yaklaşımlarından bir tanesi en basit şekilde şöyle olabilir: Her tweet de geçen olumlu kelimelere (iyi, güzel vb. gibi) +1 ve olumsuz kelimelere (kötü, çirkin vb. gibi) -1 puanı verilerek bir tweet için verilen tüm puanlar toplanır. Sonra o tweet için olumlu (sonuç pozitifse), olumsuz (sonuç negatifse) veya nötr (sonuç sıfırsa) kararına varılır ve sınıflandırılır. Bu işlem eldeki tüm tweet verilerine uygulanır. Daha sonra bunlar oranlanarak tweet ifadelerinin mesela %67'si olumlu gibi bir sonuca varılabilir ki buda kampanyanın marka üzerine etkisini daha önce yapılan rutin analizlerle kıyaslayarak en hızlı bir şekilde ölçmek için önemli bir fikir verir. Elbette bu, konuyu en basit şekilde aktarabilmek için verilen yüzeysel bir yaklaşım ama daha sofistike algoritmalar benzer mantıkla uygulanabilmektedir [26].

Metin Madenciliği teknikleri dört temel kategoriye ayrılır: sınıflandırma (classification), birliktelik analizi (association analysis), bilgi çıkarım (information extraction) ve kümeleme (clustering) [29]. Sınıflandırma işlemi nesnelerin daha önceden bilinen sınıflara ya da kategorilere dahil edilmesidir. Birliktelik analizi ise sıklıkla birlikte yer alan ya da gelişen sözcük ya da kavramların belirlenmesi ve böylece doküman içeriğinin ya da doküman kümelerinin anlaşılmasını amaçlamaktadır. Bilgi çıkarım teknikleri ile dokümanların içerisindeki yararlı veri ya da ifadeler bulunmaya çalışılmaktadır. Kümeleme analizi, doküman kümelerinin temelini oluşturan yapıların keşfedilmesi amacıyla uygulanmaktadır. [30]

Metin Madenciliği, doküman koleksiyonları üzerinde odaklanmaktadır. En basit tanımıyla bir doküman koleksiyonu, metin tabanlı çeşitli dokümanların oluşturduğu topluluktur. Pratikte metin madenciliği çözümlerinin amacı çok geniş doküman koleksiyonları içerisindeki gizli bilgileri ve desenleri keşfetmektir. Bir doküman gerçek hayatta kullanılan rapor, e-posta metni, makale, araştırma bildirisi, haber özeti gibi nesnelerle ilişkilendirilebilir [29].

2.3. Metin Kümelemede Karşılaşılan Sorunlar

Doküman koleksiyonları statik veya dinamik olabilir [31, 32]. Statik koleksiyonların içeriği zamanla değişim göstermez. Dinamik koleksiyonlarda ise zaman içerisinde yeni dokümanlar eklenmesi ya da doküman güncellenmesi sonucunda değişim olur. İçeriği çok hızlı değişen çok büyük doküman koleksiyonları üzerinde metin madenciliği uygulamalarında performans sorunları yaşanabilir [31]. Ayrıca doküman koleksiyonlarındaki çok boyutluluk sorunu önişleme (preprocessing) tekniklerinin geliştirilmesini de zorunlu kılmıştır [29].

Metin dökümanları doğal dil ile yazıldıklarından yapısal veriden çok farklıdırlar. Bundan dolayı yapısal veriler için geliştirilmiş kümeleme algoritmaları metin dökümanları için yeterince başarılı olamamaktadır [33]. Gerçek hayatta aynı anlamı ifade etmek için farklı sözcükler kullanılabileceği gibi aynı sözcük farklı anlamları ifade etmek için de kullanılabilmektedir. Metin kümelemede anlamların göz ardı edilerek sadece sözcüklerin kullanılması hata oranını artırmaktadır[34]. Gerçek hayatta değişik konulardaki dökümanların sözcük dağarcığı

birbirinden çok farklı olmasına rağmen Vektör Uzayı Modeli'nde tüm döküman vektörleri tüm koleksiyondaki sözcükleri içerecek şekilde normalize edilmek zorunda kalınmaktadır [405839]. Genellikle dökümanlar 200-10000 arası eşsiz sözcük yani boyut içerirler. Geniş döküman koleksiyonlarını etkin ve verimli bir biçimde kümelemek için çok yüksek olan boyut sayısının indirgenmesi gerekmektedir [35]. Bir dökümanın içerdiği bir ya da daha fazla konu nedeniyle aynı anda birden fazla kümeye dahil olabilmesine yani döküman kümelerinin belli ölçüde örtüşmesine izin verilmelidir [26].

Kümeleme işleminden önce küme sayısı bilinmemektedir. Döküman koleksiyonunun içeriği hakkında bilgi sahibi olunmadan oluşacak küme sayısını doğru tahmin etmek zordur. Kümeleme algoritmasına oluşması beklenilen küme sayısının verilmesi yerine bu sayıyı kümeleme algoritmasının bulması daha anlamlıdır. Metin kümelemede karşılaşılan bu gereksinim ve sorunlar, metin dökümanlarının doğası da hesaba katılarak yeni teknik ve algoritmaların geliştirilmesini gerekli kılmaktadır [26].

2.3.1. Metin Madenciliği Sistemlerinin Genel Mimarisi

Temel olarak, bir metin madenciliği sistemi ham haldeki dokümanları girdi olarak kullanıp çeşitli türde çıktılar üreten bir sistemdir. İşlevsel olarak bir metin madenciliği sistemini kabaca 4 bölümde değerlendirmek mümkündür:

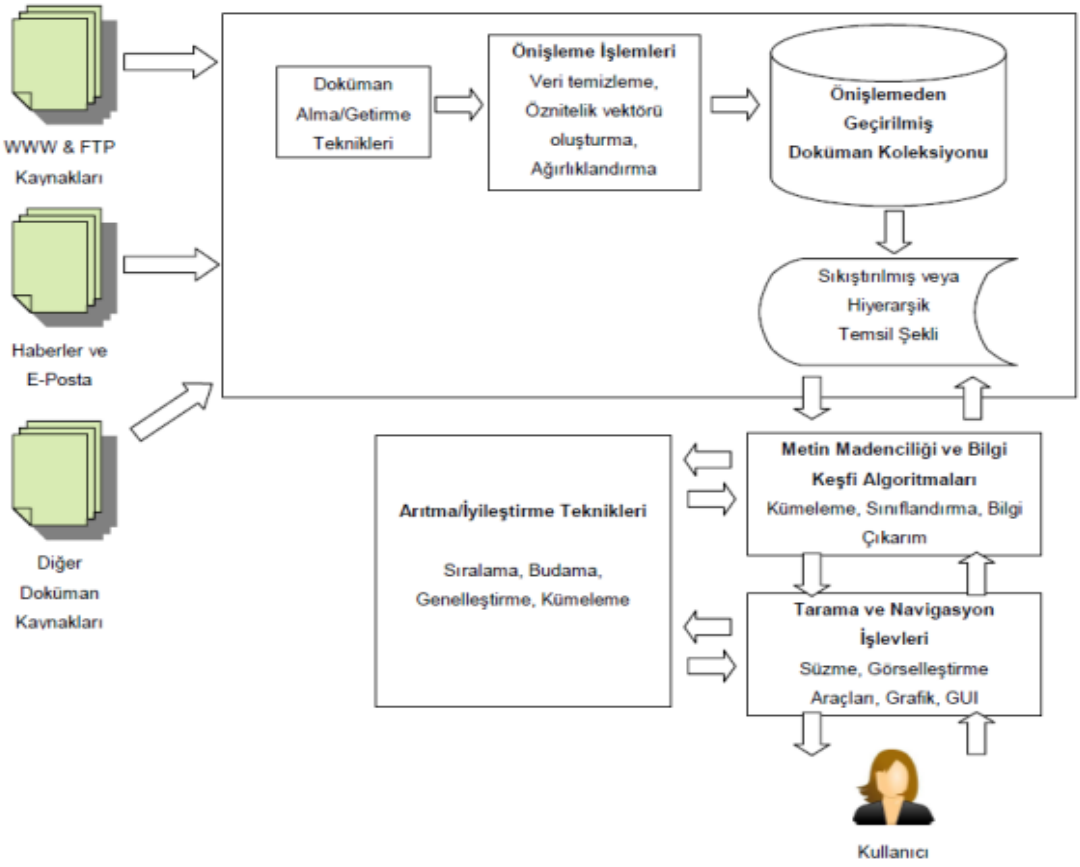
Önişleme: metin madenciliği sisteminin temel bilgi keşfi işlemleri için veri hazırlama işlemidir.

Temel madencilik işlemleri: Veri madenciliğinin kalbi durumundaki, veriden bilgi elde etme işlemleridir.

Sunum işlemleri: Görselleştirme araçları gibi çeşitli yöntem ve araçlarla sonuçların son kullanıcıya yansıtılması ve kullanıcının sistemle etkileşiminin sağlanması işlemleridir.

İyileştirme teknikleri: Sonuçların yeniden sıralanması, düzenlenmesi, filtrelenmesi gibi optimizasyon aşamasıdır. Postprocessing olarak da adlandırılır.

Şekil 2.1.'de yukarıdaki işlemlerin gerçekleşme sırası ve süreçleri görülmektedir.



Şekil 2.1. Metin madenciliği sisteminin genel mimari yapısı [29]

2.3.2. Vektör Uzay Modeli

Vektör uzay modeli bilgi çıkarımı, bilgi filtreleme, indeksleme gibi alanlarda kullanılan cebirsel bir modeldir. Dokümanlar çok boyutlu vektör uzayında temsil edilmektedirler. Vektör uzayının boyutunu dokümanlar kümesindeki ayrık terim sayısı belirlemektedir. Bu modelde vektör yapısını nesneler tanımlamaktadır. Nesnelerin sahip olduğu başka bir özellikse, vektör uzayının eksenlerini oluşturmakta ve her nesnenin sahip olduğu özelliklere göre vektör uzayında belli bir konuma sahip olmaktadır [4].

Dokümanların çok boyutlu birer vektör olduklarını düşünerek, kümeleme problemi klasik kümelemeden daha değişik işlemler gerektirmektedir. Doküman kümeleme verisi çok boyutlu, seyrek ve önemli derecede sıra dışı veri içeren bir yapıda olan kelime-doküman matrisidir. Veri matrisinin sütunları; terimleri, satırları ise dokümanları belirtmektedir. Bu matris oluşturularak kelime doküman çifti için TF-IDF değeri bulunur [36]. Bu da o kelimenin dokümandaki ağırlığını göstermektedir. Bir terim bir dokümanda diğer dokümanlara göre daha sık görünüyorsa, o dokümanın belirleyici terimidir. Bu yüzden ağırlığı yüksektir. Diğer yandan birçok dokümanda geçen terim dokümanları ayırt edici özelliğini yitirir ve terimin ağırlığını azaltır.

TF: Terim frekansıdır. Bu değer terimin ilgili dokümanda kaç defa geçtiğini gösterir. Böylece o terimin ilgili doküman için önemini gösterir. Eş.3.1'de hesaplanır [36, 37]:

$$TF_{ij} = \frac{n_{ij}}{|d_i|} \qquad n_{ij} = j.\text{terimin i. Dokümandaki sayısı}$$

$$d_i = \text{i. Dokümandaki bütün terimlerin sayısı} \qquad (2.1)$$

n_j = j. terimin göründüğü dokümanların sayısı

(sadece $TF_{ij} > 0$ olan terimler için hesaplanır)

IDF: Ters doküman frekansıdır. Terimin genel önemini gösterir. Eş.3.2'de hesaplanır [38, 39]:

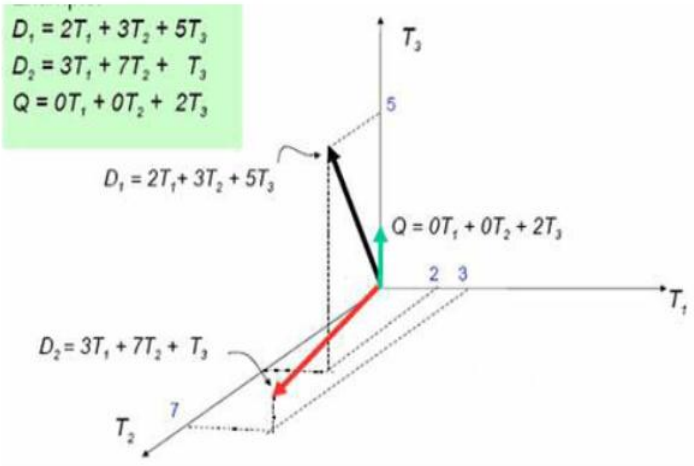
$$IDF_j = \log_2\left(\frac{n}{n_j}\right) \qquad n = \text{toplam doküman sayısı} \tag{2.2}$$

2.3.2.1. TF-IDF Ağırlıklarının Hesaplanması

$$X_{ij} = TF_{ij} * IDF_j$$

$$X_{ij} = \frac{n_{ij}}{|d_i|} \times \log_2\left(\frac{n}{n_j}\right) \tag{2.3}$$

Veri matrisini TF-IDF değerlerini hesaplayarak bulunur. Lakin bu matrisi bu şekilde kullanırsak çok büyük bir veri matrisi elde edildiğinden bellek yeterli olmayacaktır. Veri matrisinin sütunlarında bulunan bir kelime için, kelimenin bulunmadığı dokümanlardaki TF değeri sıfır olduğundan, TF*IDF değeri de sıfır olacaktır. Her dokümanda belli sayıda terim olacağı düşünüldüğünde ortaya çıkan matrisin büyük bir kısmını "0" değeri dolduracaktır. Sıfırlar çıkarılarak veri matrisi indirgenir. Bu şekilde sıfır değerleri için gereksiz bellek kullanımı engellenerek, bellek problemi çözülecektir. Benzerlik hesaplamaları gerçekleştirilirken işlem yapılacak dokümanın her satırı bir vektöre alınacaktır. O dokümanda bulunmayan terimler için "0" değeri verilerek geçici bir süre olması gereken boyuta getirilir. Bu işlemler sırayla her doküman için gerçekleşir [4].



Şekil 2.2. Doküman uzayında vektörlerin gösterilmesi [4]

Şekil 2.2.'de görüldüğü gibi dokümanlar kelimelerin vektörleri olarak ifade edilirler. T'ler aslında kelimeleri ifade etmektedirler. Anahtar kelime araması yapılan dokümanların ilişki seviyeleri doküman benzerlik teorisindeki varsayımlar kullanılarak, yani her bir doküman vektörü ile orijinal sorgu vektörü arasındaki açıların sapmalarını karşılaştırarak, hesaplanabilir [4].

2.3.3. Vektör Benzerlik Ölçütleri

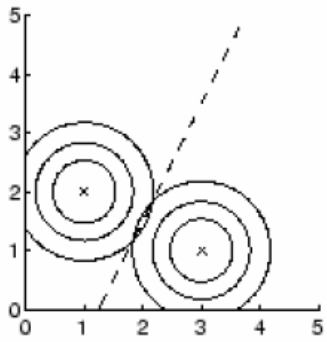
Kümeleme yöntemlerinin birçoğu, gözlem değerleri arasındaki uzaklıklarının veya benzediklerini hesaplanmasına dayanmaktadır. Bu noktada benzerlik ölçütü kavramına açıklık getirmektedir. Benzerlik ölçütü birbirinden farklı olan veri çiftlerin birbirine ne kadar benzer olduğunu tespit etmeye çalışır. Benzerlilik ölçütleri yaparak bir veriyi diğer verilerden ayırmamız mümkün olmakta ve veri kümesi üzerinde kümeleme yapmak mümkün hale gelmektedir [4].

2.3.3.1. Öklit Uzaklık Ölçüsü

Öklit uzaklık ölçüsü, iki birim arasındaki uzaklık Denklem 3.4'e göre hesaplanır.

$$d(i,j)=\sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+(x_{i3}-x_{j3})^2+\cdots+(x_p-x_p)^2} \quad (2.4)$$

Vektör olarak alındığında iki vektör arasındaki Öklit uzaklığı vektörlerin her elemanın farkının karelerinin toplamının karekökü alınılarak hesaplanır. Şekil 2.3.'de görüldüğü gibi Öklit uzaklığı kullanılarak bulunan kümeler küresel bir yapıya sahiptir.



Şekil 2.3. Öklit uzaklığının kümeleme özelliği [316607-29]

Örnek olarak V_1 vektörünü V_2 ve V_3 vektörü ile olan uzaklığı aşağıdaki gibi hesaplanır.

$V_1=(1,1,1,1)$

$V_2=(0,0,1,1)$

$V_3=(1,1,1,0)$

$d(V_1,V_2)=[(1-0)^2+(1-0)^2+(1-1)^2+(1-1)^2]^{½}=1.41$

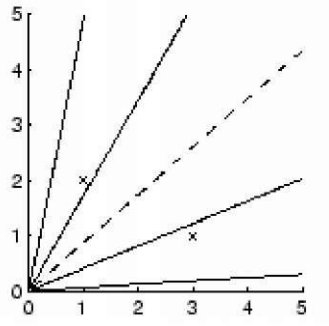
$d(V_1,V_3)=[(1-1)^2+(1-1)^2+(1-1)^2+(1-0)]^{½}=1$

V_1 vekörünün V_3 vektörü ile olan uzaklığı, V_2 vektörüyle olan uzaklığından daha küçük olduğu için V_1 vektörü V_3 vektörüne V_2 vektöründen daha çok benzemektedir.

2.3.3.2. Kosinüs Benzerliği

Kosinüs benzerliği iki vektör arasındaki kosinüs uzaklığını hesaplayarak vektörlerin birbirleriyle ne kadar benzer olduğunu ölçmek için kullanılmaktadır [40]. Kosinüs benzerliğinde vektörler

arasındaki açının değeri bulunarak benzerlik hesaplanabilir. Açının değeri küçüldükçe vektörlerin daha benzer oldukları anlamına gelir. Şekil 2.4.'te kosinüs benzerliğinin kümeleme



Şekil 2.4. Kosinüs benzerliğinin kümeleme özelliği [37]

özelliği görülmektedir. A vektörünün B vektörü ile benzerliği Denklem 2.5'de görüldüğü gibi bulunur. Burada *A.B* iç çarpımı, |A| ve |B| ise vektör uzunluğunu ifade etmektedir.

$$\text{Benzerlik}(A,B)= \cos(\alpha)= \frac{A.B}{|A|.|B|}=\frac{\sum_{i=0}^{n} A_i.B_i}{\sqrt{\sum_{i=0}^{n} A_i^2}\sqrt{\sum_{i=0}^{n} B_i^2}} \quad (2.5)$$

Buna dayanarak V_1 vekörünün V_2 ve V_3 vektörü olan benzerliği aşağıdaki gibi bulunur:

$V_1=(1,1,1,1)$

$V_2=(0,0,1,1)$

$V_3=(1,1,1,0)$

$V_1.V_2= 1\times0 + 1\times0 + 1\times1 +1\times1= 2$

$|V_1|=(1\times1+1\times1+1\times1+1\times1)^{1/2}=2$

$|V_2|=(0\times0+0\times0+1\times1+1\times1)^{1/2}=1.41$

$\text{Benzerlik}(V_1,V_2)=2 /(2^{x}1.41)=1.41$

$V_1. V_3 = 1+1^{x}1+1^{x}1+1^{x}0=3$

$|V_1|=(1\times1+1\times1+1\times1+1\times1)^{1/2}=2$

$|V_3|=(1^{x}1+1^{x}1+1^{x}1+0^{x}0)^{1/2}=1.73$

$\text{Benzerlik}(V_1,V_3)=3/(2^{x}1.73)=1.15$

Benzerlik (V_1,V_2) değeri (V_1,,V_3) değerinden büyük olduğundan V_1 vektörü V_2 vektörüne V_3 vektöründen daha çok benzemektedir.

2.3.3.3. Pearson Uzaklık Ölçüsü

Pearson ilişkisinde kosinüs benzerliğindeki gibi vektörler arasındaki açıya bakılarak iki vektörün benzerliği hesaplanır. Kosinüs benzerliğinden farkı, iki vektörün iç çarpımı yapılmadan önce her birinin farklı ortalama değerleri bulunur ve her ortalama değer ait olduğu vektörün tüm elemanlarından çıkarılır [4].

Pearson uzaklık ölçüsü kullanılarak iki nokta arasındaki benzerlik aşağıdaki eşitlikle hesaplanır.

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})/S_1^2 + (x_{i2} - x_{j2})^2/S_1^2 + \cdots + (x_{ip} - x_{jp})^2/S_p^2} \quad (2.6)$$

Denklem'de kullanılan S_p, uzaklığın hesaplandığı değişkene ait varyanttır. Bununla beraber farklı gruplar hakkında önceden bilgisi olunmadığı için, benzerlik hesaplanmasında *S* değerinin kullanılması doğru değildir. Bu sebeple Pearson uzaklık ölçüsü yerine genellikle Öklit uzaklık ölçütü daha uygun görülür. Kümeleme analizinde kullanılacak değişkenler belirli önem derecelerine göre ağırlandırılmışsalar, Pearson uzaklık ölçüsü eşitliği aşağıdaki gibi olur [4]:

$$d(i,j) = \sqrt{w_1(x_{i1} - x_{j1})^2/S_1^2 + w_2(x_{i2} - x_{j2})^2/S_1^2 + \cdots + w_p(x_{ip} - x_{jp})^2/s_p^2} \quad (2.7)$$

2.3.3.4. Manhattan Uzaklık Ölçüsü

Diğer bir uzaklık ölçüsü *Manhattan uzaklığıdır.* Manhattan ölçüsü iki vektörün toplamıdır.

Manhattan uzaklık ölçüsünde iki birim arasındaki uzaklık aşağıdaki Denklem 2.8 ile hesaplanır:

$$d(i,j)=(|x_{i1} - x_{j1|}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|) \quad (2.8)$$

Buna dayanarak V_1 vekörünün V_2 ve V_3 vektörü olan benzerliği aşağıdaki gibi bulunur:

$V_1=(1,1,1,1)$

$V_2=(0,0,1,1)$

$V_3=(1,1,1,0)$

$d(V_1, V_2)= |1-0| + |1-0| + |1-1| + |1-1| =2$

$d(V_1, V_3)= |0-1| + |0-1| + |1-1| + |1-0| =3$

Benzerlik **(V_1,V_3)** değeri **(V_1,V_2)** değerinden büyük olduğundan V_1 vektörü V_3 vektörüne V_2 vektöründen daha çok benzemektedir.

2.3.3.5. Minkowski Uzaklık Ölçüsü

P sayıda değişken göz önünde alınarak değerleri arasındaki uzaklığın hesaplanması için kullanılır. Minkowski uzaklık ölçüsü kullanılarak iki birim arasındaki uzaklık aşağıdaki Denklem 2.9 ile hesaplanır:

$$d(i,j)=[\sum_{k=1}^{p}(|x_{ik}-x_{jk}|)^{m}]^{\frac{1}{m}} \qquad i,j=1,2\ldots n;\ k=1,2\ldots p \tag{2.9}$$

Burada m=2 yazılarak Öklit uzaklığı elde edilir.

2.3.4. Performans Ölçütleri

Dokümanların kümelemesinde kullanılan performans ölçütleri dağıntı (entropy), saflık (purity) ve F ölçütü (F measure). Bu ölçütler kümelemenin sonucuna uygulanır, bunların uygulanması için tüm dokümanların etiketlenmesi gerekir. Aynı kümedeki dokümanlar benzer etiket numarasını alırlar. Örneğin; Milliyet veri setinde siyaset klasörünün altında bulunan 20 adet doküman "0" etiketine, sağlık klasörünün altında bulunan 20 adet doküman "1" etiketine ve futbol klasörünün altında bulunan 20 adet doküman "2" etiketine sahiptir. Performans ölçütlerinin hesaplanması için kümelerdeki dokümanların hangi etiket numaralarına sahip olduklarını kümeleme işleminin sonucunda belirlenmesi gerekir. Bu etiketleri belirleyerek hangi kümede hangi kategorilerden kaçar belge olduğu tespit edilecektir ve bir karmaşıklık matrisi (confusion matrix) oluşturulur. Dağıntı, saflık ve F ölçütü bu matrisi kullanarak hesaplanır [4].

2.3.4.1. Dağıntı

Dağıntı, tüm dağılımdaki düzensizliklerle ilgilenir. Her bir sınıfa ait belgelerin bir küme içerisinde nasıl dağıldığına bakar. Dağıntı bize bir kümenin ne kadar homojen olduğunu söyler. Bir kümenin homojenliği ne kadar yüksekse, dağıntı yada belirsizlik o kadar düşük olur. Bir cisimden (mükemmel homojenlik) oluşan bir kümenin dağıntısı sıfırdır.

Kümelemedeki her *j* kümesi için elde ettiğimiz C sonucu, p_{ij} ,'dir ve *j* kümesinin bir üyesi *i* sınıfına ait olabilir. Her bir *j* kümesinin dağıntısı, toplamın tüm sınıflardan alındığı standart formüller kullanılarak hesaplanır $E_j = -\sum_i p_{ij} \log(p_{ij})$, küme grubu için toplam dağıntıdır. Her bir kümenin boyutuyla ağırlıklı her bir küme dağıntılarının toplamı olarak hesaplanır:

$$E_c = \sum_{j=1}^{Nc}\left(\frac{N_j}{N} \times E_j\right) \tag{2.10}$$

Burada Nj, *j* kümesinin boyutudur ve *N*, veri nesnelerinin toplam sayısıdır [41],

2.3.4.2. Saflık

Saflık, her bir kümenin başlıca bir sınıftan belgeleri içinde bulundurma kapsamını ölçer. Belirli bir n_j boyutlu j kümesi için, bu kümenin saflığı tanımlanır:

$$P_j = \frac{1}{n_j} max_i\ n_{ji}, \tag{2.11}$$

Burada n_{ij}, *i* sınıfının *j* kümesine ayrılan belgelerinin sayısıdır. Böylece P_j , bir kümeye tahsis edilen en büyük belge sınıfının oluşturulduğu genel küme boyutunun bölümüdür. Kümeleme çözümünün genel saflığı, her bir bireysel küme saflıklarının ağırlıklı toplamıyla elde edilir.

$$P=\sum_j \frac{n_j}{n} p_j \qquad (2.12)$$

Burada n, belge yığınındaki belgelerin toplam sayısıdır. Genel olarak, saflık değerleri ne kadar büyük olursa, kümeleme çözümü o kadar iyi olur.

2.3.4.3. F-Ölçütü

Diğer dış nitelik ölçümü ise F ölçütüdür [42]. Bu hassasiyet ile bilginin geri kazanılmasıyla elde edilen anımsama (geri çağırma) fikirlerini birleştiren bir ölçüttür [43, 44]. Her bir kümeye, bir sorgulamanın sonucuymuş ve her bir sınıfa da bir sorgulama için istenen belgeler grubuymuş gibi kümeleme yapılır. Daha sonra, o kümenin verilen her bir sınıf için anımsamasını ve hassasiyeti hesaplanır. Daha spesifik olarak da *j* kümesi ile *i* sınıfı içindir.

Anımsama(i, j) = *n*ij / *n*i

Hassasiyet(i, j) = *n*ij / *n*j

Burada n_{ij},*j* kümesindeki i sınıfının üyelerinin sayısıdır, n_j,*j* kümesinin üyelerinin sayısıdır ve n_i, de *i* sınıfının üyelerinin sayısıdır.

F(i, j) = (2 * Recall(i, j) * Precision(i, j)) / ((Precision(i, j) + Recall(i, j)) (2.13)

Bir hiyerarşik kümelemenin bütünü için bir sınıfın F ölçütü, ağaçtaki herhangi bir düğümde elde ettiği maksimum değerdir ve F ölçütü için toplam değer, aşağıda verildiği gibi F ölçümü için tüm değerlerin ağırlıklı ortalaması alınarak hesaplanır.

$$F=\sum_i \frac{n_i}{n} \max\{F(i,j)\} \qquad (2.14)$$

Burada maksimum, tüm seviyelerdeki kümelerin hepsinden elde edilir ve n, belgelerin sayısıdır.

2.4. Metin Madenciliğinde Ön İşleme (Pre-Processing) Aşaması

Bir metin içerisindeki sözcükleri elde etmek için genellikle dizgeciklere (token) ayırma (tokenization) işlemi gereklidir. Bu işlem ile metin içerisindeki tüm noktalama işaretleri, tab ve satır sonu karakterleri ile diğer okunabilir olmayan (nontext ve non-readable) karakterler boşlukla (white space) değiştirilir ve metin bir sonraki süreç için daha uygun ve temiz bir hale

getirilir. Koleksiyondaki tüm dökümanlarda dizgeciklere ayırma işlemi uygulandıktan sonra tüm dökümanlarda yer alan sözcüklerin tümü ilgili koleksiyonun "sözlüğü"nü (dictionary) oluşturur [35].

Sözlük boyutunun, dolayısıyla da koleksiyonlardaki dökümanları temsil eden veri yapılarının (örneğin Vektör Uzayı Modelindeki döküman vektörlerinin) boyutunun küçültülmesi için çeşitli önişleme yöntemleri kullanılabilir [35].

Veri ön işleme tüm metin sınıflandırma algoritmaları için ilk adımdır. Veri ön işleme aşaması aşağıda belirtilen nedenlerden dolayı gerçekleştirilir.

1. Veri üzerinde bulunabilen problemleri çözmek,
2. Verinin doğal yapısını öğrenerek daha anlamlı ve kaliteli analiz yapabilmek,
3. Veriden daha anlamlı bilgi üretebilmek [12].

Türkçe dilinin yapısı gereği ön işleme zor olmaktadır. Çünkü Türkçe eklemeli bir dildir ve bir kelimeye eklenen her bir ek o kelimenin anlamını daha da genişletmekte hatta değiştirmektedir. Üstelik, sadece bir tek Türkçe kelimeden çok miktarda değişik anlamlı kelimeler oluşturulabilir. Bu karmaşık yapı sebebiyle, Türkçe; İngilizce'den ve benzeri dillerden daha farklı metin işleme teknikleri gerektirir. Bu nedenle, bütün kelimelerin küçük harfe çevrilmesi ve noktalama işaretlerinin kaldırılması dışında; joker kelimeler ile anahtar kelimelerin oluşturulması gibi bazı ön hazırlıklar yapılması gerekmektedir [2].

Metin dosyalarının kategorilerini bulabilmek için dokümanları kelime vektörleri olarak göstereceğiz. Vektörümüzün boyutu daha önceden oluşturduğumuz sözlüğümüzün boyutuna eşit olacak, ağırlıkları ise çeşitli algoritmalarla ifade edilecektir (Binary, frekans, tf-idf gibi).

2.4.1. 3.1 Ön İşleme Genel Adımları

Metinler doğal yazılışları ile bir kelime vektörü olarak ifade edilmemişlerdir. Bu bakımdan bir çok zorluk bulunmaktadır. Örneğin dokümanlarda bir çok kelime bulunmakta; bir çok doküman bulunmakta; dokümanlarda çok çeşitlilikte bilgi yer almakta; insanlar tarafından yazıldığı için bir çok hata içermekte; noktalama işaretleri, kısaltmalar bulunmaktadır. Bu yüzden ön işleme adımı etkili bir sınıflandırma için kaçınılmaz bir adımdır. Genel adımlar aşağıdaki gibidir [2]:

Kategoriler belirlenir. Bu kategoriler ile ilişkilendirilebilecek olan kelimeler sözlüğe eklenir (Bu tezde Spor, Ekonomi, Politika ve Sağlık kategorilerini kullanacağız).

Sözlüğümüzde kategori belirten kelime sayısı yaklaşık olarak 412 tanedir. Sözlükteki her kelime teker teker incelenir. Joker (Wild Card) olarak kullanılabilecek olan kelimeler bulunup sözlük güncellenir.

Örnek: Gözlükler Gözlükte Gözlüğü

Gözlü* ("Gözlü" ifadesinden sonra ne gelirse gelsin kabul et, "Gözlük" kelimesi olarak değerlendir).

Her bir doküman, sözlükte oluşan tüm bu kelimelerin, (joker kelimeler de dahil) boyutundaki vektörün ağırlıklandırılması ile gösterilir [2].

2.4.1.1. Joker (Wild Card) Yöntemi

Sistemde metinlerdeki kelimelerin kendileri yerine gövdelerinin kullanıldığı daha önceden belirtilmişti. Bunun sebebi Türkçe gibi eklemeli dillerde bir gövdenin sonuna birçok farklı ek alarak farklı biçimlerde karşımıza çıkabilmesidir. Örneğin "araba" kelimesi ile "arabadan", "arabayı", "arabada", ve "arabanın" kelimeleri eğer ayrıştırıcı olmasa ayrı ayrı kelimeler olarak görüleceklerdi. Bunun sonucu olarak hem oluşturulan sözlük boyutu çok artacak hemde sınıflandırma başarısı düşecekti (Amasyalı ve Yıldırım, 2004).

Joker kelime, aynı söz dizimi ile başlayan ve çeşitli ekler almış ancak yakın anlamda olan sözcükleri tek bir gösterimle grup altında toplayan kelimelerdir. Joker kelime gövdeleme yöntemine benzemektedir. Gövdelemede çekim ve yapım eklerinden ayrıştırılan kelimeler, ortak bir köke indirgenir. Ancak burada köke indirgeme şartı yoktur. Kökün yanında ek de kalabilir [12]. Joker kelimeler kategoriyi belirlememize yardımcı anahtar kelimelerden veya sık kullanılan kelimelerden seçilir [2].

Joker yöntemi kelimlerin ilişkili terimlerinin anlamlarını kapsaması açısından değiştirilmesidir. Örneğin jokerli bir kelime olarak "deprem*" ile (Joker olduğunu * işaretinden anlıyoruz), "deprem" kelimesinden sonra nasıl bir ek gelirse gelsin deprem kelimesi vurgulanmış olacaktır.

Örnek: simitçi ve simitçiler

Dokümanımızda "simitçi" ve "simitçiler" kelimelerinin geçtiğini düşünürsek her iki kelimeyi ayrı ayrı sözlükte tanımlamak sözlük boyutunu arttırarak performansımızı düşürecektir. Bu yüzden bu kelimeler anlamı karşılayacak bir gövdeye indirgenebilir. Yani sözlüğümüzde "simitçi*" kullanıp doküman içerisinde başı "simit" ile başlayan tüm kelimeleri "simitçi*" olarak değerlendirebiliriz [2].

Türkçe kelimeler genellikle sert sessiz harf ile biter. Sert sessizlerin yumuşaması olabileceğinden, bu tür kelimelerde hem kelimenin sert sessizli hem de yumuşak sessizli hali joker olarak seçilir (ilaç*,ilac*). Böyle bir durumda "ila*" joker olarak seçilmemeli ("ilahiyat" gibi ilgisiz kelimeleri de içerebilir diye), ancak böyle ilgisiz kelime oluşma olasılığı yoksa o zaman her iki kelimeyi de (sert, yumuşak) içeren joker seçilebilir.

(Ör:kitap* ve kitab* yerine kita*)

Not: Sert sessizler: Ç,F,T,H,S,K,P,Ş

Son hecesinde "ı" veya "i" içeren kelimelere, sesli ile başlayan bir ek geldiğinde bu "ı" ve "i" düşer. Bu durumdaki kelimeler için ya kelimenin her iki hali de, ya da en çok görülen hali joker olarak seçilir.

Yapım ekleri kelimenin anlamını değiştiren ve çekim ekleri değiştirmeyen ekler olduğundan; joker seçiminde yapım ekleri bize zorluk çıkarır. Gövdelemede ise işlem basittir, sadece köke indirgeme yapılır. Ancak joker yönteminde, yapım eki eklenerek anlamı değiştirilmiş kelimeler, farklı kategorilerde olabileceğinden bunları ayrı ayrı göstermek gerekir [2].

Evlen*, evcil*

"Ev*" seçemeyiz. Çünkü "evren", "evrim" gibi alakasız kelimeleri içerebilir.

Çekim ekleri olan sözcükler bizim için kolaydır. Çünkü bu ekler, eklendikleri kelimenin anlamını değiştirmezler (borsada,borsalar,... -> borsa* seçilebilir).

Ancak bazen, jokerlerin alakasız kelimeleri içerebileceği durumlarda, çekim eki hallerini tek tek alamayız (kampı,kampında,....->kamp* yapamayız. Çünkü "kampanya" kelimesini içerebilir).

Zaman ekleri de bir çeşit çekim ekidir. Bu nedenle yalnızca kelimenin kökünü joker almak bazı durumlarda yeterli olmuyorken; bazen de -yor ekinde olduğu gibi (Örnek:isti-yor) kelimenin kökü deforme olabilir (iste* seçilemez).

2.4.1.2. Veri Filtreleme ve Vektörün Ağırlıklandırılması

Etkili bir ön işleme yapabilmek için haber metinlerinden noktalama işaretlerini çıkarmak önemli bir ihtiyaçtır. Ayrıca aşağıdaki gibi tüm büyük harfler küçük harfe çevirilir [2].

Taraftarlar İstanbul'da bırakıldı.

Sonuç: taraftarlar istanbul bırakıldı

Örnek: trafik kazası: 1 ölü...

Sonuç: trafik kazası 1 ölü

Noktalama işaretleri çıkarılmış dokümanda geçen tüm kelimeler bir diziye atılır. Dizi: (Taraftarlar,İstanbul,bırakıldı)

Dizideki elemanlar sözlükteki elemanlar ile karşılaştırılır. Bu şekilde vektörün elemanlarının ağırlıkları belirlenir. Örneğin sözlüğümüzün aşağıdaki kelimelerden oluştuğunu var sayarsak;

taraftar*

İstanbul*

Gol*

Futbolcu*

Vektörümüz: (1,1,0,0) şeklinde oluşur. Gerek eğitim dokümanlarının ağırlıklandırılmasında gerekse sınıflandırılacak dokümanlarım ağırlıklandırılmasında, vektörlerinin oluşturulmasında bu yöntem uygulanır. Ancak ağırlıklandırma yöntemi değişebilir. Bu tezimizde 3 farklı ağırlıklandırma yöntemini kullanacağız (bit, frekans, tf-idf).

2.4.1.3. Kelime Değerleri

Doküman, kelimeler ve onların ağırlıkları ile gösterilir. Ağırlıklandırma ne kadar iyi yapılırsa, kategorizasyonda o kadar başarılı olur. Ağırlıklandırma önemli bir konu olduğundan, çeşitli teknikler geliştirilmiştir [2].

-Terim Frekansı (TF),

-Ters Doküman Frekansı (IDF),

-Terim Frekansı-Ters Doküman Frekansı (TF-IDF),

-Terim Ayrıştırma Değeri,

-Olasılıksal Terim Ağırlıklandırma,

-Tek Terim Doğruluğu,

-Genetik Algoritmalar.

Aşağıda algoritmanın kolay anlaşılabilmesi bakımından küçük boyutlu bir sözlük kullanılarak ön işleme aşamasına örnek verilmiştir.

Kelimeler(Sözlük) Jokerleri

enflasyon*

grip* grib*,

hakem*

İlaç* ilac*,

taraftar* tarım*

Örnek eğitim dokümanları aşağıdaki gibi olursa;

Gribe yakalanan hasta grip olduğunu anlamamıştı. İlacını almamıştı. (Sağlık)

İlacını aksatanlar hastalığa davetiye çıkarırlar. (Sağlık)

Yıllık enflasyon oranı bu senede yükselişte. (Ekonomi)

Tarımla uğraşanlar bu yıl tarımdan zarar edecekler. (Ekonomi)

Hakemin gözü önünde olmasına rağmen hakem penaltı çalmadı. (Spor)

6-Taraftarlara erken gelen gol ilaç gibi geldi ve taraftarlar golden sonra hiç susmadı. (Spor)

Vektörler oluşturulurken sözlükteki kelimeler sırası ile dokümanlarda aranacak, aynı zamanda kelimenin jokeri de varsa jokeri de aranacaktır. Yani "grip*" kelimesi ile birlikte "grib*" kelimesi de dokümanda aranmalıdır. Bulunan kelime sayısı boyuta eklenmelidir.

Kategorisini bulmak istediğimiz metin aşağıdaki gibi olursa:

SORGU-Taraftarlar hakeme tepki gösterdiler. Hakem sahayı terk etti.

Sözlük={enflasyon*, grip*, hakem*, İlaç*, taraftar*, tarım*} Eğer vektörlerimizi kelime frekansına göre ifade edecek olursak;

D1=(0,2,0,1,0,0)
D2=(0,0,0,1,0,0)
D3=(1,0,0,0,0,0)
D4=(0,0,0,0,0,2)
D5=(0,0,2,0,0,0)
D6=(0,0,0,1,2,0)

DQ=(0,0,2,0,1,0)

Eğer vektörlerimizi bitsel olarak ifade etmek istersek;

D1=(0,1,0,1,0,0)
D2=(0,0,0,1,0,0)
D3=(1,0,0,0,0,0)
D4=(0,0,0,0,0,1)
D5=(0,0,1,0,0,0)
D6=(0,0,0,1,1,0)

DQ=(0,0,1,0,1,0)

2.5. Metin Sınıflandırma

Metin sınıflandırma, önceden belirlenmiş sınıflara doküman atamayı hedefler (316285, Mitchell, 1997). Sınıflandırma yapılmadan önce sınıfların belirlenmesi gerekir. Dokümanların ağırlıklandırılmış değerli vektörel ifadeleri kullanılarak elde edilen benzerlik ölçüm sonuçlarına ve uygulanan algoritmaya göre sınıflandırılması gerçekleştirilir. Metin sınıflandırma, doğal dil metinleriyle çalışan bir sınıflandırmadır [11].

Metin dokümanlarının uzunluğu metin madenciliği çalışmalarındaki en büyük sıkıntıdır. Dokümanların ön işlemden geçirilmesi ve özellik seçimi uygulanması boyutu azaltarak bu sıkıntıyı gideren işlemlerdir[9].

Makine öğrenmesine ihtiyaç duyulmasının nedeni, el ile kategorizasyonun pahalı ve zaman tüketen bir iş oluşudur ki, ayrıca elle sınıflandırmada, sınıflandırmayı yapan uzmanların vermiş oldukları kararlara bağlı olarak sonuçlar da değişmektedir (316285, İlhan, 2001). Bu sebeple otomatik metotlar, algoritmalar ve büyük miktarlardaki verilerle çalışan araçlar önemli bir hale gelmiştir [45].

Metin sınıflandırma işlemlerinde eğitim dokümanları vardır. Sınıflandırma yapılırken bu eğitim dokümanları kullanılarak sonuca gidilir. Eğitim dokümanlarının sınıfları, sınıflandırma işlemlerinin karar vermesine yardımcı olur ve özellik seçimindeki sözcükler bu dokümanlardan seçilir.

Kullanılan eğitim doküman sayısının azlığı [46], haber metinlerinin kısalığı, her haberin farklı konulardan bahsetmesi sınıfı yansıtan sözcüklerin tespit edilememesine neden olur ve sınıflandırma başarısını düşürür.

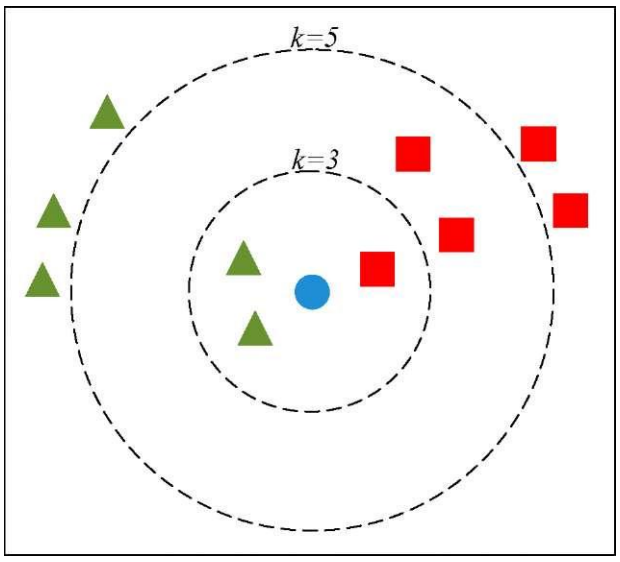
Metin sınıflandırma işleminde k-NN, Naive Bayes, SVM ve yapay sinir ağları en çok kullanılan sınıflandırma yöntemleridir.

2.5.1. k-NN Algoritması

k-NN algoritması ile sınıflandırma, önceden belirlenmiş *k* değerine göre uzaklıkları hesaplanmış eğitim dokümanları içerisinden en yakın *k* dokümandaki en yüksek frekansa sahip sınıfa göre test dokümanının sınıfını belirleme işlemidir [3, 47]. Bütün eğitim dokümanları ile

test dokümanlarının uzaklıkları tek tek hesaplanır ve belirlenecek *k* değerine göre sınıflandırma sonucuna karar verilir.

Şekil 2.5.'te görüldüğü gibi *k=3* alındığında dokümanın sınıfı üçgen olacakken *k=5* alındığında kare olmuştur. *k* değerinin yüksek seçilmesi benzemeyen dokümanların işleme dahil edilmesine, düşük seçilmesi benzeyenlerin dahil edilmemesine neden olur [9].



Şekil 2.5. *k=3* ve *k=5* değerleri için k-NN sınıflandırması [316285]

Çoğunluğun seçimi ilkesine dayanan k-NN algoritmasının dezavantajı, eğitim dokümanlarındaki dengesizliktir: Bir sınıfa ait eğitim dokümanı sayısının başka bir sınıftaki doküman sayısından fazla olmasıdır [48, 49]. Bu durumda, diğerlerine göre daha fazla sayıda eğitim dokümanı bulunduran sınıftan *k* içerisine doküman girme olasılığı artmaktadır.

2.5.2. Naive Bayes

Kolay uygulanabilmesi ve başarılı sonuçlar üretmesi nedeniyle en çok tercih edilen sınıflandırma metotlarından biridir. Hesaplamalar, sınıflar düzeyinde gerçekleştirilir. Her bir sınıf için olasılık değeri hesaplanarak en yüksek olasılık değerine sahip sınıf, sınıflandırılması yapılacak dokümanın sınıfıdır.

Sınıflandırma işleminde, her bir sözcük sınıftan bağımsızdır [50]. Ayrıca sözcükler, aynı değerde olup birbirinden bağımsız olduğu kabul edilerek hesaplama yapılır.

2.5.2.1. Multi-Variate Model

Sınıflandırma işlemlerinde her sınıf ayrı olmak üzere, sınıfı oluşturan dokümanların sözcükleriyle işlem yapılır. Diğer sınıfların dokümanları ve sözcükleri hesaplama işlemlerinde dikkate alınmaz. Sözcüğün, sınıfın dokümanlarında geçip geçmediğiyle ilgilenir.

$X = \{x_1, x_2, x_3, ..., x_n\}$ sınıflandırılması yapılacak olan vektördür. Denklem 2.15'de w_i, sözcüğünün c_k sınıfında geçme olasılığı formüle edilmiştir [50]. g_s_{wi}, *wi* sözcüğünün c_k sınıfındaki dokümanlarda geçme sayısı; *d_s*, c_k sınıfındaki toplam doküman sayıdır. Denklem 2.15'de bulunan paydaki 1 ve paydadaki 5 ifadeleri bir sözcüğün sınıf içerisinde geçmemesi durumunda sonucu 0 yapmasını engellemek içindir [51]. Bütün sınıflardaki doküman sayıları eşit

ve 5 sınıf olduğu için pay 1, payda 5 verilmiştir ve bu sonucu etkilememektedir. Çeşitli sıfır olasılık sorunu gidericiler kullanılmıştır [50].

$$p(w_i|c_k) = \frac{1 + g\_s_{w_i}}{5 + d\_s} \tag{2.15}$$

Denklem 2.16'de gösterilen formül Denklem 2.17'deki gibi revize edilmiştir. *X* vektörünün c_k sınıfında olma olasılığı ve test dokümanının sınıfı olan *C(X)* formüle edilmiştir. En yüksek $p(X\backslash c_k)$ değerine sahip sınıf test dokümanının sınıfıdır [50]. wb_{wi}, *wi*, sözcüğünün bit olarak ağırlığını ifade etmektedir.

$$C(X) = \arg\max_{c_k} p(X|c_k) = \arg\max_{c_k} \prod_{i=1}^{n} p(w_i|c_k)^{wb_{w_i}} (1 - p(w_i|c_k))^{1-wb_{w_i}} \tag{2.16}$$

$$C(X) = \arg\max_{c_k} p(X|c_k) = \arg\max_{c_k} \sum_{i=1}^{n} \log(p(w_i|c_k)^{wb_{w_i}} (1 - p(w_i|c_k))^{1-wb_{w_i}}) \tag{2.17}$$

2.5.2.2. Multi-Nominal Model

$X = \{x_1, x_2, x_3, ..., x_n\}$ sınıflandırılması yapılacak olan vektördür. Denklem 2.18'de *wi,*

sözcüğünün c_k sınıfında geçme olasılığı formüle edilmiştir [50]. g_s_w , *wi* sözcüğünün c_k sınıfında geçme sıklığını; *t_s*, c_k sınıfındaki toplam sözcük sayısını; *V*, c_k sınıfındaki toplam doküman sayısını ifade eder.

$$p(w_i|c_k) = \frac{1 + g\_s_{w_i}}{V + t\_s} \tag{2.18}$$

X vektörünün c_k sınıfında olma olasılığı, her sınıfta eşit sayıda doküman olduğundan Denklem 2.19'da gösterilen formül revize edilerek Eşitlik 2.20'deki gibi formüle edilmiştir. $p(c_k)$, c_k sınıfının olasılığını; *k* ise toplam kategori sayısını ifade etmektedir. Olasılığı gösteren en yüksek değerli $p(X\backslash c_k)$, test dokümanının sınıfıdır ve *C(X)* şeklinde ifade edilir.

$$C(X) = \arg\max_{c_k} p(X|c_k) = \arg\max_{c_k} p(c_k) k! \prod_{i=1}^{n} \frac{p(w_i|c_k)^{tf_{w_i}}}{tf_{w_i}!} \tag{2.19}$$

$$C(X) = \arg\max_{c_k} p(X|c_k) = \arg\max_{c_k} \sum_{i=1}^{n} \frac{\log(p(w_i|c_k))^{tf_{w_i}}}{tf_{w_i}!} \tag{2.20}$$

2.5.3. Elektronik Posta Sistemi

Elektronik posta (e-posta) bir kullanıcının bir bilgisayar sisteminde yazdığı, onu okuyabilecek başka bir kullanıcıya bir çeşit bilgisayar ağı üzerinden ilettiği, genellikle basit bir metin mesajı şeklinde olan elektronik mesaja verilen addır. Elektronik posta mesajları mektuplara benzerler ve başlıca iki ana kısımdan oluşurlar. Üstbilgi (header) alıcının, mesajın kopyasının gönderileceği

kişinin adını ve adresini ve mesajın konusunu içerir. Gövde (body) mesajı içeren kısımdır. Bir mektubu yollarken olduğu gibi elektronik postayı gönderirken de doğru adrese ihtiyaç vardır. Eğer yanlış veya eksik bir adres yazarsanız, mesajınız size alıcıya ulaşılamadı böyle bir alıcı yok v.s gibi bir konuyla size geri döner. Bir elektronik postamesajı aldığınızda, üstbilgiler size mesajın nereden geldiğini, ne zaman ve nasıl gönderildiğini belirtir [1].

Mektupları bir zarfla kapatırız fakat elektronik posta mesajlarında böyle bir durum yoktur, elektronik posta mesajı özel değildir. Daha çok bir posta kartına benzerler. Mesajların onlara bakması gerekli olmayan kişiler tarafından yolu kesilebilir ve okunabilirler. Eğer şifreleme kullanılmazsa, gizli bilgilerin elektronik postayla gönderilmesinden kaçınılmalıdır [1].

İnternet üzerinde veya bir bilgisayar ağı içerisinde elektronik posta göndermemizi sağlayan protokol, TCP/IP protokol kümesinin en üst kısmı olan uygulama katmanındaki SMTP protokolüdür. POP3 veya IMAP gibi protokoller sayesinde ise posta kutumuza ulaşan mesajlara erişebiliriz .

2.5.3.1. Elektronik Posta İstemcileri

Elektronik postaları okumak için Microsoft Outlook, Microsoft Outlook Express, Eudora veya Pine gibi bir yazılım kullanılır. Eğer Hotmail, Yahoo veya Gmail gibi ücretsiz bir elektronik posta servisine üye olunursa, elektronik postaları okumak için bir web sayfası olarak görünen bir elektronik posta istemcisi kullanılır.

- Bütün elektronik posta istemcilerinin 4 temel işlevi vardır [1]:
- Posta kutunuzdaki mesajların bir listesini görüntülemek,
- Listeden bir mesaj seçmek ve mesajın içeriğini okumak,
- Yeni bir mesaj oluşturmak ve göndermek,
- Eklentileri işlemek - göndereceğiniz bir mesaja eklenti yerleştirmek veya alınan bir mesajdaki eklentileri kaydetmek.

2.5.3.2. Elektronik posta sunucuları

Elektronik posta göndermek ve almak için, istemcinin bağlanması ve elektronik posta mesajını teslim edilmek üzere vermesi için internet üzerinde özel bir bilgisayara ihtiyaç vardır. Bu bilgisayar elektronik posta servisi sağlamak için bir uygulama yazılımı çalıştırır ve elektronik posta sunucu olarak adlandırılır.

İnternet üzerinde Web sunucu, Ftp sunucu, Telnet sunucu, Elektronik Posta sunucu v.s olarak uygulama yazılımları çalıştıran milyonlarca bilgisayar vardır. Bu uygulamalar, sunucu makine üzerinde sürekli çalışır ve özel portları dinlerler. İnsanların veya programların bağlanması için beklerler. SMTP sunucusu giden ve gelen elektronik postaları işler. Bu sunucu elektronik posta göndermek isteyen birisi için 25 numaralı portu dinler. Elektronik posta istemcisi posta göndermek için SMTP sunucu ile bir etkileşimde bulunur.

SMTP protokolü RFC 821 ve RFC 1123 de tanımlanmıştır. 7-bit US-ASCII karakterinde düzenli bir veri akışı talep eder, gönderenin SMTP komutlarını alıcıya yayınlamasıyla diyalog

başlar (SMTP SESSION). Alıcı göndereni, ardından kod hakkında ek bilginin yer aldığı, sayısal yanıt kodlarıyla yanıtlar [1].

SMTP sunucu, HELO, MAIL, RCPT, ve DATA gibi çok basit metin komutlarını anlar. En yaygın kullanılan komutlar şunlardır [1]:

-HELO : Sunucu kendisini tanıtır.

-EHLO : Sunucu kendisini tanıtır ve genişletilmiş modu talep eder.

-MAIL FROM : Göndereni tanımlar

-RCPT TO : Alıcıyı tanımlar

-DATA : Mesajın gövdesini tanımlar.

-RSET : Bağlantıyı yeniden başlatır.

-QUIT : Oturumu sonlandırır.

Aşağıdakiler alıcı SMTP tarafından gönderilen bazı yanıt kodlarıdır.

- 211 System Status or system help reply
- 220 domain Service ready
- 221 domain Service closing transmission channel
- 250 Requested action OK and completed
- 354 Start mail input; end with .
- 421 Domain service not available, closing connection
- 450 Mailbox unavailable, requested mail action not taken

Tipik bir değişimde, gönderen alıcı ile bağlantı kurduktan sonra, alıcı hazır olduğunu gösteren 220 koduyla yanıt gönderir. Gönderen daha sonra HELO komutunu bir argüman olarak gönderir. HELO komutu göndereni alıcıya tanıtır, ve alıcı sonra 250 yanıt koduyla yanıtlar. Bu gönderene bağlantının açık olduğunu ve devam etmek için hazır olduğunu söyler. Bir sonraki adım gönderenin ve alıcının sunucu adresini tanımlar ve doğrular. Kendisini tanıttıktan sonra, elektronik posta istemcisi "from" ve "to" adreslerini belirtir ve sonra alıcının mesajı almaya hazır olup olmadığını sormak için DATA komutunu yayınlar. Alıcı gönderenin mesajı teslim edebileceğini belirten 354 koduyla yanıt verir. İletim bir satırda yalnız bir '.' ile biter. .mesaj gönderildikten sonra, gönderen "quit" komutunu yayınlar ve alıcıda 221 yanıt koduyla yanıtlar. Aslında bu diyalog bir SMTP sunucusunun 25 numaralı portuna telnet ile bağlanılarak gerçekleştirilebilir. Dolayısıyla böyle bir işlem sonunda spoof (aldatma) yapılabilmektedir [1].

2.5.3.3. İstenmeyen Elektronik Posta (Spam)

İstenmeyen elektronik posta RFC 524 esasları üzerinde elektronik posta sistemlerinde gerçekleştirilmiş, SMTP protokolünün kötüye kullanılmasının bir şeklidir. İlk olarak 1973 de önerilen RFC 524, bilgisayar güvenliğinin belirgin bir endişe yaratmadığı bir dönemde

geliştirilmiştir. RFC 524 ün güven çok güvenli bir komut seti olmaması, SMTP protokolünü kötüye kullanılmaya karşı hassas bir hale getirir [1].

Çoğu istenmeyen elektronik posta üretme aracı SMTP'deki güvenlik açıklarını kullanmaktadır. Bunu elektronik posta üstbilgilerinin sahtesini üreterek, gönderenin adresini ve gönderen sistemi gizleyerek yapıyorlar, öyle ki gerçek gönderenin kimliğini tespit etmek zor veya imkansızdır.

SMTP protokolünün bu açıkları kapatmak için geliştirilmiş protokollerin çoğu elektronik postayı kabul etmeden önce gönderenin kimliğini doğru olarak tespit edebilmek için özellikler içerir. Ancak bu protokollerin yaygınca kullanılması çok zordur, çünkü yeni protokolü gerçekleştirenler sadece bu protokolü gerçekleştiren diğer sistemlerden e-posta alabilir. Bu yüzden yakın gelecekte daha güvenli bir SMTP olmadan, istenmeyen elektronik posta, kuruluşları etkili bir istenmeyen elektronik posta engelleme çözümü arayıp bulmaya yönelten bir problem olmaya devam edecektir [1].

Analistlerin tahminleri günümüzde dünyadaki e-postaların %60'ından fazlasının istenmeyen elektronik posta olduğunu gösteriyor. İstenmeyen elektronik posta artık sadece can sıkıcı bir şey değildir. İstenmeyen elektronik posta şimdi belirgin bir güvenlik sorunu ve finansal kaynaklar üzerinde ağır bir yüktür. Aslında, bu istenmeyen elektronik posta istilasının şirketlere her sene 20 milyon dolarlık bir verimlilik kaybına sebep olduğu tahmin edilmiştir [1].

Bugün istenmeyen elektronik posta problemini engellemeye yardımcı olmak için çok sayıda çözüm vardır. Bu çözümler elektronik postanın analizinde ve onun gerçekten istenmeyen elektronik posta olup olmadığını belirlemede farklı teknikler olarak kullanmaktadır. İstenmeyen elektronik posta sürekli değiştiğinden, en etkili istenmeyen elektronik posta engelleme çözümleri bu tekniklerden bir kaçını içermelidir [1].

2.6. Literatürde Yapılan İlgili Çalışmalar

[1] nolu uygulamada belge sınıflandırma, yönlendirmeli öğrenme kavramları ve Yalın Bayes sınıflandırmanın istenmeyen elektronik posta problemine uygulanması anlatılmıştır.

[52] nolu çalışmada; 2000-2011 yılları arasındaki veri madenciliği teknik ve uygulamaları incelenmiş, yayınlanmış makaleler ve yapılan çalışmalar kaynakları ile verilmiştir.

Bir diğer çalışmada, geçmiş fabrika verileri kullanılarak, halı üretim verimliliğini arttırmak için veri madenciliği teknikleri kullanılmıştır [53].

[54] nolu çalışmada yer bilimi verilerinin veri madenciliği teknikleriyle işlenmesiyle, petrol ve gaz rezervleri hakkında bilgi sahibi olunabileceği konusunu işlemişlerdir.

[55] nolu çalışmada; giyim sektöründe endüstriyel standartların geliştirilmesi ve üretim-pazarlamanın arttırılmasına yönelik, standart dikimler yerine müşterilerin vücut ölçülerinin veri madenciliği teknikleriyle işlenerek elde edilen sonuçlar doğrultusunda dikimlerin yapılarak satışların arttırılması hedeflenmiştir.

Metin madenciliği ile metin konularının özetlenmesi ve benzer yazıların belirlenmesinin yapıldığı [56] nolu çalışmada, yüksek idf değerli sözcüklerle özetleme, Cosine benzerliğiyle benzer yazıları belirleme işlemini gerçekleştirilmiştir.

[57] nolu çalışma, günümüzde sağlık alanında metin madenciliği kullanımını öngören bir çalışma olduğunu göstermiştir. Sağlık alanında sıkça kullanılan metin madenciliğinin, gen klinik araştırmalarında kullanımının geleceği konusu işlenmiştir. Eldeki verilerin sadece yapılan deney ve gözlemlerden ibaret olmadığı, konuyla ilgili birçok bilimsel çalışma gerçekleştirildiği, bu çalışmaların metin verisi barındırdığı belirtilmiştir. Metinlerin konuyla ilgili olup olmadıklarının belirlenmesinin metin madenciliği teknikleri kullanılarak gerçekleştirilebileceği vurgulanmıştır.

[58] nolu çalışmada kandırma ve hilelerin veri ve metin madenciliği teknikleriyle bulunmasını incelemişler ve %74,00 oranında doğru tespit gerçekleştirmişlerdir.

Metin sınıflandırma, metin madenciliğinin uygulama alanlarından biridir. Metin sınıflandırma ile ilgili Türkçe dışındaki dillerde yapılan çalışmalar vardır [59-61].

[60] ve [62] nolu çalışmalarda; k-NN (k-Nearest Neighbor), Naive Bayes ve SVM (Support Vector Machines) yöntemleri kullanılarak metin sınıflandırma performanslarının karşılaştırmışlardır. Çalışma sonucuna göre SVM ve k-NN'in Naive Bayes'e göre daha başarılı sınıflandırma yaptığı görülmüştür.

k-NN ve Naive Bayes'in bit ağırlıklandırma kullanılarak yapılan metin sınıflandırma işleminde, k-NN'in Cosine benzerliği ile birlikte uygulandığında, Naive Bayes'ten daha başarılı olmuştur [11].

[49] nolu çalışmada k değerini 30,40, 50, 60, 70, 80 alarak sınıflandırma çalışması yapmışlar ve en yüksek sınıflandırmayı %90,64 başarıyla k=50 değerinden elde etmişlerdir. Yine aynı çalışmanın başarı oranlarına bakıldığında aynı koşullarda yapılan Naive Bayes'in k-NN'den daha başarılı olduğu gözlemlenmiştir.

[63] nolu çalışmada; küçük ve büyük boyutlu veri setleri kullanarak, test dokümanlarının 6 farklı sınıftan birine atanmasını hedeflemişlerdir. tf ağırlıklandırma, Naive Bayes ile birlikte kullanmışlardır. Büyük boyutlu veri setleri ile yapılan sınıflandırma işleminde, bütün sınıflarda daha başarılı sınıflandırma gerçekleştirmiştir.

Türkçe metinler üzerine de metin sınıflandırma ile ilgili çeşitli çalışmalar bulunmaktadır [46, 64, 65].

[66] nolu çalışmada 50 eğitim ve 25 test dokümanı kullanarak çalışma yapmışlar ve maksimum %76,00'lık başarı elde etmişlerdir.

[67] nolu çalışmada; k-NN ve En Yakın Komşu metotlarını farklı şekillerde uygulamışlar ve en başarılı sonucu %88,40 ile En Yakın Komşu metodunda elde etmişlerdir.

[68] nolu çalışmada metin madenciliği tekniklerinin web verilerine uygulanmasıyla ilgili yaptıkları çalışmada, web verilerinin düz bir yazı gibi değerlendirilemeyeceğini, reklam gibi metin dışı veri barındırdığından veri madenciliği tekniklerinin web sayfalarına direkt uygulanmasının oldukça zor olduğu belirtmişlerdir.

Bir diğer çalışmada 12 sınıfa ayrılmış Çin web sayfaları, Naive Bayes kullanılarak sınıflandırılmıştır. En yüksek sınıflandırma %100,00 başarı ile sigorta sınıfından elde edilmiştir. Ortalama başarı ise %94,80'dir [69].

[70] nolu çalışmada; içerik, profil ve görsel bilgiye göre web sayfalarının sınıflandırılmasını amaçlamışlardır. İçerik bilgisindeki geçen kelimeler ve kelime sayılarına; profil bilgisindeki link sayısı, meta taglar, sayfadaki çerçeve ve açıklama sayısına; görsel bilgi içerisindeki resimlerin ten renginin bulunduğu bölgelere göre sınıflandırma yapmışlardır. Sadece içeriğe göre yapılan sınıflandırmalardan farklı olarak görsel öğeler de kullanmışlardır.

[71] nolu çalışmada kimya ile ilgili web sayfalarının sınıflandırılmasını amaçlamışlardır. Kimya terimlerinden oluşan sözlük kullanılarak, k-NN ve Cosine benzerliği birlikte uygulanmıştır. Konuya ilişkin terimlerden oluşturulan sözlükle yapılan sınıflandırmaların, sınıftaki eğitim dokümanlarında geçen kelimelerle oluşturulan sözlükle yapılan çalışmalara göre daha başarılı olduğu gözlemlenmiştir.

[46] nolu çalışmada; bit, tf ve tf-idf ağırlıklandırmalar, k-NN, SVM, C4.5 ve Naive Bayes ile birlikte uygulayarak, Türkçe web sitelerinin sınıflandırılması gerçekleştirmişlerdir. 2667 eğitim, 1219 test dokümanı ve 2442 eğitim, 1212 test dokümanı olan iki farklı veri seti kullanılarak yapılan çalışmada, k- NN ve Naive Bayes ile yapılan sınıflandırmalarda tf-idf ağırlıklandırmanın, bit ve tf ağırlıklandırmayla yapılan sınıflandırmalara göre daha başarılı olduğu görülmüştür.

Kendisiyle aynı kategorideki benzer haberlerin tespit edilmesini amaçlayan bir çalışmada, kelime köklerinin bulunması için Türkçe doğal dil işleme kütüphanesi Zemberek kullanılmıştır [72].

3. PROJE TASARIMI

Proje ile ilgili detaylı bilgiler ilerleyen bölümlerde verilmiştir.

3.1. Proje Gereksinimleri

Bu projenin yazılımsal anlamda ihtiyaç duyduğu ve kullanılan gereksinimler şunlardır:

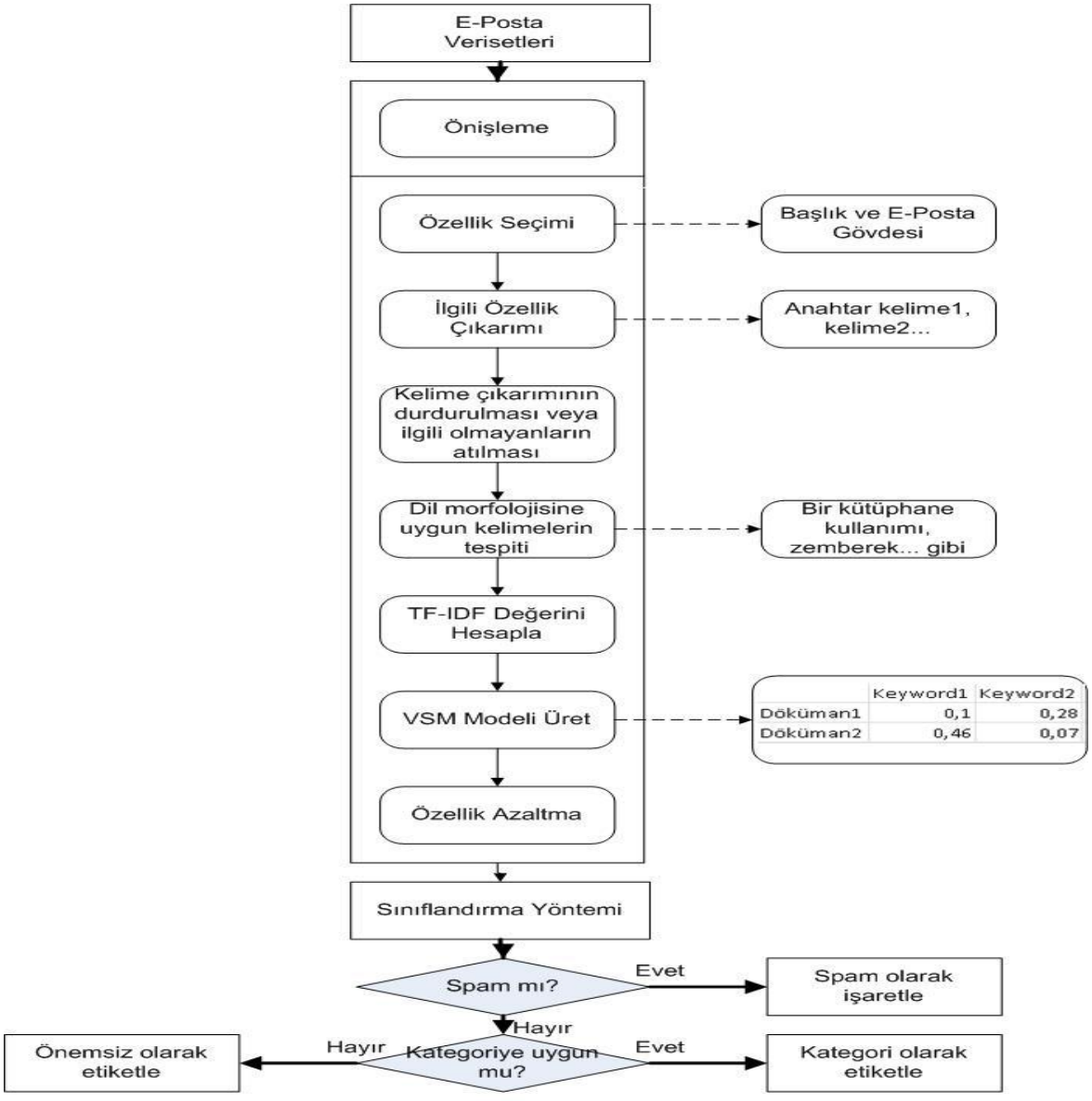
- E-posta veri setlerinin temin edilmesi
- Veri seti üzerinde önişleme tekniklerinin uygulanması
- Özellik seçimi
- E-posta metinleri ile ilgili ilişkili özelliklerin seçilmesi
- İlgili özelliklerin çıkarılması (anahtar kelimelerin)
- Gereksiz kelimelerin atılması veya kelime çıkarımının durdurulması
- Dil morfolojisine uygun kelimelerin işaretlenmesi
- TF-IDF değerlerinin her doküman için anahtar değer bazında hesaplanması
- VSM modelinin üretilmesi
- Özellik azaltılması (CFS, PCA algoritmalarının kullanılması)
- Sınıflandırma yönteminin uygulanması
- E-postanın ilgili kategoriye dahil edilmesi

Bu projenin fiziksel anlamda ihtiyaç duyduğu ve projede kullanılan gereksinimler şunlardır:

- Kullanılan araçlar olarak;
- Java dili geliştirme ortamı (Netbeans gibi.)
- Java SDK geliştirme ailesi,
- Bir veri tabanı yönetim yazılımı (MS Access gibi.)
- Proje için spesifik olarak;
- Dil morfolojisi ile özellik çıkarımı yapabilen programlama kütüphanesi (Zemberek gibi.)
- En az 10 kişiden 100 normal+50 spam olmak üzere e-posta metin veri seti

3.2. Proje Sistem Mimarisi

Akıllı E-Posta Asistanı uygulamasının nihai hedefi kullanıcıların gelen e-postalarını metinde yer alan anahtar kelimelerin çokluğu ve vektörel anlamda ağırlığı, tf-idf gibi çeşitli ölçütlere göre en yakın kategoriye otomatik dahil etmek ve kullanıcının elle ayarlamaya müsait ihtiyaçlarını da bu işleme dahil etmektir. Bu nedenle proje tasarımı özellik çıkarımından sınıflamaya kadar çeşitli aşamalar içermektedir. Bu durum Şekil 3.1.'de görülmektedir.



Şekil 3.1. Akıllı E-posta Asistanı Projesi Sistem Mimarisi

3.3. Proje Başarı Kriterleri

Bu proje çalışmasında gerçekleştirilmiş olan başarı ölçütleri şu şekildedir:

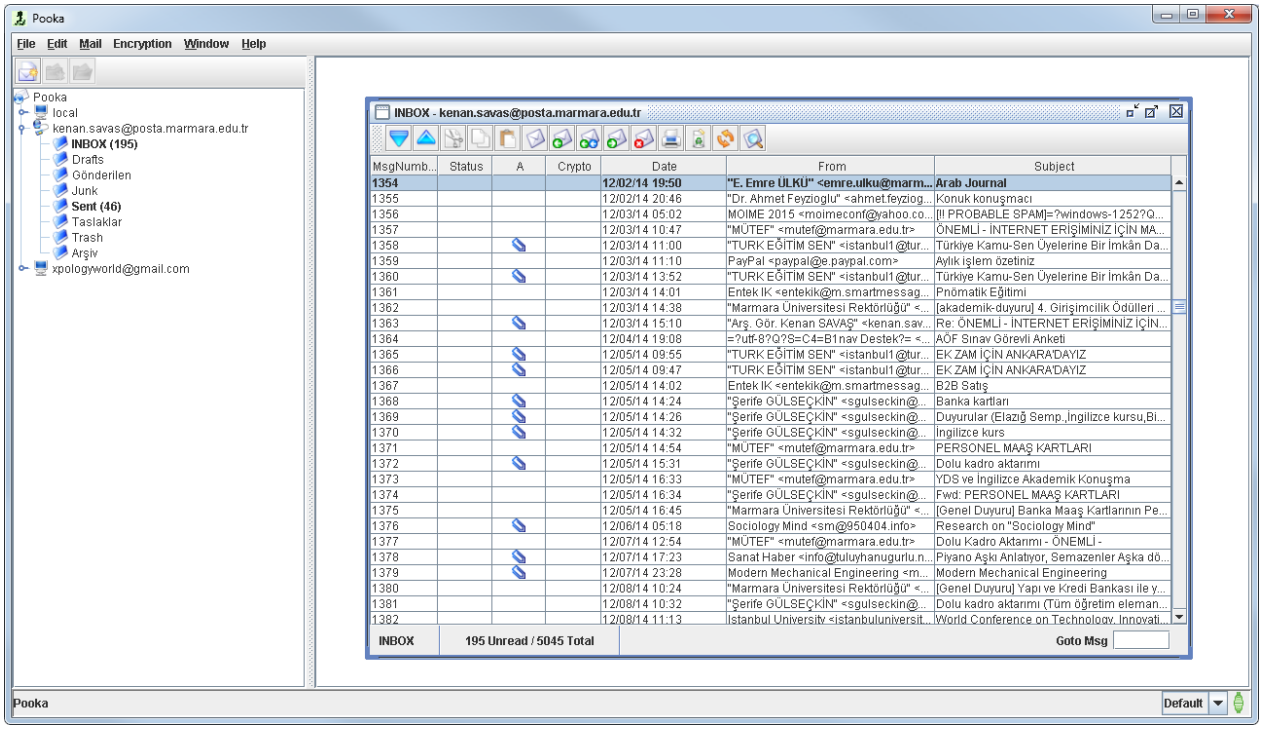
- Veriler üzerinde offline çalışma ve dönüt imkanının sağlanması (anlık gelen e-postanın değerlendirilmesi ve arayüzde hemen gösterilmesi),
- Türkçe dil morfolojisi ile uyumlu kategoriler ile etiketlenmiş bir verisetinin oluşturulması.
- %85 başarı oranı ile gelen e-postaların sınıflandırılabilmesi.
- E-postaların kişiye özel olarak otonom ve elle ayarlabilir şekilde sınıflandırılması.

4. UYGULAMA

Proje kapsamında hem epostaların alınması ve görsel olarak kullanıcıya sunulması, hem de metin madenciliği teknikleri ile incelenmesi gerektiğinden iki parçalı bir arayüz hazırlanmıştır. İlk arayüz için programcıya çok yönlü destek veren ve görsellik açısından da kullanışlılık sunan Pooka açık kaynak kodlu e-posta sisteminden yararlanılmıştır.

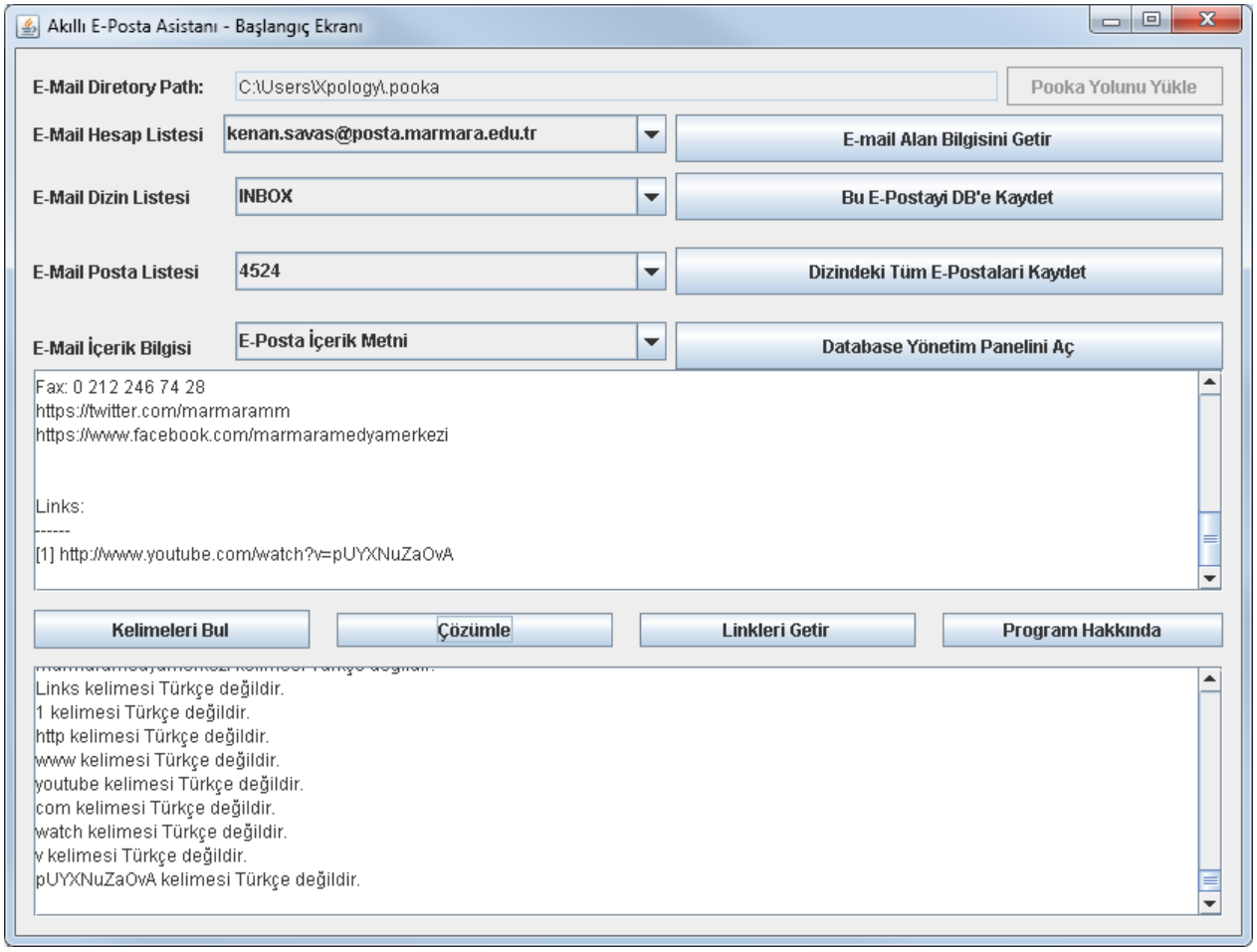
4.1. Uygulama Arayüzü

Hata! Başvuru kaynağı bulunamadı.'de uygulamaya ait arayüz görülmektedir. Bu açık kaynak kod uygulaması ile java



Şekil 3. 2 Pooaka Açık Kaynak Kod E-Posta İstemci Arayüzü

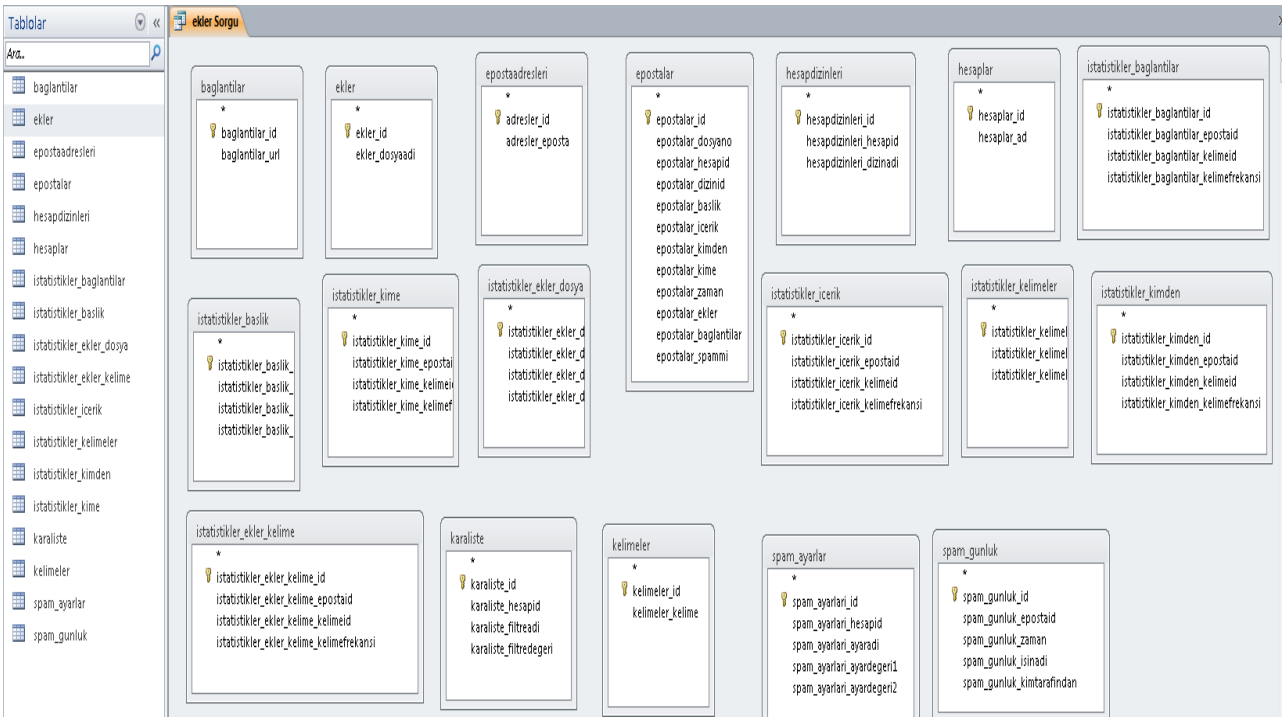
tabanlı olarak e-posta verileri üzerinde çok yönlü olarak işlem yapılabilir ve e-postalar arama, yazdırma, pek çok farklı e-posta sunucusundan farklı protokollere dayalı e-postaların okunması kolaylıkla gerçekleştirilmektedir. Proje kapsamında hedeflenen e-postaların lokal olarak metin madenciliğine tabi tutulması ve ilgili e-postalar üzerinde sık kullanıma ya da spam olarak değerlendirilmesine göre ilgili süreç Pooka sistemi ile entegre olarak gerçekleştirilmektedir. Tez geliştirme sürecinde var olan e-postalara ait e-postanın sahip olduğu başlık, zaman, kimden kime gibi bilgileri almak için basit bir arayüz üzerinde geliştirme yapılmış ve projeye dahil edilmiştir. Bu geliştirlen ufak arayüz ile sisteme kayıtlı tüm e-postalara ulaşılarak e-postaların tüm bilgileri e-posta raw verisi içerinden çıkarılabilmekte ve kelime kelime dilbilgisi analizine tabi tutulabilmektedir. Bu durum Şekil 3. 3'de görülmektedir. Öncelikle sistemin kurulu olduğu dizin yolu bilgisi ile tüm e-posta verileri liste olarak programa yüklenmektedir. Daha sonra numraya göre seçilen herhangi bir e-posta raw datasındaki kimden, kime, konu bilgisi, zaman bilgisi, başlık bilgisi, tüm raw içerik ve e-posta ana içerik bilgileri çıkarılmaktadır. Bu durum E-Mail Bilgisi Getir butonu ile gerçekleştirilmektedir. Daha sonra aşağıda yer alan 4 adet buton ile dil yapısı ile ilgili işlemler gerçekletirlmektedir. Mesela getirilen e-postaya ait kelimelerin çıkarılması, kelimelerin Türkçe olup olmaması veya e-posta bilgisi içindeki tüm email adresleri veya http, https gibi URL bağlantıları da e-postalardan çıkarılabilmektedir.



Şekil 3. 3 E-Postalar Üzerinde İlgili E-Posta Alanlarının Alınması ve Dil Açısından Analiz Arayüzü

4.1.1. Uygulama Veritabanı Yapısı

Çalışmada oluşturulan veritabanına ait tablolar Şekil 3. 4'de görülmektedir.



Şekil 3. 4 Geliştirilen Yazılıma ait Veritabani Tabloları Yapısı

4.2. DENEYLER

Ölçüler : Duyarlık, Anma, Doğruluk

Bu çalışmada filtrelemenin başarısı değerlendirilirken duyarlık, anma ve doğruluk ölçüleri kullanılmıştır. Değerlendirmek istenen şey; filtrenin istenmeyen elektronik posta seçebilme başarısıdır [1]. Tez çalışması kapsamında geliştirilen yazılımı değerlendirirken Tablo 3. 1 kullanılmıştır.

Tablo 3. 1 E-Posta Filtreleme Karşılatırma Tablosu

	(Uzman Görüşü) Evet (True)	(Uzman Görüşü) Hayır(False)
İstenmeyen e-posta (positive)	a	b
Normal e-posta (negative)	c	d

a (true positive)
b (false positive)
c (true negative)
d (false negative)

a = filtrenin istenmeyen elektronik posta olarak seçtiği ve uzman kişi tarafından istenmeyen elektronik posta olarak belirlenmiş elektronik posta sayısı

b = filtrenin istenmeyen olarak seçtiği fakat uzman kişi tarafından normal olduğu belirlenmiş elektronik posta sayısı

c = filtrenin normal olarak seçtiği ve uzman kişi tarafından normal elektronik posta olarak belirlenen elektronik posta sayısı

d = filtrenin normal olarak seçtiği fakat uzman kişi tarafından istenmeyen olarak belirlenmiş elektronik posta sayısı

Bu bilgiler ışığında istenmeyen elektronik posta için duyarlık ve anma aşağıdaki gibi hesaplanır.

Anma = c / (c+d)

Duyarlık= a / (a+b)

Yapılan deneylerde toplam 679 adet mesajdan oluşan örneklemimiz öğrenme ve test kümesi olmak üzere ikiye ayrılmıştır. Öğrenme kümesi alınma tarihlerine göre sıralanmış, 68 adet istenmeyen ve 477 adet normal elektronik posta mesajı mesajından oluşturulmuştur.

Test kümesi ise yine alınma tarihine göre sıralanmış ve alınma tarihleri talim kümesindeki mesajlardan daha sonra olan 16'sı istenmeyen ve 118'ü normal elektronik posta mesajı olmak üzere toplam 134 mesajdan oluşturulmuştur. Test kümesindeki mesajların filtrenin daha önceden hiç görmediği mesajlar olması ölçme işleminin doğruluğunu arttıracaktır.

Testimizde esas ve eşik değerini 0,05 olarak belirlediğimiz filtreleme işleminde;

İstenmeyen elektronik postalar için

Duyarlık = 14/(14+2)=0,875 = %87,5

Anma = 104/(104+14)=0,88 = %88

Normal elektronik postalar için

Duyarlık = 104/(104+0)=1,00 = %100

Anma = 14/(14+2)=0,875 = %87,5

olarak elde edilmiştir. Bu veriler karşılaştırmalı olarak

Tablo 3. 2'de ayrıntılı olarak görülebilir.

Tablo 3. 2 Karşılaştırmalı Spam E-Postaları Filtreleme Test Sonuçları

	Toplam mesaj sayısı	Test Edilen mesaj sayısı	İstenmeyen elektronik posta mesajları		Normal elektronik posta mesajları	
			Duyarlık (%)	Anma(%)	Duyarlık (%)	Anma (%)
Örneklem	679	134	87,5	88	100	87,5

5. SONUÇ VE ÖNERİLER

Bu çalışmada istenmeyen elektronik postaların sınıflandırması ve bu probleme uygulanışı tfidf hesaplamalarına dayalı olarak metin madenciliği yöntemlerinden biri olan naive bayes sınıflandırma tekniği kullanlarak incelenmiş ve yapılan deneylerle başarılı sonuçlar elde edilmiştir. Günümüzde bu yönteme dayalı çok sayıda hem istemci, hem sunucu tabanlı istenmeyen elektronik posta engelleme yazılımı mevcuttur. İstenmeyen elektronik posta engelleme teknikleri ve kullanıcının kullanımına dayalı kategorize edilebilir e-postalar bahsinde de belirtildiği gibi bugün bu probleme çok farklı açılardan yaklaşılmaktadır. Bununla birlikte sadece bu yöntem değil, pek çok farklı yöntem kullanılarak açık kodlu yazılımların sayısı zamanla artmakta ve kullandıkları yöntemlerde küçük farklılıklar göstermektedir.

Yapılan çalışmada açık kaynak koda dayalı ve java ortamında sunulan Pooka e-posta istemci yazılımın sağladığı metot ve kütüphanenin kullanımı ile tüm e-postaların alınabilmesi kolay bir şekilde sağlanmış olup, bu yazılım pek çok protokol ile (POP3, IMAP, SMTP gibi) desteklediğinden kullanıcı farklı farklı e-posta hesaplarını kolay bir şekilde yönetebilmekte, tanımlayabilmekte, arama ve yazdırma gibi işlemleri yapabilmektedir. Bu kütüphanenin en önemli avantajı e-postaları _hdr ve _msg şeklinde dosya son ekleri ile tüm e-postaları disk üzerinde cache alabilmekte ve bu proje kapsamında da bu yöntem ile –postalara kolaylıkla erişilebilmektedir. İleride Pooka kütüphanesinin sağladığı spam özellikleri de projenin devamı eklenerek daha tümleşik ve daha entegre bir çalışma söz konusu olabilir. Bu çalışmada Pooka'dan bağımsız olarak e-postalar değerlendirilmekte, ve kendi veritabanımızda değerlendirilerek spam olup olmaması işaretlenmektedir.

Bu çalışmada kelime oluşlarının sayısını da hesaba katmanın filtreleme başarısını daha da arttırdığı görülmüştür. İstenmeyen ve normal elektronik postaları tek bir kullanıcının arşivinden ve sayıca daha fazla miktarlarda seçmenin başarımı çok daha arttıracaktır.

Tamamı Türkçe elektronik posta mesajlarından oluşturulan örneklem üzerinde deneyler yapılmıştır. Mesajların Türkçe olmasının çalışmayla alakası sadece Türkçe istenmeyen elektronik posta gönderenlerin mesaj karakteristiklerini değerlendirme bağlamındadır. Mesajlarda herhangi bir dilbilimsel çalışma yapılmamıştır. Türkçe dil yapısına dayalı kelimelerin tespitinde çalışmada Zemberek açık kaynak kod kütüphanesinden yararlanılmıştır. İkili veya üçlü ifade grupları, sıfat isim tamlamaları bu çalışmada eklenmemiş olup, sadece kelimelerin bireysel frekansları dikkate alınmıştır. Html içerikli elektronik posta mesajlarında ayrıca html mesajın içerdiği tüm URL bağlantıları, içerdiği ekler ve içerdiği tüm resim dosya linkleri de veritabanına atılmaktadır. Ayrıca html yapıda taglardan arınmış düzgün bir metin bilgisi veritabanına kaydedilmektedir. İleride yapılması planlanan çalışmalarda bu özellikler ile projenin devam ettirilerek ekte yer alan dosyalara bağlı filtreleme ya da kelime gruplarına dayalı filtreleme yapma imkanı da söz konusudur. Bu hususlarında göz önünde bulundurulmasının sağlayacağı katkılar olumlu bir şekilde projeye katkı sağlayacaktır.

Ayrıca istenmeyen elektronik postaları normal elektronik postalardan büyük ölçüde ayıran bir husus da aynı anda çok sayıda kişiye gönderilmeleridir. Bu durum istenmeyen posta mesajlarının boyutlarıyla ilgili de bize fikir vermektedir. Örneğin bir istenmeyen elektronik posta mesajı herhangi bir ek dosya içerme eğiliminde olmayacaktır. Bir ek dosya içermesi durumunda hem dağıtımı zorlaşacak hem de kullanıcı mesajın konusunu görür görmez ekte iletilen bilgiyi dikkate almayacaktır. Bu tip küçük ayrıntıları da dikkate almak ileri çalışmalarda yenilik ve fonksiyonellik açısından olumlu katkı sağlayacaktır.

ÖZGEÇMİŞ

İsmim Kenan Savaş. 1983 İstanbul doğumluyum. 2000 yılında girmiş olduğum Marmara Üniversitesi Bilgisayar Programcılığı önlisans programından 2002 yılında başarıyla ve onur öğrencisi olarak mezun oldum. Daha sonra dikey geçiş programı ile 2003 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar ve Kontrol Öğretmenliği programına yerleştim. 1 yıl İngilizce hazırlık eğitim programı da içeren bu öğretmenlik programından 2007 yılında yüksek mezuniyet notu ile başarıyla mezun oldum. Mezuniyet sonrası öğretim görmüş olduğum kendi bölümümde Kontrol anabilim dalında açılan Araştırma Görevlisi kadrosunu kazandım. 2007 yılında girmiş olduğum Bilgisayar ve Kontrol Eğitimi Yüksek Lisans programını 2010 yılında "Kontrol Eğitimi için Web Tabanlı Uygulama Araçlarının Geliştirilmesi" başlıklı tez çalışmam ile başarıyla tamamladım. Aynı yıl girmiş olduğum Bilgisayar ve Kontrol Eğitimi doktora programına halen devam etmekte "Özelleştirilebilir Kontrolör Tasarımına Dayalı Uzaktan Gerçek Zamanlı Sistem Kontrolü" konu başlıklı tez çalışmam üzerinde araştırma yapmaya devam etmekteyim. Teknik Eğitim Fakültelerinin YÖK tarafından alınan karar gereği 2010 yılında kapatılması nedeniyle 2015 yılında dolu kadro aktarıma ile geçtiğim Marmara Üniversitesi Teknoloji Fakültesi Elektrik-Elektronik Mühendiliği bölümünde Araştırma Görevlisi olarak akademik hayatıma devam etmekteyim. 2013 yılı yaz döneminde açılan ve benim de başvurduğum ÖSYM Mühendilik Tamamlama sınavı ile aynı yıl Gebze Teknik Üniversitesi Bilgisayar Mühendiliği programını kazandım. Halen bu programda da eğitim-öğretim faaliyetlerine devam etmekteyim.

KAYNAKLAR

[1] C. ALTUNYAPRAK, "Bayes Yöntemi Kullanarak İstenmeyen Elektronik Postaların Filtrelenmesi," MSc Thesis, Fen Bilimleri Enstitüsü, Mugla Üniversitesi, Muğla, Turkey, 2006.

[2] İ. F. PİLAVCILAR, "Metin Madenciliği ile Metin Sınıflandırma," MSc Thesis, Fen Bilimleri Enstitüsü, Yıldız Teknik Üniversitesi, Istanbul, Turkey, 2007.

[3] J. Han and M. Kamber, "Data Mining: Concepts and Techniques," 2nd ed: Morgan Kaufmann Publishers, San Francisco, 2006, pp. 5-7, 105-106, 348-350.

[4] S. M.TAHA, "Metin Madenciliği ile Döküman Demetleme," MSc Thesis, Bilişim Enstitüsü, Gazi Üniversitesi, Ankara, Turkey, 2011.

[5] A. Visa, "Technology of Text Mining," in *Machine Learning and Data Mining in Pattern Recognition*, ed: Springer Science + Business Media, 2001, pp. 1-11.

[6] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, "Knowledge discovery in databases: an overview," *AI Magazine*, vol. 13, pp. 57-70, 1991.

[7] M. Konchady, *Text Mining Application Programming*, 1st ed.: Charles River Media, Boston, 2006.

[8] A. Adsız, "Metin Madenciliği," in *Bilişim Sistemleri ve Mühendislik Fakültesi, Kazakistan*, ed: Ahmet Yesevi Üniversitesi, 2006, pp. 17-19.

[9] M. F. KARACA, "Metin Madenciliği Yöntemi ile Haber Sitelerindeki Köşe Yazılarının Sınıflandırılması," MSc Thesis, Fen Bilimleri Enstitüsü, Karabük Üniversitesi, Karabük, Turkey, 2012.

[10] T. M. Mitchell, *Machine Learning*, 1st ed. McCraw Hill, New York, 1997.

[11] P. Soucy and G. W. Mineau, "A simple KNN algorithm for text categorization," presented at the Proceedings 2001 IEEE International Conference on Data Mining, 2001.

[12] U. İlhan, "Application Of KNN and FPTC Based Text Categorization Algorithms to Turkish News Reports," Bilkent University, 2001.

[13] O. Durmaz and H. Ş. Bilge, "Metin sınıflandırmada boyut azaltmanın etkileri ve özellik seçimi," in *Signal Processing and Communications Applications (SIU 2011), 2011 IEEE 19th Conference*, Antalya, 2011, pp. 21-24.

[14] C.-M. Chen, H.-M. Lee, and C.-C. Tan, "An intelligent web-page classifier with fair feature-subset selection," *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 967-978, 2006/12 2006.

[15] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos, "An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages," presented at the Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '00, 2000.

[16] G. Cormack and T. Lynam, "Spam corpus creation for TREC," in *Proceedings of Second Conference on Email and Anti-Spam, CEAS'2005*, 2005.

[17] S. H. Mueller. (2010, April). *What is spam?* Available: http://spam.abuse.net/overview/whatisspam.shtml

[18] A. PARLAK, "Spam E-Mail Detection And Filtering Using Neuro-Fuzzy Classifiers And Genetic Algorithms," MSc Thesis, The Graduate School of Natural and Applied Sciences, Bahcesehir University, Istanbul, Turkey, 2010.

[19] (01 April 2010). *The global economic impact of spam. report #409*. Available: http://www.ferris.com/2005/02/24/the-global-economic-impact-of-spam-2005

[20] G. Silahtaroğlu, *Kavram ve Algoritmalarıyla Temel Veri Madenciliği*. İstanbul: Papatya Yayıncılık, 2008.

[21] "Bilgi Türleri," in *Büyük Larousse Sözlük Ve Ansiklopedisi* vol. 23, ed. İstanbul: İnterpress Basın ve Yayıncılık, 1986, pp. 12164-12165.

[22] A. S. Albayrak and K. Yılmaz. (2009) Veri Madenciliği, Karar Ağacı Algoritmaları ve İMKB Verileri Üzerine Bir Uygulama. *Isparta: Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Dergisi*. 31-52.

[23] M. R. Berthold, C. Borgelt, F. Höppner, and F. Klawonn, "Guide to Intelligent Data Analysis," in *Texts in Computer Science*, ed: Springer London, 2010.

[24] U. Fayyad, Piatetsky-Shapiro, and G. S. P, "From Data Mining to Knowledge Discovery: An Overview," *Advances in Knowledge Discovery and Data Mining*, pp. 1-34, 1996.

[25] H. Akpınar. (2010, Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği. *İstanbul Üniversitesi İşletme Fakültesi Dergisi 29(1)*, 1-22.

[26] A. Demirel, "Metin Madenciliği Yöntemleri ile Sosyal Medyadan Toplanan Fotoğraflı Paylaşımların, Metin-Fotoğraf Eşleşmesinin İncelenmesi," MSc Thesis, Fen Bilimleri Enstitüsü Matematik Bilgisayar Anabilim Dalı, Beykent Üniversitesi, Istanbul, Turkey, 2015.

[27] A. Hotho, A. Nürnberger, and G. Paaß, "A Brief Survey of Text Mining," *GLDV Journal for Computational Linguistics and Language Technology*, 2005.

[28] H. A. D. Prado and E. Ferneda. (2007, Emerging Technologies of Text Mining:Techniques and Applications. *Idea Group Reference*.

[29] R. Feldman and J. Sanger, "Introduction to Text Mining," in *Advanced Approaches in Analyzing Unstructured Data*, ed: Cambridge University Press (CUP), 2007, pp. 1-18.

[30] V. TUNALI, "Metin Madenciliği için İyileştirilmiş Bir Kümeleme Yapısının Tasarımı ve Uygulaması," PhD Thesis, Fen Bilimleri Enstitüsü, Marmara Üniversitesi, Istanbul, Turkey, 2011.

[31] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman, "Incremental hierarchical clustering of text documents," presented at the Proceedings of the 15th ACM international conference on Information and knowledge management - CIKM '06, 2006.

[32] T. Jo, "The Implementation of Dynamic Document Organization Using the Integration of Text Clustering and Text Categorization," PhD Thesis, School of Information Technology and Engineering, University of Ottawa, Istanbul, Turkey, 2006.

[33] Y. Li, "High Performance Text Document Clustering," PhD. Thesis, Wright State University, 2007.

[34] D. Reforgiato Recupero, "A new unsupervised method for document clustering by using WordNet lexical and conceptual relations," *Information Retrieval*, vol. 10, pp. 563-579, 2007/10/16 2007.

[35] M. Shafei, S. Wang, R. Zhang, E. Milios, B. Tang, J. Tougas, *et al.*, "A Systematic Study of Document Representation and Dimension Reduction for Text Clustering," Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Technical Report, CS-2006-052006.

[36] A. Strehl and J. Ghosh, "Relationship-Based Clustering and Visualization for High-Dimensional Data Mining," *INFORMS Journal on Computing*, vol. 15, pp. 208-230, 2003/05 2003.

[37] (2006). *Bilgisayar Proje 2 Sunumu*. Available: www.cs.itu.edu.tr~gunduz/courses/projeII/clustering.pdf

[38] G. Fung. (2001, June). *A Comprehensive overview of basic clustering algorithms*. Available: http://www.cs.wisc.edu/~gfung/clustering.pdf

[39] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *In Proceedings of the 20th VLDB Conference*, Santiago, Chile, 1994, pp. 144-155.

[40] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, pp. 613-620, 1975/11/01 1975.

[41] K. Hammouda and M. Kamel, "Collaborative Document Clustering," in *Proceedings of the 2006 SIAM International Conference on Data Mining*, ed: Society for Industrial & Applied Mathematics (SIAM), 2006, pp. 453-463.

[42] B. Larsen and C. Aone, "Text mining using linear-time document clustering," presented at the KDD-99, San Diego, California, 1999.

[43] C. J. V. Rijsbergen, *Information Retrieval*: Buttersworth, London, 1998.

[44] G. Kowalski, *Information retrieval systems – Theory and Implementation*: Kluwer Academic Publishers, 1997.

[45] K. Lagus, *Text Mining with the WEBSOM*. Helsinki University of Technology, Finland, 2000.

[46] C. Toraman, F. Can, and S. Kocberber, "Developing a text categorization template for Turkish news portals," presented at the 2011 International Symposium on Innovations in Intelligent Systems and Applications, 2011.

[47] B. V. Dasarathy, "Nearest-neighbor classification techniques," *IEEE Computer Society Press, Los Alamitos, California*, 1991.

[48] D. Coomans and D. L. Massart, "Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules," *Analytica Chimica Acta, Elsevier BV*, vol. 136, pp. 15-27, 1982.

[49] A. Sanwaliya, K. Shanker, and S. C. Misra, "Categorization of News Articles: A Model Based on Discriminative Term Extraction Method," presented at the 2010 Second International Conference on Advances in Databases, Knowledge, and Data Applications, 2010.

[50] S. Eyheramendy, Lewis, D.D. and Madigan, D., "On the naive bayes model for text categorization," in *Proceedings of Artificial Intelligence and Statistics*, 2003, pp. 3-6.

[51] R. J. Roiger and M. W. Geatz, *Data Mining: A Tutorial-Based Primer*: Addison Wesley, 2003.

[52] S.-H. Liao, P.-H. Chu, and P.-Y. Hsiao, "Data mining techniques and applications – A decade review from 2000 to 2011," *Expert Systems with Applications*, vol. 39, pp. 11303-11311, 2012/09 2012.

[53] C. Çiflikli and E. Kahya-Özyirmidokuz, "Implementing a data mining solution for enhancing carpet manufacturing productivity," *Knowledge-Based Systems*, vol. 23, pp. 783-788, 2010/12 2010.

[54] F. Bao, X. He, and F. Zhao, "Applying Data Mining to the Geosciences Data," *Physics Procedia*, vol. 33, pp. 685-689, 2012.

[55] C.-H. Hsu, "Data mining to improve industrial standards and enhance production and marketing: An empirical study in apparel industry," *Expert Systems with Applications*, vol. 36, pp. 4185-4191, 2009/04 2009.

[56] A. Pons-Porrata, R. Berlanga-Llavori, and J. Ruiz-Shulcloper, "Topic discovery based on text mining techniques," *Information Processing & Management*, vol. 43, pp. 752-768, 2007/05 2007.

[57] C. Gieger, H. Deneke, and J. Fluck, "The future of text mining in genome-based clinical research," *Biosilico*, vol. 1, pp. 97-102, 2003/07 2003.

[58] C. M. Fuller, D. P. Biros, and D. Delen, "An investigation of data and text mining methods for real world deception detection," *Expert Systems with Applications*, vol. 38, pp. 8392-8398, 2011/07 2011.

[59] W. W. Cohen and H. Hirsh, "Joins that generalize: Text classification using WHIRL," in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, Wuhan, China, 1998, pp. 169-173.

[60] Y. Yang and X. Liu, "A re-examination of text categorization methods," presented at the Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99, 1999.

[61] F. Sebastiani, "Machine learning in automated text categorization," *CSUR*, vol. 34, pp. 1-47, 2002/03/01 2002.

[62] S. S. R. Mengle, N. Goharian, and A. Platt, "FACT: Fast Algorithm for Categorizing Text," presented at the 2007 IEEE Intelligence and Security Informatics, 2007.

[63] L. Li, Y.-g. Huang, and Z.-w. Liu, "Chinese text classification for small sample set," *The Journal of China Universities of Posts and Telecommunications,* vol. 18, pp. 83-89, 2011/09 2011.

[64] M. F. Amasyalı and B. Diri, "Automatic Turkish Text Categorization in Terms of Author, Genre and Gender," in *Natural Language Processing and Information Systems*, ed: Springer Science + Business Media, 2006, pp. 221-226.

[65] A. Güran, S. Akyokuş, N. Güler, and Z. Gürbüz, "Turkish text categorization using n-gram words," in *International Symposium on INnovations in Intelligent SysTems and Applications (INISTA 2009)*, Trabzon, 2009.

[66] M. F. Amasyalı and T. Yıldırım, "Otomatik haber metinleri sınıflandırma," in *Signal Processing and Communications Applications (SIU 2004), 2004 IEEE 12th Conference*, Aydın, 2004, pp. 224-226.

[67] R. Aşlıyan and K. Günel, "Metin içerikli Türkçe doküman sınıflandırılması," ed: Akademik Bilişim 2010, Muğla Üniversitesi, Muğla, 2010.

[68] S. Yin, Y. Qiu, and J. Ge, "Research and Realization of Text Mining Algorithm on Web," presented at the 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007), 2007.

[69] W. Huang, LuXiongXu, J. Duan, and Y. Lu, "Chinese Web-page Classification Study," presented at the 2007 IEEE International Conference on Control and Automation, 2007.

[70] A. Ahmadi, M. Fotouhi, and M. Khaleghi, "Intelligent classification of web pages using contextual and visual features," *Applied Soft Computing*, vol. 11, pp. 1638-1647, 2011/03 2011.

[71] C.-Y. Liang, L. Guo, Z.-J. Xia, F.-G. Nie, X.-X. Li, L. Su, *et al.*, "Dictionary-based text categorization of chemical web pages," *Information Processing & Management,* vol. 42, pp. 1017-1029, 2006/07 2006.

[72] A. Karadağ and H. Takçı, "Metin madenciliği ile benzer haber tespiti," ed: Akademik Bilişim 2010, Muğla Üniversitesi, Muğla, 2010.

EKLER

Tez çalışması ciltlenmiş kitap halinde ve içine ilgili yazılım ve tez çalışma raporunun yer aldığı CD ile Bilgisayar Mühendisliği bölümüne 3 nüsha halinde teslim edilmiştir.