Metin Madenciliği ile E-Ticaret Sitelerinin Belirlenmesi

Tuğba KAŞIKÇI[1], Hadi GÖKÇEN[2]

[1]Yönetim Bilişim Sistemleri, Bilişim Enstitüsü, Gazi Üniversitesi, Ankara, Türkiye
[2]Endüstri Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi, Gazi Üniversitesi, Ankara
ozdemir.tugba@gazi.edu.tr, hgokcen@gazi.edu.tr
(Geliş/Received: 30.10.2013; Kabul/Accepted: 10.03.2014)

DOI: 10.12973/bid.2014

Özet– Bu çalışmada kullanıcı tarafından belirtilen internet sitelerinin içeriğini analiz ederek metin madenciliği yöntemleri kullanarak bu sayfaların elektronik ticaret (e-ticaret) sitesi olup olmadığına karar veren bir uygulama geliştirilmiştir. Uygulama özel çalışmalarda kullanılmak üzere hazırlanmıştır ve kullanıcılara e-ticaret sitelerinin bulunmasını kolaylaştırmayı amaçlamaktadır. Çalışmada metin sınıflandırmada kullanılmak üzere farklı kaynaklardan veriler toplanmış ve kullanıma hazırlanmıştır. Bu veriler üzerinde k-NN En-Yakın-Komşu ve Naïve Bayes sınıflandırma algoritmaları kullanılarak elde edilen sonuçlar karşılaştırılmıştır. Daha iyi sonuç verdiği gözlemlenen algoritma seçilip Java programlama dili ile masaüstü uygulaması olarak hazırlanan ara yüze aktarılarak kullanımına sunulmuştur.

Anahtar Kelimeler– metin madenciliği, metin sınıflandırma, elektronik ticaret, e-ticaret

Determination of E-Commerce Sites by Text Mining

Abstract– In this study, an application which decide whether or not the pages specified by the user are electronic commerce (e-commerce) sites by analyzing the contents of web sites and by using text mining techniques is developed. The application is designed to be used in special studies and aims to facilitate users to find e-commerce sites. In this study, the data was collected from different sources to be used in text classification and prepared for useage. k-NN Nearest Neighbour and Naïve Bayes classification algorithms are used on this data and obtained results are compared. Algorithm that gives better results is selected and a desktop application with the Java programming language is developed for usage.

Keywords– text mining, text classification, electronic commerce, e-commerce

1. GİRİŞ (INTRODUCTION)

İnternetin hızla gelişmesiyle birlikte artık bilgiye ulaşmadaki engeller büyük ölçüde ortadan kalmıştır. Sadece 5 yıl içinde 50 milyon kullanıcıya ulaşan internet, 7'den 70'e herkesin kullandığı teknoloji ürünüdür ve her gün hatta her an yenilenmektedir [1]. Ancak bu durum beraberinde farklı bir zorluk getirmiştir. Bu, istenilen bilgiyi elde etmedeki zorluktur. Bilgi kirliliği ve kullanıcı hata ya da eksikliklerinden kaynaklanan, özensiz bilgi paylaşımı, bilgiye erişimi zorlaştırmıştır. Bu bilgi karmaşasının içerisinde kaybolmadan istenileni elde etmek için çözüm metin madenciliği ve metin sınıflandırmadır.

Metin madenciliği, doğal dil işleme ile veri madenciliğinin bir arada kullanılmasıdır [2]. Doğal dil işleme, insana özgü tüm dillerin işlenmesi ve kullanılması amacı ile araştırma yapan bilim dalıdır [3]. Veri madenciliği büyük veri yığınları içerisinden gelecekle ilgili tahminde bulunabilmemizi sağlayabilecek bağıntıların bilgisayar programı kullanarak aranmasıdır [4]. Metin madenciliğinde veri madenciliği yöntemleri ile eldeki veriler sınıflandırılarak, gruplandırılarak ya da veriler arasında ilişkiler, bağıntılar, istatistiksel sonuçlar oluşturularak modeller oluşturulur. Oluşturulan model, oluşturulduğu veri kümesinde olmayan yeni bir kayıt geldiğinde, bu kayıt hakkında tahminleme yapma imkânı verir [5]. Metin sınıflandırma metin madenciliğinin yan uygulama alanıdır [6] ve dokümanları bilinen sınıflara atama işlemini yapar.

Bu çalışmada öncelikle metin sınıflandırma kavramı üzerinde durulmuş ve metin sınıflandırma sürecinden bahsedilmiştir. Metin sınıflandırmada sıklıkla kullanılan ve başarılı sonular verdiği gözlemlenen [7-9] k-NN En-Yakın-Komşu ve Naïve Bayes algoritmalarına değinilmiştir. Daha sonra bir kısmı elektronik ticaret sitelerinden oluşan bir eğitim kümesi ele alınmıştır. Ön işleme aşamaları, anahtar kelime sözlüğü oluşturma ve vektör uzay modelinin oluşturulması aşamalarının

ardından eğitilmeye hazır hale gelen veriler k-NN En-Yakın-Komşu ve Naïve Bayes algoritmalarıyla eğitilmiştir. Ardından bu algoritmaların uygulanmasıyla ortaya çıkan sonuçlara değinilmiştir. Daha sonra çalışma Java programlama dili ile hazırlanan ara yüze aktarılarak kullanıcının kullanımına sunulmuştur.

2. METİN SINIFLANDIRMA (TEXT CLASSIFICATION)

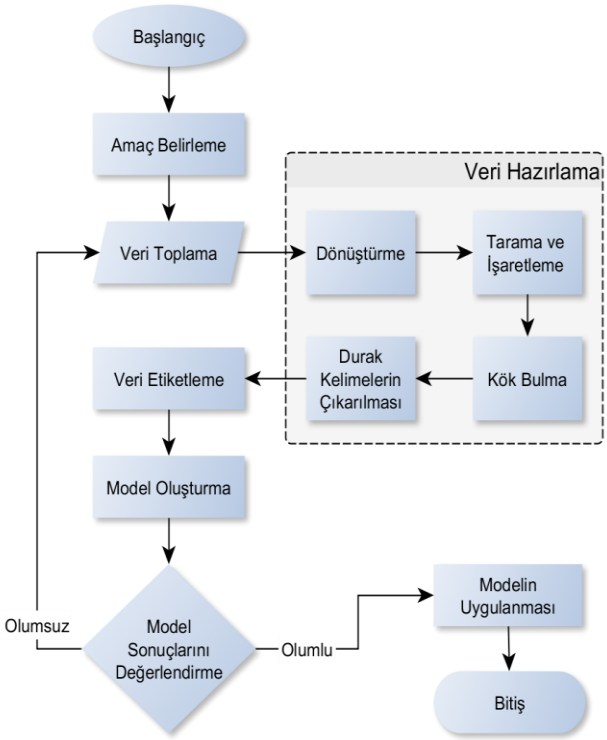
Sınıflandırmanın amacı yeni bir nesnenin özelliklerini açıklamak ve bu yeni nesnenin daha önceden tanımlanmış sınıf setlerinden birine atamasını yapmaktır [10]. Günümüzde metin sınıflandırma çok çeşitli alanlarda uygulanmaktadır. Bunlardan bazıları şunlardır: Metin bulup getirme ve kütüphane organizasyonu gibi alanlarda destek sağlayan "dokümanlara sınıf ataması yapmak", sınıfları insanların atadığı uygulamalarda "sınıf atamasında yardımcı rol oynamak", mesajları, haberleri ve diğer metin dizisi halindeki bilgileri alıcılara ulaştırmak", doğal dil işleme sistemlerinin bir parçası olarak; "ilgisiz metinleri ve metin parçalarını filtrelemek", "metinleri, sınıf bazlı işleme mekanizmalarına yönlendirmek" veya "sınırlı şekillerde bilgi edinimini sağlamak", "sözcük analizi işlerinde yardımcı olmak (sözcük belirsizliğini giderme gibi)" vb. [11].

Günümüze kadar metin sınıflandırmada birçok yöntem kullanılmıştır. Bu yöntemlerden bazıları şunlardır: K-En-Yakın-Komşu sınıflandırması (k-NN) [7-9], Naïve Bayes olasılıklı sınıflandırma [7,8], Karar Ağaçları [7,8], Yapay Sinir Ağları [7,8], Destek Vektörleri (SVM) [7]. Bunların yanında birden fazla yöntemin bir arada kullanılıp sonuçlarının karşılaştırmasının yapıldığı çalışmalar da olmuştur [12-16].

Bu çalışmada iyi sonuçlar verdiği gözlemlenen denetimli öğrenme algoritmalarından k-NN ve Naïve Bayes sınıflandırma yöntemleri kullanılmıştır. Denetimli öğrenmede sonuçlar eldeki bir kısım verinin eğitilmesiyle elde edilir. Metin sınıflandırma süreci iki aşamadan oluşur:

1. Eğitim süreci (öğrenme kümesi – training)
2. Sınıflandırma süreci (deneme kümesi – test) [8]

Eğitim sürecinde hangi sınıfa ait olduğu bilinen ve veri tabanından rastgele seçilen bir kısım veri (öğrenme kümesi - training set) eğitilerek bir model oluşturulur. Sınıflandırma sürecindeyse yine veri tabanından seçilen ve öğrenme kümesinde yer almayan (deneme kümesi- test set) veriler ile uygulanır [17]. Metin sınıflandırma süreci Şekil1'de gösterilmiştir.



Şekil 1. Sınıflandırma aşamaları (Classification phases)

2.1. *Metin Gösterimi (Text Display)*

Dokümanları sınıflandırmaya başlamadan önce, metin formatının düzgün ve kullanılmaya hazır olduğundan emin olmak için dokümanların bazı işlemlerden geçmesi gerekmektedir. Bu işlemler sırasıyla açıklanmıştır.

2.1.1. *Veri Ön İşleme (Data Pre-processing)*

Veri ön işlemede başlangıç dokümanının eğitim ve sınıflandırma süreçlerine hazırlanması sağlanır. Veri ön işleme veri üzerinde bulunabilen problemleri çözmek, verinin doğal yapısını öğrenerek daha anlamlı ve kaliteli analiz yapabilmek ve veriden daha anlamlı bilgi üretebilmek gibi amaçlar için yapılır [18]. Veri ön işleme adımlarından sonra metin bir diziye aktarılır. Veri ön işleme adımları sırasıyla açıklanmıştır.

1) *Dönüştürme:* İnternette dokümanlar genellikle HTML, XML gibi çeşitli tiplerde tutulduğundan bunları düz metin haline dönüştürmek gerekmektedir. Bu aşamada metinler HTML ve XML etiketlerden temizlenir.

2) *Tarama ve İşaretleme:* Bu adım, metin içindeki terimleri ayıklamak için yapılır. Metin simgeler ya da noktalama işaretleri ile kelime ya da ifadelere ayrılır [19] ardından küçük harfe çevrilir.

3) *Durak Kelimelerin Çıkarılması:* Metin içerisinde çok sık geçen fakat sınıflandırmada bir anlam ifade etmeyen edat, bağlaç ve zamir gibi kelimeler metinden çıkartılır.

4) *Kök Bulma:* Aynı kökten gelen farklı ek almış kelimelerin doküman içerisindeki kelime sıklıklarına

bakılırken aynı kelime olarak algılanması için köklerinin bulunması gerekmektedir.

2.1.2. Anahtar Kelimelerin Oluşturulması (Forming the Keywords)

Dokümanlar arasında kıyaslama yapabilmek için anahtar kelimelerden oluşan bir sözlüğe ihtiyaç duyulmaktadır.

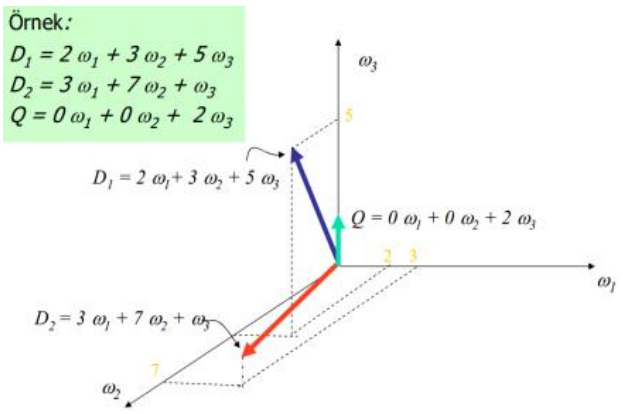
Anahtar kelime sözlüğü iki şekilde oluşturulur:
i. Belirli kelimelerin seçilmesi [11]. Burada dikkat edilmesi gereken, anahtar kelimelerin sınıflara özel olarak, kendisini diğer sınıflardan ayıracak şekilde seçilmeye çalışılmasıdır. Anahtar kelimeler o konunun uzmanları ile görüşülerek belirlenirse nispeten daha sağlıklı sonuçlar elde edilebilir.
ii. Metinde geçen tüm kelimelerin seçilmesi [20].

Bu iki yöntemde de kelimeler statik olarak seçilmiş olmaktadır. Bu çalışmada kelimeleri dinamik olarak seçmek, kullanıcının yanlış / eksik kelime seçmesini engellemek ve tüm kelimeleri alarak çok boyutlu bir uzayda çalışmayı engellemek amacıyla farklı bir yöntem ile oluşturuldu. Bu yöntem uygulama bölümünde anlatılacaktır.

2.1.3. Vektör Uzay Modeli (Vector Space Model)

Vektör uzay modeli bilgi çıkarımı, bilgi filtreleme, indeksleme gibi alanlarda kullanılan matematiksel bir modeldir ve doğal dil belgelerinin çok boyutlu uzayda özel bir anlamını simgelemektedir [11]. Metinleri vektörlerle olarak ifade edebilmek için öncelikle uzay eksenlerinin belirlenmesi gerekir. Uzay eksenleri o sınıfın belirleyici kelimelerinden oluşur.

Dokümanlar kelimelerin vektörleri olarak ifade edilir (Şekil 2). Eksenler (w_i) kelimeleri ifade etmektedir [21]. D_i eğitim dokümanlarını, Q ise sorgu dokümanını belirtmektedir.



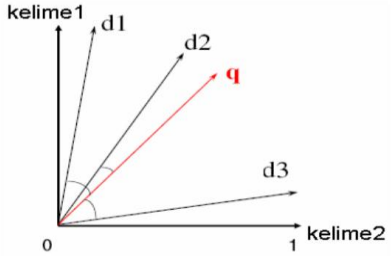
Şekil 2. Vektör uzay modeli (Vector space model) [21]

Vektör uzay modelinde kelimeler ve dokümanlar i - j boyutlu bir uzayda gösterilir. Her bir boyut bir kelimeye karşılık gelmektedir. Bu şekilde her doküman bir vektör olarak gösterilebilir. Örnek bir Doküman-Kelime Tablosi Tablo 1'de gösterilmiştir. i incelenen kelime sayısıdır. j ise incelenen doküman sayısıdır (Tablo 1).

Tablo 1. Örnek doküman-kelime Tablosi
(Example document – word table)

	Kelime 1	**...**	**Kelime i**
Doküman 1	Kelime Sıklığı	…	Kelime Sıklığı
Doküman 2	Kelime Sıklığı	…	Kelime Sıklığı
...	Kelime Sıklığı	…	Kelime Sıklığı
Doküman j	Kelime Sıklığı	…	Kelime Sıklığı

Vektör uzay modelinde dokümanların nasıl göründüğü gösterilmiştir (Şekil 3).



Şekil 3. Vektör uzay modelinde dokümanlar
(Documents in vector space model) [11]

2.1.4. Dokümanlar Arası Benzerlik (Document Similarity)

Dokümanlar arasındaki ilişki bulunmak istendiğinde, her bir eğitim vektörü ile sorgu vektörü arasındaki açının *kosinüs değeri* hesaplanır ve sıralanır. İki belge arasındaki benzerlik şu şekilde ifade edilir: $s(d,d') \in R$. İki vektör arasındaki açının kosinüsü verilmiştir (Eşitlik 1) [21].

$$\cos(d_1,d_2) = d_1 \bullet d_2 / \| d_1 \| \| d_2 \| \quad (1)$$

$d_1 \bullet d_2$: d_i dokümanlarının uzunlukları çarpımı

$\| d_i \|$: d_i dokümanının uzunluğu

2.1.5. Sonuçların Değerlendirilmesi (Evaluation of Results)

Eğitim dokümanlarının ve sorgu dokümanının (Q) vektör uzay modeli bulunur. Kullanıcı sorgusunun, her bir eğitim dokümanıyla olan benzerliği hesaplanır. Tüm dokümanlar vektörel olarak temsil edilir (Eşitlik 2).

$$d_i = (wd_{i1}, wd_{i2}, ...wd_{ij},) \quad (2)$$

d_i : terimin doküman içerisindeki ağırlığı
w_{ij} : eğitim dokümanı vektörüdür.
q : sınıfının bulunması istenen vektör

Sorgu dokümanı ile diğer dokümanlar arasındaki kosinüs benzerliği hesaplanır **(**Eşitlik 3, 4) [22].

$$sim(d_i, q) = \cos\theta \tag{3}$$

$$sim(d_i, q) = \frac{d_i * q}{|d_i \| q|} = \frac{\sum_j (w_{i,j} * w_{q,j})}{\sqrt{\sum_j w_{i,j}^2} \sqrt{\sum_j w_{q,j}^2}} \tag{4}$$

Eğer iki dokümanın hiç bir ortak kelimesi yoksa sonuç değeri 0 olacaktır ($sim(d_i, q) = 0$). İki dokümanın tüm kelimeleri ortaksa (iki dokümanda da anahtar kelimelerin tamamı geçiyorsa) sonuç değeri 1 olacaktır ($sim(d_i, q) = 1$). İşlem sonucunda çıkan değerler 0 ile 1 arasında olacaktır ve yüksek olan değerler aranacaktır [23]. Benzerlik oranları karşılaştırılır. En büyük benzerlik oranı dikkate alınarak sorgu vektörü ilgili sınıfa atanır.

Metin Sınıflandırma Algoritmaları (Text Classification Algorithms)

Bu kısımda metin sınıflandırma algoritmalarından yaygın olarak kullanılan [7-9] mesafeye dayalı sınıflandırma yöntemlerinden K-En-Yakın-Komşu (k-NN) Algoritması ve istatistiğe dayalı sınıflandırma yöntemlerinden Naïve Bayes algoritması anlatılmıştır.

K-En-Yakın-Komşu Algoritması (K-Nearest Neighbour Algorithm)

K-En-Yakın-Komşu Algoritması içinde farklı sınıftan elemanların bulunduğu bir gruptan sorgulamak istediğimiz dokümanın sınıfına benzeyen sınıfları bulur. En Yakın Komşu araması, uzaklık bakımından kendisine en yakın olanı noktayı/noktaları bulmaya yarar [24].

Yeni bir dokümanı sınıflandırmak için, eğitim setinde onun "komşu"su olarak adlandırılan dokümanlar kontrol edilir. Bu karşılaştırma yeni doküman ile her bir doküman arasındaki uzaklığa göre yapılır. Karşılaştırma sonucunda yeni doküman kendisine en yakın olan k sayıdaki (önceden eğitilmiş) dokümandan sınıfça çoğunlukta olanın sınıfına atanır. k sayısı dikkate alınacak en yakın komşu sayısıdır [23].

2.1.6. Naïve Bayes Algoritması (Naïve Bayes Algorithm)

Naïve (Saf) Bayes Algoritması istatistiğe dayalı Bayes Sınıflandırma Algoritmalarından birisidir. Olasılık tabanlıdır. Bu algoritmaya göre kelimeler sınıftan bağımsızdır [25]. Uygulanabilirliği kolaydır ve performans bakımından başarılıdır [26]. (Eşitlik 5, 6, 7)

$$p(d \mid c_j) = p(|d|)\,|d|!\prod_{t=1}^{|V|} \frac{p(w_t \mid c_j)^{x_t}}{x_t!} \tag{5}$$

$$p(w_t \mid c_j) = \frac{1 + N_{jt}}{|V| + V_j} \tag{6}$$

$$M(C) = P(word1|C)^{n1} P(word2|C)^{n2} ... P(wordv|C)^{nv} P(|C) \tag{7}$$

d : sınıf sayısı
N_{jt}: j sınıfındaki dokümanlar içinde t kelimesinin görülme sıklığı
N_j : j sınıfındaki toplam kelime sayısı
$P|d|$: kategori olasılığı
x_t : kelimenin frekansı
$|V|$: sözlükteki kelime sayısı

Sorgu dokümanı M(C) değeri en büyük olan sınıfa atanır.

3. UYGULAMA (IMPLEMENTATION)

Çalışmada öncelikle dizin sitelerinden[1] eğitim verisi olarak kullanılmak üzere site isimleri toplanmıştır. Ardından Java programlama dili ile yazılan kod parçacıklarıyla toplanan sitelere bağlanılıp site içerikleri elde edilmiştir. Veri ön işleme aşamaları gerçekleştirilerek veriler etiketlenmiştir (sınıf belirleme). Daha sonra denetimli öğrenme tekniklerinden K-En-Yakın-Komşu ve Naïve Bayes algoritmaları kullanılmıştır. Ardından algoritmaların eğitim verisi üzerindeki başarıları karşılaştırılmış ve model kurulmuştur. Modelin girdi olarak kullanılacağı bir ara yüz hazırlanmıştır. Bu ara yüz ile kullanıcıların istedikleri siteleri test ederek ilgili sitenin e-ticaret sitesi olup olmadığının bildirilmesi sağlanmıştır.

İnternet sayfalarını sınıflandırma metin, resim, doküman yapısı gibi birçok parametreye bağlı olarak gerçekleştirilebilmektedir. İnternet sayfalarının sınıflandırması şu şekilde kategorilere ayırılmıştır:

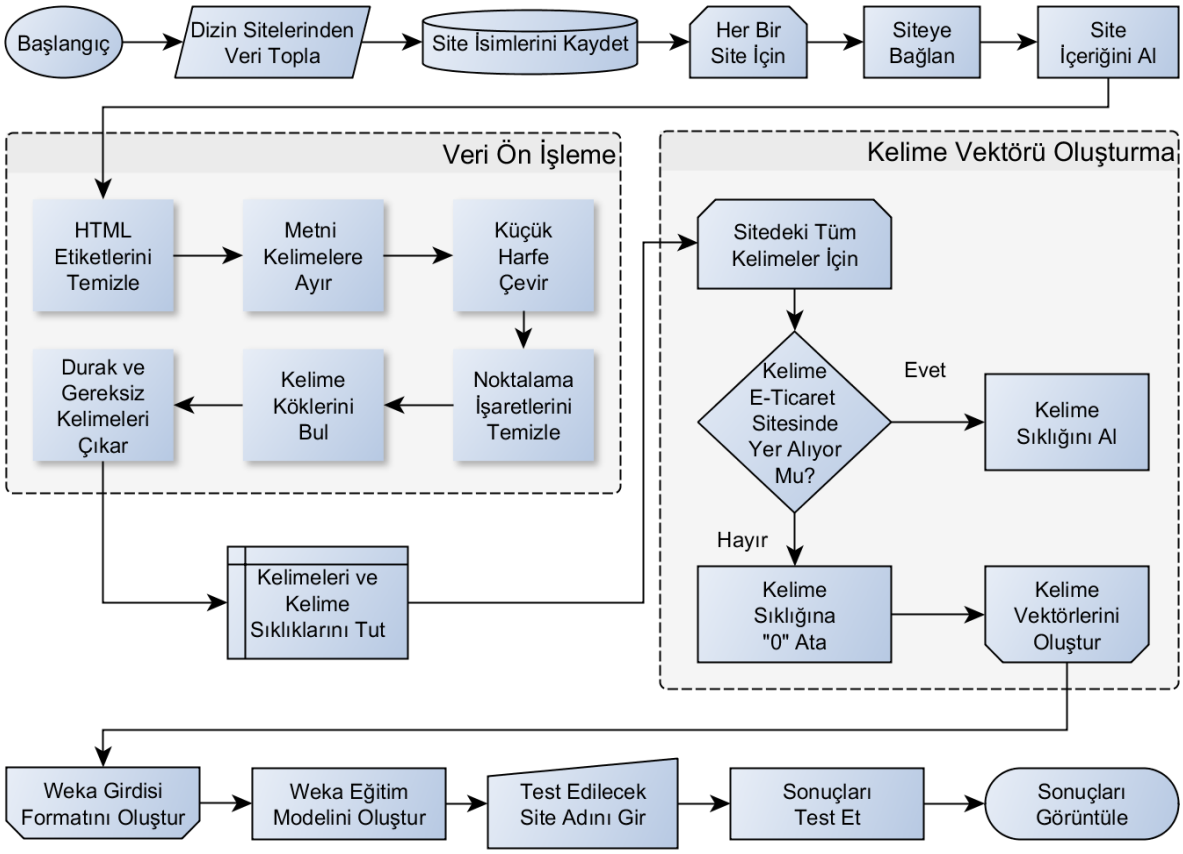
1. Alan adı uzmanlarınca yapılan elle sınıflandırma
2. Kümeleme yaklaşımları ile sınıflandırma
3. Meta etiketleri[2] ile sınıflandırma
4. Doküman içeriği ve meta etiketleri ile sınıflandırma
5. Sadece doküman içeriğine göre ile sınıflandırma
6. Bağlantı ve içerik analizine göre sınıflandırma [29].

Bu çalışma 4. Madde olan doküman içeriği ve META etiketlerini dikkate alarak yapılmıştır. Çalışmada iki sınıf bulunmaktadır:
1. "elektronik ticaret sitesi olan siteler" sınıfı (olumlu) ve
2. "elektronik ticaret sitesi olmayan siteler" sınıfı (olumsuz).

Bu çalışmada 4 dizin sitesinin "alışveriş" kategorisinde yer alan siteler ele alınmıştır. Bu dizin siteleri şunlardır: "www.alisverisrehberi.com", "www.dizin.sitesi.web.tr", "www.dmoz.org", "www.trdizin.org". Dizin sitelerinden toplanan siteler incelendiğinde, e-ticaret sitesi olarak eklenmiş olan bazı sitelerin gerçekte sipariş verme, ödeme yapma gibi e-ticaret sitesi özelliklerini [30] taşımadığı, bazılarınınsa sadece reklam amaçlı olduğu gözlemlenmiştir.

[1] Dizin siteleri kullanıcıların aradıkları bilgileri kolayca bulmasını amaçlayan sitelerdir ve siteleri elle veri tabanına eklenir [27].
[2] İnternet sitesi hakkında sayfa bilgisi, anahtar kelimeler, telif hakkı gibi yapısal bilgilerin tutulduğu kısımdır [28].

Doğru veriler ile yola çıkmanın veri madenciliğinin en önemli aşamalarından biri olduğu [31] göz önünde bulundurularak 309 adet site tek tek incelenmiştir. Bunlardan 36 tanesine sayfanın kullanımda olmaması nedeniyle bağlantı sağlanamamıştır. 116 tanesinin gerçekten e-ticaret sitesi olduğu, 157 tanesinin e-ticaret sitesi olmadığı, görülmüştür. Bu veriler doğrultusunda 116 siteden "e-ticaret sitesi" sınıfı, 157 siteden "e-ticaret sitesi değil" sınıfı oluşturulmuştur. Bu site isimleri bir metin dosyasına kaydedilmiştir. Eğitim seti verisini elde etmede Java programla dili kullanılmıştır. Uygulama iş akışı gösterilmiştir (Şekil 4).



Şekil 4. Uygulama iş akışı (Implementation workflow)

3.1. Veri Elde Etme (Obtaining Data)

Toplanan 273 adet sitenin ana sayfa içerikleri veri olarak kullanılmıştır. JSoup adlı Java kütüphanesini kullanarak yazılan Java kodu ile metin dosyasındaki site adları okunarak siteye bağlanılmış ve ana sayfa içerikleri alınmıştır.

3.2. Doküman Ön İşleme (Data Pre-processing)

Sitelerin (dokümanların) metin şeklinde gösterilebilmesi için Jsoup ve Zemberek [32] adlı açık kaynak kodlu Türkçe Doğal Dil İşleme Java Kütüphanelerinden yararlanılmıştır. Zemberek yazım denetimi, hatalı kelimeler için öneri, heceleme, hatalı kodlama temizleme gibi işlevlere sahiptir [33]. Bu kütüphaneler kullanılarak yazılan Java kodu ile sitelere bağlanılıp site içerikleri alınmıştır. Ardından doküman ön işleme aşamaları olan *dönüştürme, tarama ve işaretleme*, *kök bulma ve durak ve gereksiz kelimelerin çıkarılması* işlemleri yapılmıştır. Bu bağlamda metin içerisindeki noktalama işaretleri kaldırılmış, metnin tamamı küçük harfe çevrilerek kelimelere ayrılmış ve içerisindeki HTML etiketleri temizlenmiştir. Metin içerisinde çok sık geçen fakat sınıflandırmada bir anlam ifade etmeyen edat, bağlaç ve zamir gibi kelimeler ile Türkiye'ye ait şehir isimleri metinden çıkartılmıştır. Ardından kelime kökleri bulunmuştur.

3.3. Anahtar Kelimelerin Oluşturulması (Forming the Keywords)

Bu aşamada öncelikle e-ticaret sitelerinde sıkça geçtiğini gözlemlenen 30 adet anahtar kelime seçildi; "alışveriş", "banka", "ekle", "fırsat", "fiyat", "gönder", "haberdar", "hesap", "hizmet", "iade", "incele", "kampanya", "kargo", "katalog", "kdv", "marka", "müşteri", "öde", "promosyon", "sat", "sepet", "sipariş", "sözleşme", "teslim", "ticaret", "tl", "toptan", "ucuz", "ücret", "ürün". Fakat yapılan incelemeler sonucunda gözden kaçan kelimelerin olabileceği görüldü.

Ardından e-ticaret sitelerinin tamamında geçen kelimeler ele alınarak anahtar kelime sözlüğü oluşturuldu. Fakat bu sefer kelime boyutu çoğaldı ve bir sitede çok sayıda geçen

fakat diğerinde hiç geçmeyen ya da yazım hatası yapılarak farklı şekilde algılanan kelimeler de sözlüğe alınmış oldu. Bu nedenle anahtar kelimelerin kullanılan test verisinde geçen kelimelere bağlı olarak geliştirilen program tarafından otomatik olarak seçilmesine karar verildi.

Bu amaçla elde edilen kelime kökleri bir araya getirildi. Ardından tüm e-ticaret sitelerinin %10'undan azında geçen kelimeler elendi. %10 oranı isteğe göre değiştirilebilmektedir. E-Ticaret sitelerinin %90'ında ortak olarak geçen kelimeler dikkate alınarak vektör oluşturma aşamasında kullanılmak üzere Anahtar kelime sözlüğü oluşturuldu. Bu şekilde yazım yanlışlarının ve özel isimlerin sözlüğe alınmasının önüne geçildi.
Eğitim verisi olarak kullandığım 273 siteden e-ticaret sitelerinde geçen tekil kelime sayısı 4495 iken, en az 28(>273*0,10) e-ticaret sitesinde birden geçen kelime sayısı 110 olarak bulunmuştur. Anahtar kelime sözlüğü bu 110 kelime ile oluşturulmuştur. Böylece anahtar kelimeler önceden belirlenmemiş, eğitim verisindeki e-ticaret sitelerindeki ortak kelimeler ele alınmıştır.

3.4. Vektör Uzay Modelinin Oluşturulması
(Forming the Vector Space Model)

Tüm sitelerde anahtar kelimelerin kaçar kez geçtiği bilgisi sayılarak 273 site için doküman vektörleri oluşturuldu.

3.5. Dokümanlar Arası Benzerlik ve Sonuçların Değerlendirilmesi (Document Similarity and Evaluation of Results)

Metin madenciliği algoritmalarının kullanılması için açık kaynak kodlu WEKA Java Kütüphanesi kullanılmıştır. Doküman vektörleri algoritmalarda kullanılmak üzere WEKA'nın kabul ettiği ".arff" uzantılı dosya formatında kaydedilmiştir. Bu eğitim verileri ile k-NN En Yakın Komşu ve Naïve Bayes sınıflandırma algoritmaları kullanılmıştır. k-NN En Yakın Komşu algoritmasında "k" değeri için "1, 3, 5, 7, 9, 11, 13 ve 15" değerleri denenmiştir. En başarılı sonucun k = 3 olduğu durumda elde edildiği görülmüştür (Şekil 5).

Şekil 5. "k" değerine bağlı başarı yüzdeleri
(Success depends on "k" value)

Algoritma sonuçları karşılaştırıldığında (Tablo 2) Naïve Bayes algoritmasının (%85.3047) k-NN En Yakın Komşu (k = 3) algoritmasına (%83.871) göre daha iyi sonuç verdiği görülmüştür.

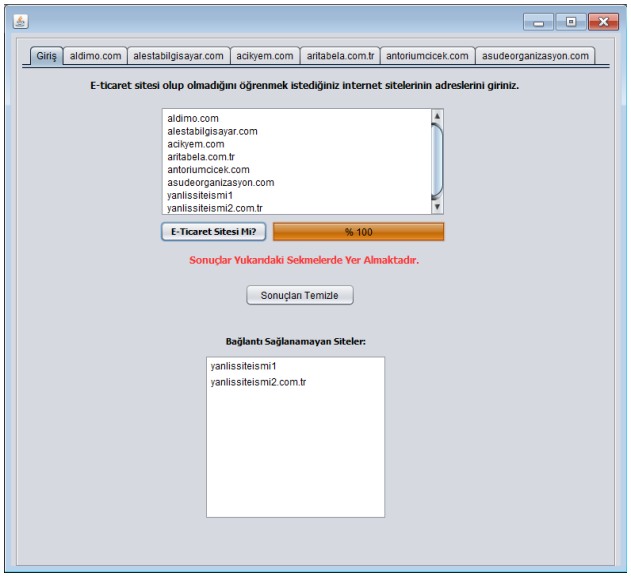
Tablo 2. Algoritma sonuçları (Algorithms results)

		Doğru Sınıflanan Doküman		Yanlış Sınıflanan Doküman	
		Sayısı	Yüzdesi	Sayısı	Yüzdesi
Algoritma	k-NN En Yakın Komşu	234	83.8710	45	16.1290
	Naïve Bayes	238	85.3047	41	14.6953

Bu sonuçlara dayanarak Naïve Bayes algoritması kullanılarak bir eğitim modeli oluşturulmuştur. Ardından bu modelin girdi olarak kullanılacağı bir uygulama ara yüzü geliştirilmiştir. Söz konusu model kullanılarak ara yüz ekranına girilecek sitelerin hangi sınıfa dâhil olduğu sonucuna ulaşılır.

4. UYGULAMA ARAYÜZÜ (APPLICATION INTERFACE)

Uygulama ara yüzü Java programlama dili ve Swing Kütüphanesi kullanılarak yapılmıştır (Şekil 6). Kullanıcının uygulamayı kolay bir şekilde kullanabilmesi için sade bir ara yüz tasarlanmıştır. Giriş ekranından sorgulanmak istenilen sitelerin isimleri alındıktan sonra arka planda daha önce anlatılmış olan metin madenciliği işlemlerine ve oluşturulan modele göre sorgulama yapılmakta ve ardından sorgulama sonuçları ekrana gelmektedir.



Şekil 6. Uygulama giriş ekranı (Application home screen)

Programın kullanımı şu şekildedir:

- Giriş ekranındaki alana e-ticaret sitesi olup olmadığı öğrenilmek istenen internet sitelerinin adresleri yazılır (Şekil 7).

Şekil 7. Sorgu giriş ekranı (Application query screen)

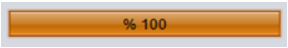
- "E-Ticaret Sitesi Mi?" düğmesine basıldığı zaman girilen siteler önceden kurulan modele göre değerlendirilir (Şekil 8).

Şekil 8. "E-Ticaret Sitesi Mi?" düğmesi
(Button of "E-Ticaret Sitesi Mi?")

- Listede yanlış yazım ya da sitenin aktif olmaması gibi sebeplerden dolayı bağlanılamayan siteler varsa bunlar "Bağlantı Sağlanamayan Siteler" adı altında oluşturulacak listede görüntülenir (Şekil 9).

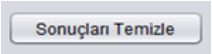
Şekil 9. Bağlantı sağlanamayan sitelerin listesi
(List of the connected sites)

- Ekranda işlemin devam ettiğini/bittiğini gösteren durum çubuğu yer almaktadır (Şekil 10).



Şekil 10. Durum Çubuğu (Status bar)

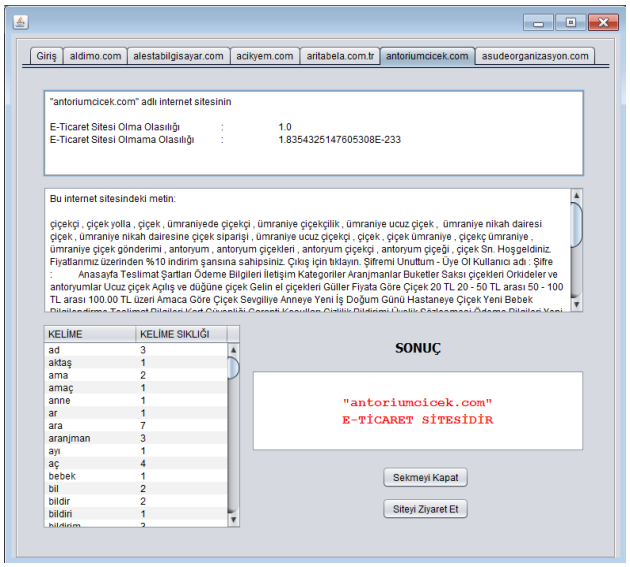
- "Sonuçları Temizle" düğmesine tıklandığı zaman şu işlemler gerçekleştirilir:
  - Açık olan sekmeler kapanır,
  - "Bağlantı Sağlanamayan Siteler" listesi temizlenir,
  - "Bağlantı Sağlanamayan Siteler" listesi kalkar ve
  - Durum çubuğu sıfırlanır.



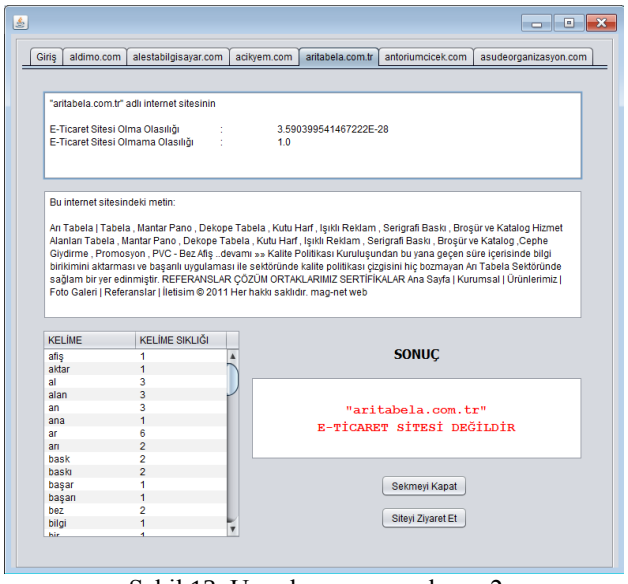
Şekil 11. "Sonuçları Temizle" Düğmesi
(Button of "Sonuçları Temizle")

Uygulama bağlantı sağlanan ve işlemleri yapılan her bir site için sonuç verilerini içeren yeni bir sekme oluşturur. Bu sekmelerde girilen adrese ait şu bilgiler yer almaktadır (Şekil 12, 13):
- Siteye yönlendiren bir düğme,
- Sitede yer alan metin,
- Sitenin e-ticaret sitesi olma olasılığı,
- Sitenin e-ticaret sitesi olmama olasılığı ve
- Sonuç (sitenin e-ticaret sitesi olup olmadığı bilgisi).

Sonuca ulaşmak için daha önce anlatılan işlemleri yapan Java kodu çalışmaktadır.



Şekil 12. Uygulama Sonuç Ekranı 1
(Application result screen 1)



Şekil 13. Uygulama sonuç ekranı 2
(Application result screen 2)

5. SONUÇ VE ÖNERİLER(CONCLUSION & SUGGESTIONS)

Bu çalışmada kullanıcı tarafından belirtilen internet sitelerinin içeriğini analiz ederek metin madenciliği yöntemleri ile bu sayfaların elektronik ticaret sitesi olup olmadığına karar veren bir ara yüz hazırlanmıştır. Kullanılan metin sınıflandırma algoritmalarından Naïve Bayes algoritmasının k-NN En Yakın Komşu algoritmasına göre daha iyi sonuç verdiği gözlemlenmiştir.

Uygulamanın kullanım alanları arasında dizin sitelerinde ilgili kategori altına yapılacak ekleme sırasında kategori önerme (mevcut durumda e-ticaret kategorisi), bankalar

tarafından düşük faizli kredi ve GSM operatörleri tarafından uygun GSM tarifesi gibi kampanyalar için hedef kitlesi olarak kullanılacak e-ticaret sitelerinin tespiti olabilir.

Bu uygulama e-ticaret sitelerinin sınıflandırılması için yapılmış olmasına rağmen, veri setleri farklı kategorilerden olacak şekilde seçildiği takdirde farklı konulardaki sınıflandırma işlemlerini de yapabilmektedir. Uygulamanın etkinliğinin artırılması için veri seti çoğaltılmalıdır. Ayrıca eğitim modelini oluşturma sırasında *Özellik Çıkarımı* yapılarak referans sözlüğünde yer alan kelimeler içerisinden daha az öneme sahip olanlar çıkartılarak modelin iyileştirilip iyileştirilemeyeceğinin araştırılması gerekmektedir.

İlerleyen çalışmalarda programın "Giriş" ekranına ek olarak "Modeli Eğitme" ekranı tasarlanarak kullanıcının veri setine ekleme yapması sağlanabilir. Ayrıca programın başlangıç kısmına bir arama motoru eklenebilir. Arama motorunun bulduğu siteler kullanıcıya sunularak kullanıcıdan sınıf bilgisinin girilmesini isteyebilir. Bu veriler de eğitim verisine eklenir ve model tekrar eğitilebilir. Bu şekilde programın daha dinamik olması sağlanmış olur.

KAYNAKLAR (REFERENCES)

[1] İnternet: İnternet, http://tr.wikipedia.org/wiki/%C4%B0nternet/, 08.01.2013.

[2] İnternet: H., Takcı, Metin Madenciliği Çerçevesi, http://verimadencisi.blogspot.com/2012/10/metin-madenciligi-cercevesi.html/, 08.01.2013.

[3] İnternet: Doğal Dil İşleme, http://tr.wikipedia.org/wiki/Doğal_dil_işleme/, 08.01.2013.

[4] İnternet: Veri Madenciliği, http://tr.wikipedia.org/wiki/Veri_madencili%C4%9Fi/, 08.01.2013.

[5] C., Coşkun, A., Baykal, "Veri Madenciliğinde Sınıflandırma Algoritmalarının Bir Örnek Üzerinde Karşılaştırılması", **Akademik Bilişim 2011**, İnönü Üniversitesi, Malatya, 02-04 Şubat, 2011.

[6] G., Silahtaroğlu, F., Demircan, "Çeviri Yazılımlarında Sözcüklerin Bağlam İçindeki Anlamını Algılamaya Yönelik Öneri", **Akademik Bilişim 2013**, Akdeniz Üniversitesi Hukuk Fakültesi, Antalya, 23-25 Ocak, 2013.

[7] G., Guo, H., Wang, D., Bell, Y., Bi, K., Greer, "Using kNN Model for Automatic Text Categorization", *Soft Computing*, 10 (5), 423-430, 2006.

[8] Z., Deng, M., Zhang, "Improving Text Categorization Using the Importance of Words in Different Categories", **Computational Intelligence and Security**, Xi'an, Çin, 458-463, 15-19 Aralık, 2005.

[9] S., Manne, S. K., Kotha, And S.S., Fatima, "Text Categorization with K-Nearest Neighbor Approach", **Information Systems Design and Intelligent Applications 2012**, Visakhapatnam, Hindistan, 413-420, Ocak 2012.

[10] Y. K., AKIN, **Veri Madenciliğinde Kümeleme Algoritmaları ve Kümeleme Analizi**, Doktora Tezi, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, 2008.

[11] İ. F., Pilavcılar, **Metin Madenciliği ile Metin Sınıflandırma**, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, FBE Matematik Mühendisliği Anabilim Dalı, 2007.

[12] H. K., Yıldız, M., Gençtav, N., Usta, B., Diri, M. F., Amasyalı, "Metin Sınıflandırmada Yeni Özellik Çıkarımı", **15. Sinyal İşleme, İletişim ve Uygulamaları Kurultayı**, Eskişehir, 11-13 Haziran, 2007.

[13] V. C., Gandhi, J. A., Prajapati, "Review on Comparison between Text Classification Algorithms", *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 1 (3), 2012.

[14] S. M., Weiss, C., Apte, F. J., Damerau, D. E., Johnson, F. J., Oles, T., Goetz, T., Hampp, "Maximizing Text-Mining Performance" *IEEE Intelligent Systems and Their Applications*, New York, 14(4), 63-69, 1999.

[15] K., Wu, B. L., Lu, M., Utiyama, H., Isahara, "An Empirical Comparison of Min–Max-Modular K-NN with Different Voting Methods to Large-Scale Text Categorization", *Soft Computing*, 12(7), 647-655, 2008.

[16] A., Kehagias, V., Petridis, V.G., Kaburlasosand, P., Fragkou, "A Comparison of Word and Sense-based Text Categorization Using Several Classification Algorithms", *Journal of Intelligent Information Systems*, 21(3), 227-247, 2001.

[17] İnternet: S., Albayrak, Veri Madenciliği Sınıflama ve Kümeleme Yöntemi, http://www.ce.yildiz.edu.tr/personal/songul/file/332/Veri+Madencili%C4%9Fi-S%C4%B1n%C4%B1flamaKumeleme.ppt, 17.03.2013.

[18] U., İlhan, **Application Of KNN and FPTC Based Text Categorization Algorithms to Turkish News Reports**, Doktora Tezi, Bilkent Üniversitesi, Mühendislik Fakültesi, 2001.

[19] İnternet: C., Janssen, Tokenization, http://www.techopedia.com/definition/13698/tokenization/, 08.01.2013.

[20] K., Çalış, O., Gazdağı, O., Yıldız, "Reklam İçerikli Epostaların Metin Madenciliği Yöntemleri ile Otomatik Tespiti", *Bilişim Teknolojileri Dergisi*, 6(1), 1-7, 2013.

[21] İnternet: Ş.G., Öğüdücü, "Metin Madenciliği", http://ninova.itu.edu.tr/tr/dersler/bilisim-enstitusu/195/bbl-606/ekkaynaklar?g33056/, 21.04.2013.

[22] C. D., Manning, H., Schütze, **Foundations of Statistical Natural Language Processing**, 2, MIT Press, Londra, İngiltere, 1999.

[23] W., Wang, **An Empirical Study on Hierarchical Text Categorization**, Yüksek Lisans Tezi, Guelph Ünviversitesi, The Faculty of Graduate Studies, 2007.

[24] İnternet: Nearest Neighbor Search, http://en.wikipedia.org/wiki/Nearest_neighbor_search/, 02.07.2013.

[25] S., Eyheralendy, D., Lewis, D., Madigan, "On the Naive Bayes Model for Text Categorization", **Ninth International Workshop on Artificial Intelligence and Statistics**, Florida, Amerika Birleşik Devletleri, 3-6 Ocak, 2003.

[26] K.A. Vidhya, G. Aghila, "A Survey of Naïve Bayes Machine Learning approach in Text Document Classification", *International Journal of Computer Science and Information Security (IJCSIS)*, 7(2), 206-211, 2010.

[27] İnternet: Dizin Sitesi Nedir?, http://sarkos.net/dizin-sitesi-nedir/, 09.07.2013.

[28] İnternet: Meta Tag, http://www.r10.net/meta-tag/206346-meta-tag-listesi.html/, 05.07.2013.

[29] S., Shibu, A., Vishwakarma, N., Bhargava, "A Combination Approach for Web Page Classification Using Page Rank and Feature Selection Technique", *International Journal of Computer Theory and Engineering*, 2(6), 897-900, 2010.

[30] İnternet: E-Ticaret Genel Özellikleri, http://www.mpluseticaret.com/eticaret_genel_ozellikleri.asp/, 08.08.2013.

[31] S., TÜZÜNTÜRK, "Veri Madenciliği ve İstatistik.", *Uludağ Üniversitesi İİBF Dergisi*, 29(1), 65-90, 2010.

[32] İnternet: Zemberek, http://code.google.com/p/zemberek/downloads/list, 08.01.2013.

[33] İnternet: Zemberek Yazılımı, http://tr.wikipedia.org/wiki/Zemberek_(yaz%C4%B1l%C4%B1m)/, 08.01.2013.