FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİ TEKNİKLERİYLE TÜRKÇE WEB SAYFALARININ KATEGORİZE EDİLMESİ

YÜKSEK LİSANS TEZİ

Seçil ŞEKERCİ HÜSEM

Anabilim Dalı: Bilgisayar Mühendisliği

MAYIS 2017

FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİ TEKNİKLERİYLE TÜRKÇE WEB SAYFALARININ KATEGORİZE EDİLMESİ

YÜKSEK LİSANS TEZİ

Seçil ŞEKERCİ HÜSEM
(150221001)

Anabilim Dalı: Bilgisayar Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. Ayla GÜLCÜ

Teslim Tarihi: 11 Mayıs 2017

FSMVÜ, Mühendislik ve Fen Bilimleri Enstitüsü'nün 150221001 numaralı Yüksek Lisans Öğrencisi Seçil ŞEKERCİ HÜSEM, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "VERİ MADENCİLİĞİ TEKNİKLERİYLE TÜRKÇE WEB SAYFALARININ KATEGORİZE EDİLMESİ" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Yrd. Doç. Dr. Ayla GÜLCÜ** .....
Fatih Sultan Mehmet Vakıf Üniversitesi

Jüri Üyeleri : **Yrd. Doç. Dr. Ayla GÜLCÜ** .....
Fatih Sultan Mehmet Vakıf Üniversitesi

Prof. Dr. Ali Yılmaz ÇAMURCU .....
Fatih Sultan Mehmet Vakıf Üniversitesi

Doç. Dr. Turgay Tugay BİLGİN .....
Bursa Teknik Üniversitesi

Teslim Tarihi : **11 Mayıs 2017**
Savunma Tarihi : **14 Haziran 2017**

Anneme ve eşime,

ÖNSÖZ

Tez çalışmam boyunca desteğini ve yardımını esirgemeyen, değerli bilgileriyle beni her zaman yönlendiren danışmanım Sayın Yrd. Doç. Dr. Ayla GÜLCÜ'ye;

Engin bilgi ve tecrübeleriyle beni aydınlatan Sayın Prof. Dr. Ali Yılmaz ÇAMURCU'ya;

Lisans öğrenimimden bu yana yanımda olan ve desteğini her zaman hissettiren, can yoldaşım Hürkal HÜSEM'e;

Tüm öğrenim hayatım boyunca maddi ve manevi desteği ile her zaman yanımda olan annem Handan ÖZATEŞ'e ve aileme;

… çok teşekkür ederim.

Mayıs 2017

Seçil ŞEKERCİ HÜSEM
Bilgisayar Mühendisi

İÇİNDEKİLER

Sayfa

KISALTMALAR

BKO : Bilgi Kazanım Oranı
DN : Doğru Negatif
DP : Doğru Pozitif
DVM : Destek Vektör Makineleri
KNN : k-Nearest Neighbor
M-NB : Multinomial Naive Bayes
NB : Naive Bayes
RDF : Resource Definition Framework
YP : Yanlış Pozitif
YN : Yanlış Negatif
W3C : World Wide Web Consortium
WEKA : Waikato Environment for Knowledge Analysis
XML : eXtensible Markup Language

SEMBOLLER

μ	**:** Mikro-ortalamalar
M	**:** Makro-ortalamalar
wa	**:** Ağırlıklı ortalama
$\{\boldsymbol{A_1}, \dots, \boldsymbol{A_p}\}$	**:** Özellikler
m	**:** Mevcut sınıf sayısı
I(X)	**:** Bilgi kazanımı
$\boldsymbol{p_j}$	**:** Sınıfın görülme olasılığı
$\boldsymbol{\lvert X_i \rvert}$	**:** BKO'da örnekler içerisinde A_k özniteliğinin değerleri
$\boldsymbol{\lvert X \rvert}$	**:** BKO'da mevcut örneklerin sayısını
wn	**:** Kelime vektörünü oluşturan sözlükteki kelimeler
$\boldsymbol{\overrightarrow{W}}$	**:** M-NB'de kelime vektörü, DVM'de ağırlık vektörü
$\boldsymbol{d_n}$	**:** *n*. doküman
cn	**:** Sınıflar
$\boldsymbol{\vec{x}}$	**:** DVM'de kullanılan veriler
$\boldsymbol{b}$	**:** Öğrenilen sabit değer
$\boldsymbol{\tilde{S}_1},\ \boldsymbol{\tilde{S}_2},\ \boldsymbol{\tilde{S}_3}$	**:** Destek vektörleri
α1, α2 ve α3	**:** Destek vektörlerinin ağırlıklandırma değeri
$\boldsymbol{\widetilde{w}}$	**:** Hiper-düzlem
P	**:** Hassasiyet (Precision)
R	**:** Duyarlılık (Recall)
F	**:** f-değeri (f-score)

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

VERİ MADENCİLİĞİ TEKNİKLERİYLE TÜRKÇE WEB SAYFALARININ KATEGORİZE EDİLMESİ

ÖZET

Veri madenciliği, insanın işleyebileceğinden çok miktarda veri üzerinde çalışabilen, bu verileri anlamlandırmak, örtük bağlantıları ortaya çıkarmak amacıyla uygulanan yöntemler bütünüdür. Örneğin, herhangi bir web sayfasının önceden tanımlanmış kategoriler arasından hangi kategoriye ait olduğunun bulunması el ile kolaylıkla yapılabilirken sayfaların sayısı arttıkça her bir sayfanın hangi kategoriye ait olduğunun bulunması imkansız hale gelmektedir. Bu nedenle otomatik sınıflandırma tekniklerinin kullanımı gittikçe daha fazla önem kazanmaktadır. Web sayfalarının sınıflandırılmasından yola çıkılarak oluşturulan veriseti ile eğitilen bir sistemde yalnızca web sayfaları için değil metin tabanlı herhangi bir dokümanın da kategorisinin belirlenme işlemi yapılabilir. Böylece anlamsız bir şekilde bir arada bulunan veri yığınları, içeriklerine uygun kategorilere ayrılmış bir katalog haline getirilebilir.

Literatürde Türkçe web sayfalarının sınıflandırılmasıyla ilgili yapılan çalışmaların sayısı azdır. Aynı zamanda bu çalışmalar için kullanılabilecek hazır durumdaki verisetlerinin sayısı ve çeşitliliği kısıtlıdır. Bu tez çalışmasıyla hem Türkçe veriseti ihtiyacına cevap verebilmek hedeflenmiş, hem de literatürdeki metin sınıflandırma için kullanılan çeşitli yöntemler bu veriseti üzerinde denenmiştir. Bu algoritmaların çeşitli durumlardaki performansları kıyaslanarak bu alandaki çalışmalara katkı sağlamaya çalışılmıştır. Veriseti oluşturulurken el ile kategorize edilmiş web sayfalarının tutulduğu DMOZ sisteminden faydalanılmıştır. Buradan Türkçe sayfa verilerini çok kanallı yöntemle çekebilen bir web-gezer tasarlanmıştır. Elde edilen sayfa içeriklerinden sınıflandırma için anlamsız olan veriler otomatik olarak temizlenmiştir ve böylece bir Türkçe veriseti elde edilmiştir. Oluşturulan veriseti üzerinde yapılan ön işlem aşaması sırasında ya da sonrasında, kelime ekleme veya

çıkarma gibi hiçbir müdahalede bulunulmamıştır. Elde edilen veriseti benzer çalışmalara kaynak sağlayabilecek niteliktedir.

Bu çalışma kapsamında metin sınıflandırma için sıkça kullanılan Naive Bayes (NB) ve Destek Vektör Makineleri (DVM - Support Vector Machines) algoritmalarına n-gram kelime vektörü (n-gram Word Vector) seçimi ve bilgi kazanım oranı (BKO - Information Gain Ratio) yaklaşımları uygulanarak performansları karşılaştırılmıştır. Bunların yanında kategori sayısı, modeli eğitmek için kullanılan veriseti içeriği ve bu verisetinin tamlığı konularına da odaklanılmış ve farklı durumlarda algoritmaların sınıflandırma başarıları da incelenmiştir.

Deneyler sonucunda kategori sayısının azlığı sınıflandırma başarısını olumlu etkilemiştir. Eğitim verilerindeki web sayfalarının başlık, anahtar kelime ve açıklama yönünden eksiksiz olmasının sınıflandırma başarısına DMOZ verilerinden daha fazla katkı sağladığı görülmüştür. Dengesiz dağılım gösteren verisetleri üzerinde yapılan deneylerde en yüksek başarıyı gösteren ve değişimlerden en az etkilenen algoritmanın Multinomial Naive Bayes (M-NB) olduğu görülmüştür. M-NB algoritması kelime 2-gram özellik vektöründe daha yüksek sonuçlara ulaşmıştır. Buna rağmen BKO yaklaşımının M-NB algoritmasına önemli bir katkı sağlamadığı görülürken DVM algoritmasına, M-NB algoritmasına oranla daha fazla katkı sağladığı gözlenmiştir.

CATEGORIZING THE TURKISH WEB PAGES BY DATA MINING TECHNIQUES

SUMMARY

Data mining can be described as a collection of the methods that are able to work on large-scale data, extract meaningful information and discover hidden patterns from the data. For example, identifying the category of a given web page is a data mining job. Although it seems to be quite easy job to determine the category of a given web page manually; it happens to be impossible to do by hand as the number of these web pages increases. Nowadays, the use of data mining techniques to automatically place web pages into predefined categories has become more important. Additionally, a system that has been trained to classify web pages using a given text dataset can also be used to classify all other text documents. Classification can convert piles of text data into categorized documents.

As far as we are aware, there are only a few studies in the literature in which text classification methods have been applied on Turkish text data. In addition, there is lack of proper Turkish dataset in the literature. Therefore, in this study, we decided to address both of these needs by first generating a Turkish corpus for text classification and then by testing some algorithms using this Turkish corpus. A comparison of these algorithms under different configurations have also been presented to contribute other works on this subject. DMOZ data, which is the most extensive human-made data source consisting of pre-classified web pages, is decided to use. A web-crawler that brings only Turkish pages along with their classes is designed. After cleaning redundant information for classification task on these web pages automatically, a Turkish corpus is obtained. During or after the cleaning phase, there has been no manual intervention such as removing or adding some words. The obtained dataset is in such a high quality that it can be used as a test bed for other studies, as well.

In this study, the performance of Naive Bayes and Support Vector Machines algorithms, which are among the most frequently used algorithms for text classification have been compared. Selection of n-gram word vector and information gain ratio approach have also been considered. Moreover, it has been focused on the number of categories, the content of data used to train the model and the completeness of this data, and also the effects of these on classification success are examined.

The results show that the performance of both algorithms increase significantly when instances with small number of categories are used. Also, the quality of the content such as including title, keywords and description completely provided to be another factor that affects the classification performance more than the DMOZ data. When the algorithms are trained with instances that are composed of web pages with no missing information such as the web site title and meta data their performance is again, seem to be better. The results show that Multinominal Naive Bayes algorithm is more robust when compared to Support Vector Machines method. In addition, it has been shown that the performance of Multinomial Naive Bayes can further be improved by using 2- gram word vectors. The inclusion of Information Gain Ratio did not seem to improve the performance of Naive Bayes, however it did affect the performance of Support Vector Machines in the positive way.

1. GİRİŞ

Günümüzde artan veri miktarı ile başa çıkabilmek için tek başına insan gücünün kullanılması mümkün görünmemektedir. Bu sebeple benzer dokümanları bir araya gruplayabilecek ya da dokümanları, önceden tanımlı kategorilere belli kurallara göre yerleştirecek bazı otomatikleştirilmiş yöntemlere ihtiyaç duyulmaktadır. Çalışma kapsamında yüksek miktarda veri ile ilgilenen veri madenciliği içerisinde oldukça önemli bir yer tutan metin sınıflandırma konusuna değinilmiştir.

Gürcan (2009) M-NB, DVM, KNN ve karar ağacı algoritmaları ile yaptığı çalışmada dengeli dağılım gösteren beş kategori üzerinde Türkçe metin sınıflandırma için en yüksek başarıya M-NB algoritması ile ulaşmıştır.

Yılmaz (2013), Türkçe dokümanlar üzerinde yaptığı sınıflandırma işleminde dengeli dağılım gösteren altı kategori üzerinde KNN, çok katmanlı algılayıcı ve DVM ile yaptığı çalışmalarda sözcük, hece ve karakterler için n-gram analizleri yapmış ve en yüksek başarıya DVM ile ulaşmıştır.

Kolyiğit (2013) dengeli dağılım gösteren altı kategori üzerinde hece, kelime, gövde ve karakter n-gram kelime vektörü kullanarak KNN, DVM, yapay sinir ağı ile yaptığı çalışmada en yüksek başarıya DVM ile ulaşmıştır.

Pilavcılar (2007) dengesiz dağılım gösteren dört kategori üzerinde KNN ve M-NB ile çalışmış, en yüksek başarıya M-NB algoritması ile ulaşmıştır.

Tarımcı (2009) DMOZ verileriyle çalışarak dengesiz dağılım gösteren beş kategori üzerinde M-NB ve çok katmanlı algılayıcı kullanarak yaptığı çalışmada en yüksek başarıya M-NB ile ulaşmıştır.

Değerli (2012) içeriklerine göre web günlüğü sınıflandırma çalışmasında dengesiz dağılım gösteren sekiz kategori üzerinde M-NB algoritması ile çalışmıştır.

Amasyalı (2009) dengeli dağılım gösteren beş kategoriden oluşan veriseti üzerinde doğrusal regrasyon, adımlı regrasyon, KNN, DVM ve rastgele orman yöntemleriyle yaptığı çalışmada en yüksek başarıya doğrusal regrasyon ile ulaşmıştır.

Kaliyeva (2013), Amasyalı (2009) çalışmasında kullanılan veriseti üzerinde karakter 2-gram ve 3-gram özellik vektörleri kullanarak KNN, NB, M-NB ve DVM ile yaptığı çalışmalarda en yüksek başarıya M-NB ile ulaşmıştır.

Kaşıkçı ve arkadaşları (2014), kullanıcı tarafından belirlenen internet sitelerinin içeriğini analiz ederek e-ticaret sitesi olup olmadığına karar veren çalışmada KNN ve NB algoritmalarını kullanmıştır. En yüksek başarıya NB algoritması ile ulaşmıştır.

1.1 Tezin Önemi ve Amacı

Kullanıcıların daha ilgili oldukları web sayfalarına ulaşımlarının kolaylaştırılabilmesi amacıyla web sayfaları konularına göre sınıflandırılabilir ve ilgili kategoriler yardımıyla kullanıcıların ilgilenebileceği web sayfaları sunulabilir. Bu çalışma gerektiğinde bir sosyal imleme sitesine, konu odaklı bir web tarayıcısı tasarımına veya geliştirilmekte olan bir arama motoruna katkı sağlayabilir.

Tez kapsamında DMOZ web sayfası üzerindeki Türkçe kategorisinde bulunan web sayfalarından faydalanılarak geniş bir veriseti oluşturulmuştur. Oluşturulan verisetinde 14 kategori ve 22 bin kayıt bulunmaktadır. Bu veriseti, Türkçe metin sınıflandırma uygulamaları için kaynak niteliğindedir (Şekerci Hüsem, 2017).

Oluşturulan veriseti ile eğitilen bir sistem tasarlanarak metin sınıflandırmaya uygun veri madenciliği ve makine öğrenmesi algoritmaları kullanılmış ve böylece eğitilen sisteme verilen herhangi bir web sayfasının DMOZ kategorilerinden hangisine ait olabileceği tespit edilmiştir. Bu sayede insan yapımı güvenilir bir verisetinden yola çıkılarak web sayfalarının içeriklerine göre sınıflandırılmasındaki tespitler daha kararlı olacaktır.

1.2 Tezin Yapısı

Tezin ikinci bölümünde, Metin sınıflandırma işleminin uygulanabilmesi için metinler üzerinde yapılması gereken ön işlemler anlatılmış ve örnekler verilmiştir. Ön işlemden geçirilmiş metinler üzerinde sınıflandırma yapılmadan önce uygulanabilecek kelime vektörü (n-gram) ve bilgi kazanım oranından (BKO) bahsedilmiştir. Metin sınıflandırma işlemlerinde kullanılan veri madenciliği

teknikleri hakkında bilgi verilmiştir. Günümüzde metin sınıflandırmanın kullanıldığı alanlardan bahsedilmiştir.

Üçüncü bölümde, çalışmanın yapılacağı verisetinin oluşturulma aşamalarından ve verisetinin saklandığı ortamdan bahsedilmiştir. DMOZ verisetinin kullanımı, sayfa dolaşımları ve sınıflandırma için metinler üzerinde gerçekleştirilen ön işlemler anlatılmıştır. Altı adet deney verisetinin hangi kriterler ve benzerlikler kullanılarak oluşturulduklarına değinilmiştir. Bu deney verisetlerinde bulunan özelliklerden ve kategorilere göre çalışılan sayfa sayılarından ayrıca bahsedilmiştir.

Dördüncü bölümde, seçilen veri madenciliği algoritmalarının uygulaması ve sonuçları bulunmaktadır. Oluşturulmuş altı deney veriseti üzerinde Multinomial Naive Bayes (M-NB) ile Destek Vektör Makinesi (DVM) algoritmaları uygulanmıştır. Kategori sayısı, modeli eğitmek için kullanılan veriseti içeriği ve bu verisetinin tamlığı konularına değinilerek verisetleri üzerinde bilgi kazanım oranı (BKO) ve n-gram kelime vektörü uygulandığında sonuçların nasıl etkilendiği incelenmiş ve test sonuçları listelenmiştir.

2. METİN SINIFLANDIRMA

Veri madenciliği, büyük veri yığınlarından anlamlı ve işe yarar bilgiler üretme işlemidir. Pazarlama, bankacılık, sigortacılık gibi birçok alanda uygulanmaktadır. Pazar araştırması, kredi risk analizi, market sepeti analizi, mevcut müşterilerin elde tutulması, satış tahmini, kredi kartı dolandırıcılıklarının tespiti ve benzer belgelerin tespiti gibi birçok somut örnek ortaya çıkmaktadır (Baykal, 2006).

Metin sınıflandırma ise veri madenciliği ve doğal dil işleme yöntemleri kullanılarak, doğal dil ile yazılmış çok sayıda dokümanın önceden tanımlanmış sınıf listesinden hangisine dahil olabileceğini bulmaktır (Sebastiani, 2002). Örneğin, birçok alanda yazılmış makalelerin bulunduğu veritabanında, makalelerin sınıflandırılmasının elle yapılması maliyetli olabilir. Bu makalelerin sınıflandırma teknikleri kullanılarak sistem tarafından yapılması zaman ve maliyet açısından olumlu katkı sağlamaktadır.

Bunun yanında haber içeriğine göre filtreleme, duygu analizi, e-posta sınıflandırma ve zararlı e-posta (spam) tespiti güncel hayatta kullanılan metin sınıflandırma uygulamalarından bazılarıdır (Aggarwal & Zhai, 2012).

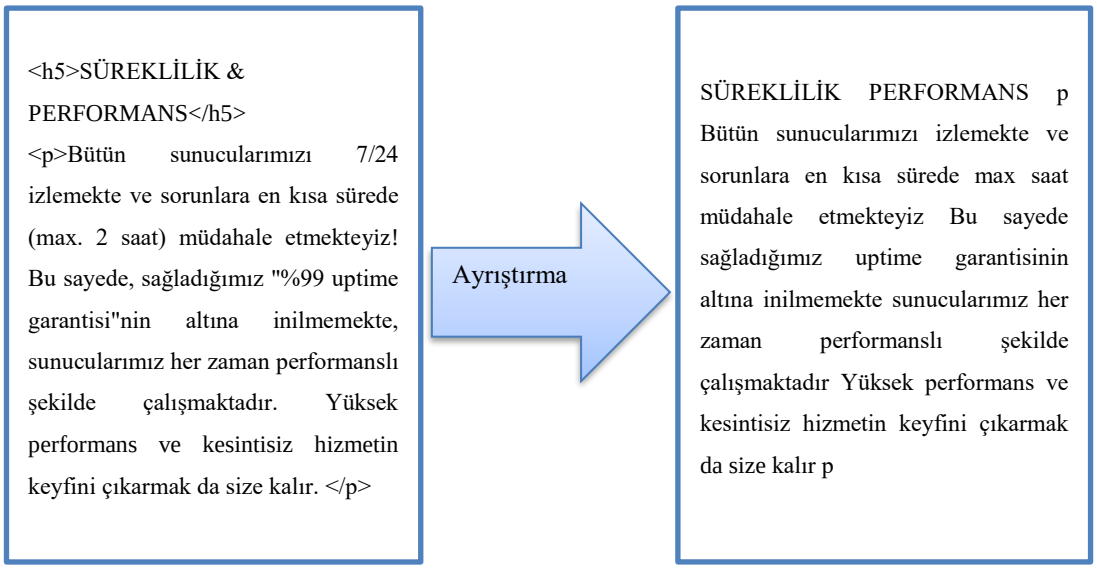
2.1 Metin Sınıflandırma İçin Gerekli Ön İşlemler

Metin sınıflandırma algoritmalarının başarılı bir şekilde uygulanabilmesi için veri üzerinde bazı hazırlık işlemleri yapılır.

2.1.1 Ayrıştırma (Tokenization)

Doküman üzerinde yapılacak çalışmada, herhangi bir anlam içermeyecek karakterleri ayrıştırmak için uygulanan işlemdir (Tunalı & Bilgin, 2012). Burada asıl amaç yalnızca kelimelere ulaşabilmektir. Dolayısıyla tüm noktalama işaretleri, boşluklar, sayılar ve çalışılan alfabe dışında karşılaşılmayacak karakterler temizlenir.

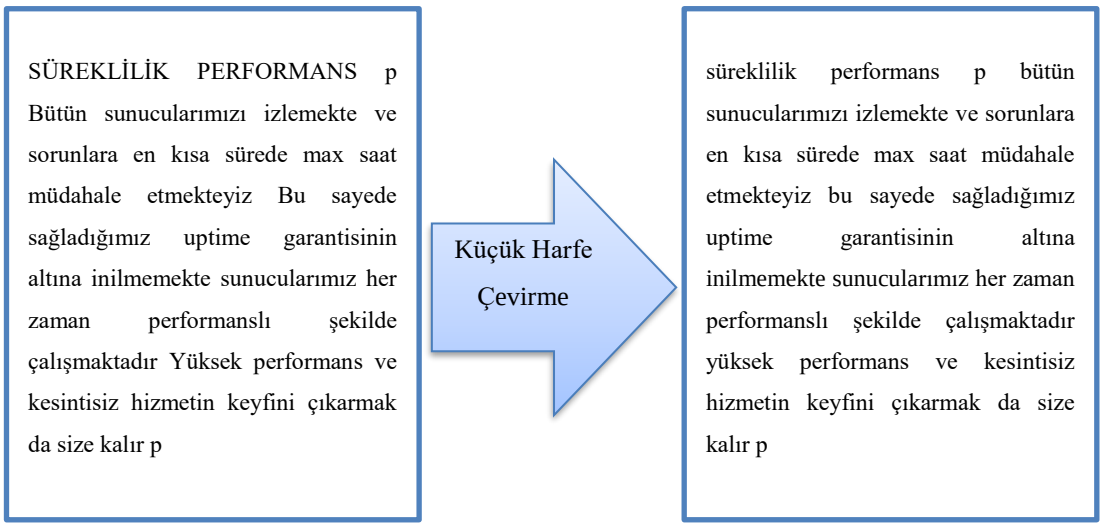
Bir web sayfasından alınmış içeriğin ayrıştırma işlemi öncesi ve sonrasında verinin durumu Şekil 2.1'de gösterilmiştir.



Şekil 2.1 : Ayrıştırma işleminden önceki ve sonraki durum.

2.1.2 Tüm karakterlerin küçük harfe çevrilmesi

Çalışılan platformlarda büyük ve küçük harfler nedeniyle farklı algılanabilecek ifadelerin önüne geçebilmek amacıyla tüm metin ait olduğu dilin alfabesi dikkate alınarak küçük harfe dönüştürülür. Web sayfasından alınmış ve ayrıştırma işlemine tabi tutulmuş metnin tüm harflerinin küçük harfe çevirme işlemi öncesi ve sonrası Şekil 2.2'de gösterilmiştir.



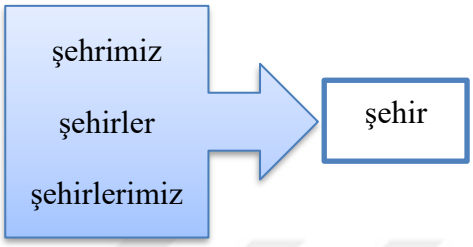
Şekil 2.2 : Küçük harfe çevirme işleminden önceki ve sonraki durum.

2.1.3 Köke indirgeme (Stemming)

Kelimelere ayrıştırılmış veriler içerisinde anlamca farklı olsa da aynı kök veya gövdeye sahip kelimeler üzerinde kök veya gövdeye indirgeme işlemi yapılır

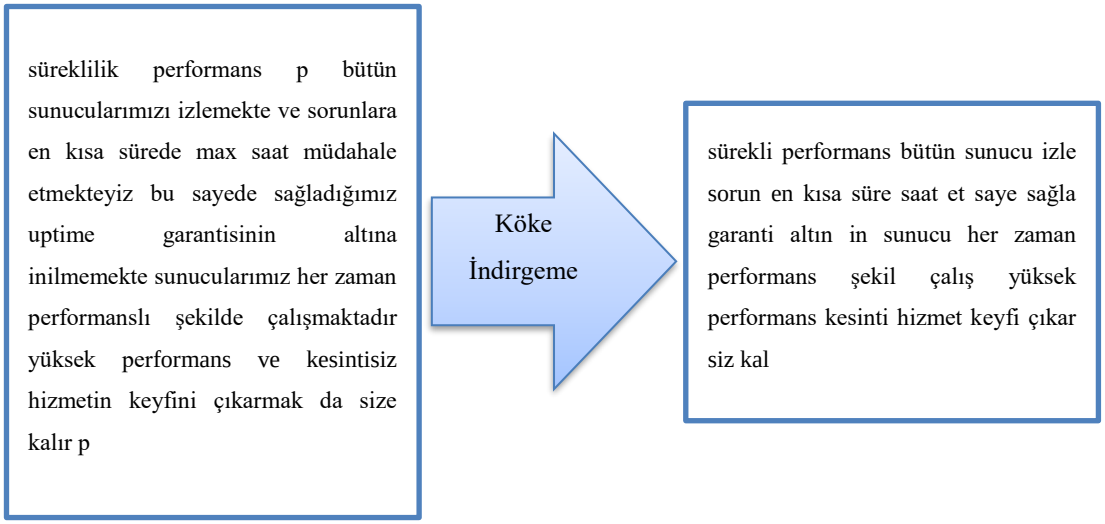
(Sandhya ve diğ, 2016). Buradaki amaç, aldığı ekler sayesinde anlamca farklı olsa da aslında aynı şeyi ifade edebilecek kelimeleri aynı kelime olacak şekilde değerlendirerek başarıyı artırabilmektir. Böylece kelime vektörü de benzer anlamlı kelimelerden arındırılır.

Örneğin Şekil 2.3'te gösterildiği gibi "şehir" kelimesi doküman içerisinde "şehrimiz", "şehirler", "şehirlerimiz" gibi farklı şekillerde bulunabilir.



Şekil 2.3 : Köke indirgeme işlemi.

Köke indirgeme işlemi yapılmadığı takdirde bu kelimelerin hepsi farklı bir kelime olarak yorumlanır ve metin sınıflandırma başarısını ciddi ölçüde düşürür. Köke indirgeme işleminin örnek üzerinde öncesi ve sonrası Şekil 2.4'te gösterilmiştir. Bu örnekte de olduğu gibi Zemberek yazılımının Türkçe sözlüğünde bulunmayan kelimeler çıkarılmıştır.



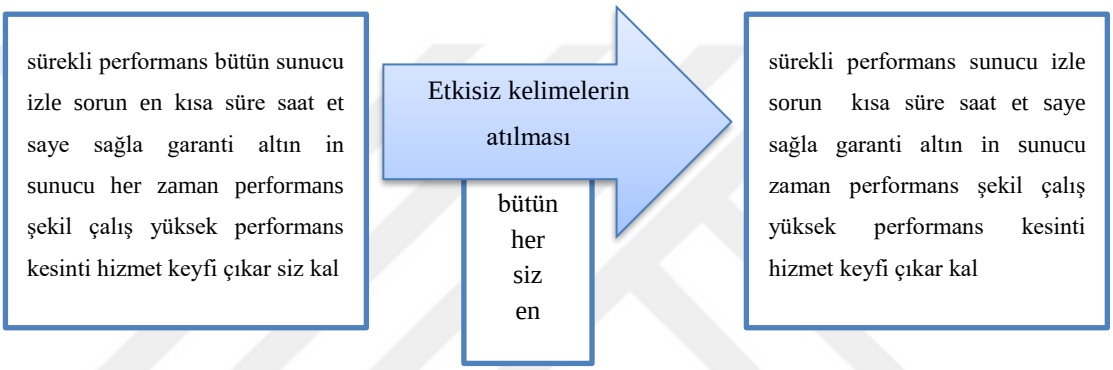
Şekil 2.4 : Köke indirgeme işleminden önceki ve sonraki durum.

Köke indirgeme işleminde indirgenen kelimenin anlamına bakmadan yapım veya çekim ekleri atıldığından dolayı her durumda doğru sonuç vermemektedir. Örneğin "altına" kelimesi, verilen örnekte kökünün "alt" olması gerekirken köke indirgeme

işleminden sonra “altın” kelimesi olarak çıktı alınmıştır. Fakat bu iki kelimenin anlamı birbiriyle örtüşmemektedir.

2.1.4 Etkisiz kelimeler (Stop-words)

Sınıflandırma başarısını yükseltmek için, her dokümanda bulunabilecek ancak sınıflandırma için bir anlam ifade etmeyecek etkisiz kelimeler atılmalıdır. Bu kelimeler çalışılan dile göre değişmektedir. Örneğin Türkçe’de “böyle”, “halbuki”, “ve”, “veya” gibi kelimeler veri kümesinden atılır (Tunalı & Bilgin, 2012). Etkisiz kelimelerin çıkarılma işleminin örnek üzerinde öncesi ve sonrası Şekil 2.5’te gösterilmiştir.



Şekil 2.5 : Etkisiz kelimelerin çıkarılma işleminden önceki ve sonraki durum.

Bu kelimeler, dokümanların tamamında geçebilecek kelimeler olduklarından sınıflandırmaya katkıları olmadığı gibi kelime vektörü boyutunu da artıracaklarından dolayı, sınıflandırma işleminden önce dokümanlardan bu kelimeler çıkartılır.

2.2 Kelime Vektörü Oluşturma

Veri madenciliği yöntemlerinde sözlükteki kelimelerin indisler ile temsil edilmesi yaygın bir şekilde kullanılmaktadır (Maas ve diğ, 2011).

n-Gram Kelime Vektörü

n-gram, kelime vektörü oluşturmada kullanılan en genel yaklaşımlardan biridir. n-gram kelime modelinde temel yaklaşım, n. kelimenin kendinden önceki n-1 kelimeyle birlikte değerlendirilmeye alınmasıdır (Liu, 2011). Dolayısıyla n=1 ifadesinde her kelime kendi başına değerlendirilirken n=2 ifadesinde kelime grupları

ikişerli olacak şekilde değerlendirmeye alınır. Benzer şekilde n=3 için kelimeler üçerli gruplar halinde değerlendirilir.

Örneğin "Bilgisayarın atası sayılan hesap aygıtı abaküstür." cümlesi üzerinde ön işlemler gerçekleştirdikten sonra "bilgisayar ata say hesap aygıt abaküs" ifadesi ortaya çıkar. Bu ifade, kelime n-gram kelime vektörü kullanılarak Çizelge 2.1'de örneklendirilmiştir.

Çizelge 2.1 : Kelime n-gram yönteminin n=1, 2 ve 3 için örneklendirmesi.

kelime 1-gram	bilgisayar, ata, say, hesap, aygıt, abaküs
kelime 2-gram	bilgisayar ata, ata say, say hesap, hesap aygıt, aygıt abaküs
kelime 3-gram	bilgisayar ata say, ata say hesap, say hesap aygıt, hesap aygıt abaküs

2.3 Öznitelik Seçimi ve Değerlendirmesi

Bilgi kazanım oranı (Information Gain Ratio), bilgi kazanımı (Informatin Gain) tekniğinin çok sayıda özellik içeren özellik setlerindeki olası hatalı eğilimlerin üstesinden gelebilmek amacıyla geliştirilmiştir (Baig ve diğ, 2011).

Bilgi kazanımı, sınıf bilgisinden özniteliğin bilgisi çıkartılarak denklem (2.1)'de görüldüğü gibi hesaplanır. X, örnekleri temsil ederken $\{A_1, \dots, A_p\}$ ise özellikleri temsil eder. *I(X)* ile *m* adet mevcut sınıf içerisinde *X* örneklerinin dağılımı denklem (2.2)'de gösterildiği gibi ölçülür. p_j, tüm örnekler içinde f_j sınıfının görülme olasılığını belirtir.

$$Bilgi\ Kazanımı\ (A_k, X) = I(X) - E(A_k, X) \quad \textbf{(2.1)}$$

$$I(X) = -\sum_{j=1}^{m} p_j log_{\,2}(p_j), \quad p_j = \frac{|X \cap f_j|}{X} \quad \textbf{(2.2)}$$

Denklem (2.1)'de sözü edilen $E(A_k, X)$ ise denklem (2.3)'te gösterildiği gibi hesaplanır. Burada *n,* A_k özniteliğinin alabileceği değerleri niteler. $|X_i|$, örnekler içerisinde A_k özniteliğinin alabileceği değerleri belirtir. $|X|$ ise mevcut örneklerin sayısını temsil eder.

$$E(A_k, X) = \sum_{i=1}^{n} \frac{|X_i|}{|X|} I(X_i) \quad \textbf{(2.3)}$$

Bilgi kazanım oranı da denklem (2.4)'te gösterildiği gibi bilgi kazanım değerinin denklem (2.5)'teki ayrışma bilgisine oranıdır.

$$Bilgi\ Kazanım\ Oranı\ (A_k, X) = \frac{Bilgi\ Kazanımı\ (A_k, X)}{Ayrışma\ Bilgisi\ (A_k)} \quad \textbf{(2.4)}$$

Ayrışma bilgisi, denklem (2.5)'te gösterildiği gibi hesaplanır. Ayrışma bilgisi ile özelliğin değerinin bilgisi ölçülür ve böylece sınıflandırmada kullanılacak özniteliğin değeri belirlenmiş olur (Mantaras, 1991).

$$Ayrışma\ Bilgisi(A_k) = -\sum_{i=1}^{n} \frac{|X_i|}{|X|} log_2 \frac{|X_i|}{|X|} \quad \textbf{(2.5)}$$

2.4 Metin Sınıflandırmada Öğrenme Yöntemleri

Veri madenciliğinde temelde iki adet öğrenme yönteminden söz edilir. Ancak bu öğrenme yöntemlerinden hangisinin kullanılabileceği verinin yapısına göre değişmektedir.

Denetimli (Supervised) Öğrenme

Bu öğrenme yöntemleri, insan öğrenmesine benzer şekilde yorumlanabilir. İnsanın geçmiş deneyimlerinden yola çıkılarak yeni durumlara uyum sağlamasına benzer şekilde, makine öğrenmesinde de verilerden yola çıkılarak anlamlı bir model oluşturma, bu model yardımıyla da yeni durumlara tahmin üretme süreci olarak söylenebilir (Liu, 2011, s. 63).

Denetimli öğrenme yönteminde tüm veriler bir sınıfa dahildir. Bu sınıflar birbirinin içine geçmeyen, ayrık verilerden oluşur. Bu tür öğrenmeden söz ediliyorken en az iki sınıfın varlığından söz edilir. Her bir veri, insan öğrenmesine kıyasla "geçmiş bir deneyim" olarak yorumlanır. Ancak makine öğrenmesi literatürüne göre örnek, durum veya vektör olarak isimlendirilir (Liu, 2011, s. 63). Örneğin zararlı e-postaların tespit edilmesi işleminde, tüm e-postaların barındırdığı kelimeler, her bir

e-postanın vektörünü oluşturur. E-postaların zararlı olup olmadığına karar verilirken hesaplanan vektörlerin değerleri göz önüne alınır.

Denetimsiz (Unsupervised) Öğrenme

Bu öğrenme yöntemlerinde denetimli öğrenmede olduğu gibi bir sınıf bilgisi bulunmaz. Bu nedenle sınıflandırma yaklaşımından bahsedilemez. Bunun yerine, bu tür verilerle çalışmak için kümeleme denilen yaklaşımdan yararlanılır (Liu, 2011, s. 133).

Kümeleme işleminde veriler, benzerlik veya yakınlıklarına göre kümelere ayrılırlar. Benzer özellik gösteren veriler aynı kümede toplanırken veriler arasındaki fark fazlaysa farklı ve hatta uzak kümelerde bulunabilirler (Liu, 2011, s. 133).

Örneğin dokümanlar kategorilere ayrılmak istendiğinde ilk aşamada birbirlerine en yakın anahtar kelime içeren doküman kümeleri oluşturulup bu kümelere kategori ismi atanabilir. Ancak bu kümeler ilk durumda etiketsizdir.

2.5 Metin Sınıflandırmada Kullanılan Algoritmalar

Veri madenciliğinde En Yakın k Komşu (k-Nearest Neighbor - KNN), Naive Bayes, Destek Vektör Makineleri (Support Vector Machines), Yapay Sinir Ağları (Artificial Neural Networks) ve Karar Ağacı (Decision Tree) gibi birçok algoritma mevcuttur. Tez kapsamında aşağıdaki yöntemlerle çalışılmıştır.

2.5.1 Naive bayes

Naive Bayes (NB), metin tabanlı dokümanları da sınıflayabilen olasılıksal bir yöntemdir (Liu, 2011, s. 104). Denetimli sınıflandırma algoritmaları içerisindedir. Bayes teoremine göre sınıf etiketleri kullanılarak mevcut dokümanlardan üretilen model çerçevesinde yeni gelen dokümanın hangi sınıfa ait olabileceği tahmin edilir (Yussouf Nahayo, 2016).

Bernoulli Doküman Modeli

Doküman setinden elde edilen kelimeler vektöre dönüştürülerek mevcut doküman içerisinde ilgili kelimenin olup olmadığı ile ilgilenir. Bu modelde kelimenin varlığı ve yokluğu üzerinden işlem gerçekleştirilir (McCallum & Nigam, 1998).

Kelime vektörü, mevcut sınıflandırma işlemi için oluşturulan sözlükte geçen $w_1, w_2, \ldots w_n$ kelimelerinden oluşmaktadır. Her bir doküman için sözlük yardımıyla denklem (2.6)'da gösterilen kelime vektörü kullanılır.

$$\overrightarrow{W} = w_1\ w_2 \ldots\ w_n \quad \textbf{(2.6)}$$

Denklem (2.7)'de *n* adet *d* dokümanı içindeki kelimeler, kelime vektörü aracılığıyla içerdikleri kelimeler 1, içermedikleri 0 ile işaretlenir.

$$\forall d_n\ için \begin{cases} w_i = 1, & eğer\ d_n\ içinde\ w_i\ varsa \\ w_i = 0, & eğer\ içinde\ yoksa \end{cases} \quad \textbf{(2.7)}$$

Bahsedilen işlemlere göre üç doküman ve altı kelimeden oluşan örnek Çizelge 2.2'de gösterilmiştir.

Çizelge 2.2 : Bernoulli doküman modeline göre kelime vektörü.

Doküman/Kelime	w_1	w_2	w_3	w_4	w_5	w_6
d_1	1	1	1	0	1	0
d_2	1	0	0	0	0	1
d_3	0	1	0	1	0	0

Multinomial Doküman Modeli

Çok sınıflı çalışmalarda Multinomial Naive Bayes (M-NB) yöntemi kullanılmaktadır. Bernoulli doküman modelindekine benzer şekilde, kelimeler vektöre dönüştürüldükten sonra kelimenin dokümanda varlığı ve yokluğundan ziyade, kelimenin dokümanda kaç kez geçtiği ile hesaplama yapılmaktadır. Bu yöntem ile dokümanın, ait olması gereken sınıfa olan bağlılığı güçlendirilmeye çalışılmıştır (Bermejo ve diğ, 2010).

Sınıfların $C = \{c_1, c_2, \ldots c_n\}$ şeklinde bir kümede olduğu varsayılır. Aynı şekilde, doküman setindeki kelime kümesinde *m* adet kelime $W = \{w_1, w_2, \ldots w_m\}$ şeklinde tanımlı olsun. Bayes teoremine göre verilen *d* dokümanının c_i sınıfına aitliği denklem (2.8) ile hesaplanır.

$$p(c_i|d) = \frac{p(d|c_i)\, p(c_i)}{p(d)} \quad \textbf{(2.8)}$$

d dokümanı için en muhtemel sınıfı bulan genelleştirilmiş denklem (2.9)'da gösterilmiştir.

$$c^*(d) = argmax_{c_i} \;\; \mathrm{p}(c_i|d) = \mathrm{p}(c_i) \prod_{t=1}^{m} p(w_t|c_i)^{n(w_t,d)} \quad \textbf{(2.9)}$$

Denklem (2.9)'da, $n(w_t, d)$ ifadesi w_t kelimesinin d dokümanı içerisinde kaç kez geçtiğini belirtir. Böylece ilgili kelimenin sınıf içinde bulunma olasılığı, frekans sayısı kadar kendisiyle çarpılmış olur. Bunun yanında $p(w_t|c_i)$ ifadesi de denklem (2.10)'da gösterildiği şekliyle elde edilir (Dhillon ve diğ, 2002).

$$p(w_t|c_i) = \frac{\sum_{d_j \epsilon c_i} n(w_t, d_j)}{\sum_{t=1}^{m} \sum_{d_j \epsilon c_i} n(w_t, d_j)} \quad \textbf{(2.10)}$$

Eğer doküman içinde, hesaplanan kelimeyi içermiyorsa denklem (2.10)'un sonucu sıfır olacaktır. Buradan hareketle denklem (2.9)'un da sonucu sıfır olarak hesaplanacaktır. Bu durumun ortadan kaldırılabilmesi amacıyla Laplace dönüşümü uygulanmakta ve denklem (2.11) elde edilmektedir.

$$p(w_t|c_i) = \frac{1 + \sum_{d_j \epsilon c_i} n(w_t, d_j)}{m + \sum_{t=1}^{m} \sum_{d_j \epsilon c_i} n(w_t, d_j)} \quad \textbf{(2.11)}$$

Naive Bayes kuralı, bilgi teorisi çerçevesinde yeniden yazılarak denklem (2.12) elde edilmektedir. Burada $p(w_t|d)$ ifadesi, d dokümanı içerisinde w_t kelimesinin geçme olasılığıdır.

$$c^*(d) = argmin_{c_i} \;\; \sum_{t=1}^{m} p(w_t|d) \log \frac{p(w_t|d)}{p(w_t|c_i)} - \log p(c_i) \quad \textbf{(2.12)}$$

M-NB algoritmasının çalışma yapısı çok sınıflı bir metin üzerinde örneklenmiştir. Bu örnekte *'Ankara', 'İstanbul', 'İzmir', 'Van'* olmak üzere dört adet sınıf bulunmaktadır. Eğitim verilerinde bu sınıfları örnekleyen kelimeler bulunurken test verilerinde metnin hangi sınıfa dahil olacağını belirlememize yarayan kelimeler

bulunmaktadır. Eğitim verisinde bulunan sınıf etiketleri, sistemi sınıf özelliklerine göre eğitmemizi sağlarken test verisinde bulunan sınıf etiketleri, doğru sınıflandırma yapılıp yapılmadığının kontrolü için gereklidir. Eğitim ve test verileri Çizelge 2.3'te gösterilmiştir.

Çizelge 2.3 : M-NB algoritmasının uygulandığı örnek eğitim ve test verileri.

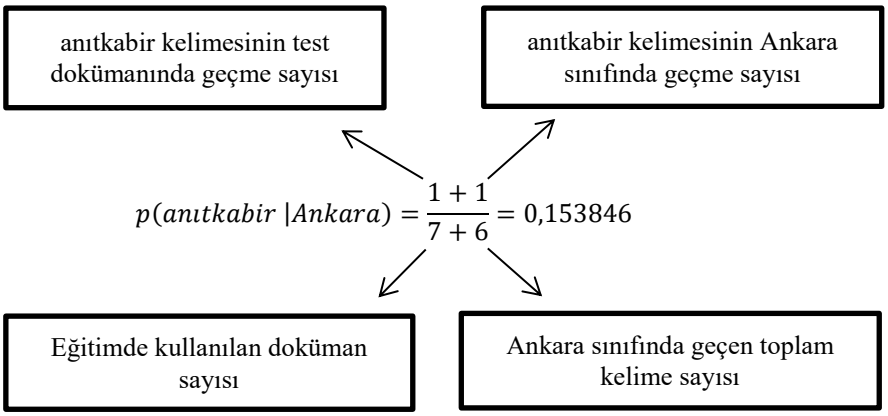
Eğitim Veriseti		Test Veriseti	
Ankara	"anıtkabir başkent kedi"	Ankara	"başkent kedi"
İstanbul	"turist boğaz"	İzmir	"turist"
İzmir	"turist sıcak göl sıcak"	Van	"kedi göl kedi"
Van	"kedi göl"		
Ankara	"başkent kedi göl"		
İstanbul	"boğaz göl turist"		
Van	"göl"		

Çizelge 2.3'te bulunan eğitim verilerine göre kelimelerin sınıflar içerisinde kaç kez geçtiği Çizelge 2.4'te gösterilmiştir.

Çizelge 2.4 : M-NB algoritması örneğinde kelimelerin sınıflarda bulunma sayısı.

Kelimeler \ Sınıflar	Ankara	İstanbul	İzmir	Van
anıtkabir	1	0	0	0
başkent	2	0	0	0
boğaz	0	2	0	0
göl	1	1	1	2
kedi	2	0	0	1
sıcak	0	0	2	0
turist	0	2	1	0

Örneğin "anıtkabir" kelimesinin tüm sınıflarda bulunup bulunmaması ve kaç adet bulunduğuna bağlı olarak kelime ağırlıkları aşağıdaki gibi hesaplanır. Şekil 2.6'da açıklanan hesaplama, denklem (2.8) temel alınarak gerçekleştirilmiştir. Bu işlem eğitim verisetinde bulunan tüm kelimeler için gerçekleştirilir ve model bir kez oluşturulur. Tüm kelimelerin belirtilen sınıflara ait değerleri Çizelge 2.5'te gösterilmiştir.



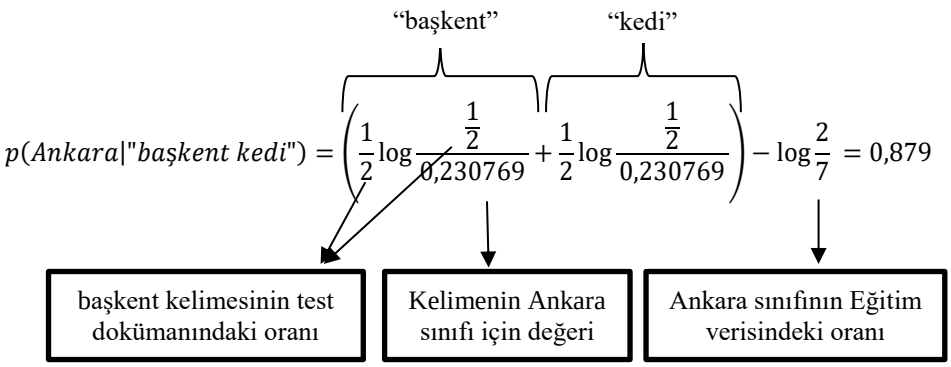
Şekil 2.6 : M-NB yöntemine göre kelimenin sınıf içindeki değerinin hesaplanması.

Çizelge 2.5 : Kelimelerin sınıflara göre değerleri.

Kelimeler \ Sınıflar	Ankara	İstanbul	İzmir	Van
anıtkabir	0,153846	0,083333	0,100000	0,100000
başkent	0,230769	0,083333	0,100000	0,100000
boğaz	0,076923	0,250000	0,100000	0,100000
göl	0,153846	0,166666	0,200000	0,300000
kedi	0,230769	0,083333	0,100000	0,200000
sıcak	0,076923	0,083333	0,200000	0,100000
turist	0,076923	0,250000	0,200000	0,100000

$$p(Ankara|"başkent\ kedi") = \left(\frac{1}{2}\log\frac{\frac{1}{2}}{0{,}230769} + \frac{1}{2}\log\frac{\frac{1}{2}}{0{,}230769}\right) - \log\frac{2}{7} = 0{,}879$$

Örneğin sınıfı *Ankara* olan "*başkent kedi"* içerikli test verisinin ait olduğu muhtemel sınıfın bulunabilmesi için tüm sınıflar üzerinde denklem (2.12) kullanılarak aşağıdaki hesaplamalar yapılmaktadır. Bu hesaplamalar sonucunda, dokümanın aitlik değeri en düşük olan sınıfa dahil olduğu kabul edilir.



Şekil 2.7 : M-NB yöntemine göre test verisinin sınıf tahmini hesaplaması.

$$p(İstanbul|"başkent\ kedi") = \left(\frac{1}{2}\log\frac{\frac{1}{2}}{0{,}083333} + \frac{1}{2}\log\frac{\frac{1}{2}}{0{,}083333}\right) - \log\frac{2}{7} = \quad 1{,}322$$

$$p(İzmir|"başkent\ kedi") = \left(\frac{1}{2}\log\frac{\frac{1}{2}}{0{,}100000} + \frac{1}{2}\log\frac{\frac{1}{2}}{0{,}100000}\right) - \log\frac{1}{7} = \quad 1{,}544$$

$$p(Van|"başkent\ kedi") = \left(\frac{1}{2}\log\frac{\frac{1}{2}}{0{,}100000} + \frac{1}{2}\log\frac{\frac{1}{2}}{0{,}200000}\right) - \log\frac{2}{7} = \quad 1{,}092$$

"*başkent kedi"* içerikli test verisinin *Ankara* sınıfına ait olduğu hesaplamalarda görülmektedir. Test dokümanından kontrol edildiğinde doğru sınıflandırıldığı tespit edilmektedir.

Başka bir örnek olarak sınıfı *İzmir* olan "*turist"* içerikli test verisinin sınıf tahmini için aşağıdaki hesaplamalar yapılmaktadır.

$$p(Ankara|"turist") = \left(\frac{1}{1}\log\frac{\frac{1}{1}}{0.076923}\right) - \log\frac{2}{7} = \quad 1{,}658$$

$$p(İstanbul|"turist") = \left(\frac{1}{1}\log\frac{\frac{1}{1}}{0.250000}\right) - \log\frac{2}{7} = \quad 1{,}146$$

$$p(İzmir|"turist") = \left(\frac{1}{1}\log\frac{\frac{1}{1}}{0.200000}\right) - \log\frac{1}{7} = \quad 1{,}544$$

$$p(Van|"turist") = \left(\frac{1}{1}\log\frac{\frac{1}{1}}{0.100000}\right) - \log\frac{2}{7} = \quad 1{,}544$$

Bu hesaplamalar sonucunda en küçük değerin *İstanbul* sınıfına ait olduğu görülmektedir. Test dokümanında kontrol edildiğinde ise aslında *İzmir* sınıfına ait olduğu görülmektedir. Bunun sebebi eğitim verisinde "turist" kelimesinin *İstanbul* sınıfına ait dokümanlarda daha fazla tekrar etmesi ve *İzmir* sınıfına ait az sayıda doküman olmasının sınıflandırmayı olumsuz yönde etkilemesidir.

2.5.2 Destek vektör makineleri (Support vector machines)

Destek vektör makinesi (DVM), öncelikle iki sınıflı durumları ayırmak için geliştirilmiş bir algoritmadır. Ayırma işleminin gerçekleştirilmesi için karar fonksiyonu üretilerek iki kategoriyi en iyi şekilde ayıracak bir hiper-düzlem (hyperplane) hesaplanır (Dhillon ve diğ, 2003).

Ancak gerçek hayat problemlerinin doğrusal ve iki sınıf düzeyine indirgenemeyecek olması, DVM'lerin doğrusal olmayan ve çok sınıflı verileri sınıflayabilecek şekilde geliştirilmesine neden olmuştur (Kavzoğlu & Çölkesen, 2010).

Doğrusal olarak ayrılabilir veri kümelerinde belirlenecek hiper-düzlemin iki sınıf arasındaki maksimum uzaklığa denk gelecek şekilde oluşturulması önemlidir.

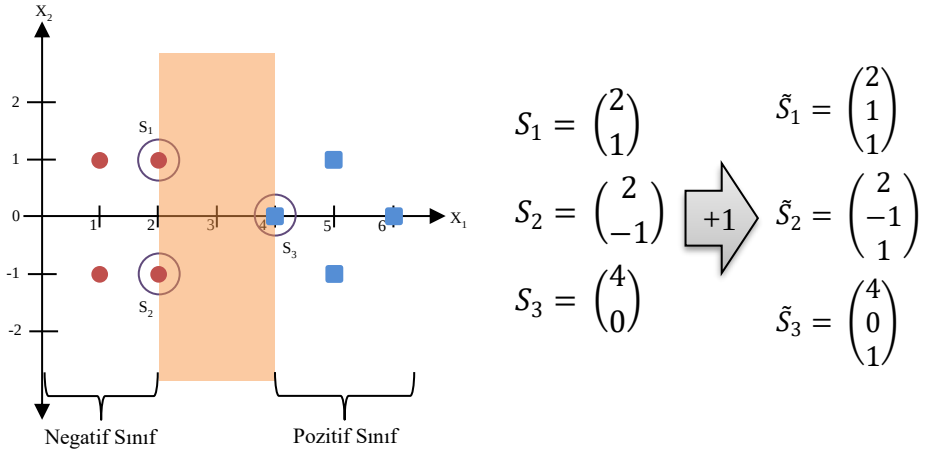
Denklem (2.13)'te gösterilen $\vec{x}$ verileri, $\vec{w}$ ağırlık vektörünü ve b eğitim verilerinden öğrenilen sabit değeri ifade eder.

$$\vec{w}.\vec{x} + b = 0 \qquad \textbf{(2.13)}$$

Sınıf etiketlerini temsil eden $y \in \{-1, +1\}$ olacak şekilde, -1 değeri bir sınıfı, +1 değeri diğer sınıfı niteler. Eğitim verisinin {x, y} ikilileri şeklinde olduğu düşünülürse, en uygun hiper-düzlem eşitsizlikleri denklem (2.14)'te gösterildiği gibidir (Kavzoğlu & Çölkesen, 2010).

$$\begin{aligned} w.x_i + b &\geq +1, \quad \forall\, y = +1\ için \\ w.x_i + b &\leq +1, \quad \forall\, y = -1\ için \end{aligned} \qquad \textbf{(2.14)}$$

Örneğin, Şekil 2.8'de sekiz adet verinin bulunduğu 2-boyutlu bir düzlemde destek vektörleri S_1, S_2, S_3 olarak gösterilmiştir. Bu destek vektörleri, giriş değeri olacak şekilde 1 eklenerek $\tilde{S}_1$, $\tilde{S}_2$, $\tilde{S}_3$ şeklinde gösterilmiştir.



$$S_1 = \begin{pmatrix}2\\1\end{pmatrix} \quad S_2 = \begin{pmatrix}2\\-1\end{pmatrix} \quad S_3 = \begin{pmatrix}4\\0\end{pmatrix} \quad \xrightarrow{+1} \quad \tilde{S}_1 = \begin{pmatrix}2\\1\\1\end{pmatrix} \quad \tilde{S}_2 = \begin{pmatrix}2\\-1\\1\end{pmatrix} \quad \tilde{S}_3 = \begin{pmatrix}4\\0\\1\end{pmatrix}$$

Şekil 2.8 : Düzlem üzerinde sekiz farklı verinin gösterimi ve destek vektörleri.

Örnekte üç adet destek vektör olduğu için üç adet α parametresine ihtiyaç duyulmaktadır. Dolayısıyla her bir destek vektöründe kullanılacak α_1, α_2 ve α_3 değerleriyle denklem (2.15) ile hesaplanmaktadır (Silva, 2014).

$$\begin{aligned}
&\propto_1 \tilde{S}_1\tilde{S}_1 + \propto_2 \tilde{S}_2\tilde{S}_1 + \propto_3 \tilde{S}_3\tilde{S}_1 = -1 \quad (Negatif\ sınıf)\\
&\propto_1 \tilde{S}_1\tilde{S}_2 + \propto_2 \tilde{S}_2\tilde{S}_2 + \propto_3 \tilde{S}_3\tilde{S}_2 = -1 \quad (Negatif\ sınıf)\\
&\propto_1 \tilde{S}_1\tilde{S}_3 + \propto_2 \tilde{S}_2\tilde{S}_3 + \propto_3 \tilde{S}_3\tilde{S}_3 = +1 \quad (Pozitif\ sınıf)
\end{aligned} \qquad \textbf{(2.15)}$$

Her bir destek vektörü için üretilen $\tilde{S}_1$, $\tilde{S}_2$, $\tilde{S}_3$ ile denklem (2.15) kullanılarak hesaplama yapılır.

$$\propto_1 \begin{pmatrix}2\\1\\1\end{pmatrix}\begin{pmatrix}2\\1\\1\end{pmatrix} + \propto_2 \begin{pmatrix}2\\-1\\1\end{pmatrix}\begin{pmatrix}2\\1\\1\end{pmatrix} + \propto_3 \begin{pmatrix}4\\0\\1\end{pmatrix}\begin{pmatrix}2\\1\\1\end{pmatrix} = -1$$

$$\propto_1 \begin{pmatrix}2\\1\\1\end{pmatrix}\begin{pmatrix}2\\-1\\1\end{pmatrix} + \propto_2 \begin{pmatrix}2\\-1\\1\end{pmatrix}\begin{pmatrix}2\\-1\\1\end{pmatrix} + \propto_3 \begin{pmatrix}4\\0\\1\end{pmatrix}\begin{pmatrix}2\\-1\\1\end{pmatrix} = -1$$

$$\propto_1 \begin{pmatrix}2\\1\\1\end{pmatrix}\begin{pmatrix}4\\0\\1\end{pmatrix} + \propto_2 \begin{pmatrix}2\\-1\\1\end{pmatrix}\begin{pmatrix}4\\0\\1\end{pmatrix} + \propto_3 \begin{pmatrix}4\\0\\1\end{pmatrix}\begin{pmatrix}4\\0\\1\end{pmatrix} = +1$$

Yukarıdaki denklemler sadeleştirildiğinde;

$$6 \propto_1 + 4 \propto_2 + 9 \propto_3 = -1$$

$$4 \propto_1 + 6 \propto_2 + 9 \propto_3 = -1$$

$$9 \propto_1 + 9 \propto_2 + 17 \propto_3 = +1$$

Bu üç denklemin ortak çözümü ile şu değerler elde edilir:

$$\propto_1 = \propto_2 = -3{,}25$$

$$\propto_3 = 3{,}5$$

Örnekteki pozitif ve negatif sınıfları ayıracak hiper-düzlem denklem (2.16)'da gösterilmiştir.

$$\tilde{w} = \sum_i \alpha_i \tilde{S}_i \qquad \textbf{(2.16)}$$

Denklem (2.16) kullanılarak aşağıdaki hesaplama gerçekleştirilir.

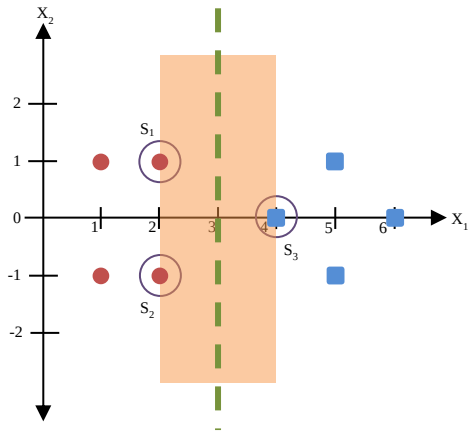
$$\tilde{w} = \propto_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \propto_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \propto_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$

$$\tilde{w} = (-3{,}25)\begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + (-3{,}25)\begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + (3{,}5)\begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

Bu çözüme göre *w* vektörü ile *b* katsayısı aşağıdaki gibi bulunur:

$$w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad b = -3$$

Bulunan *w* değeriyle birlikte denklem (2.13) kullanılarak çözülür ve hiper-düzlem'in $x_1=3$ noktasından geçtiği, x_2'yi kesmediği tespit edilir.



Şekil 2.9 : Hiper-düzlem ile pozitif ve negatif sınıfların ayrılması.

2.6 Metin Sınıflandırmada Performans Ölçütleri

Sınıflandırma işleminde yöntemin etkinliğini ölçmek ve diğer yöntemlerle kıyaslayabilmek adına doğru ve yanlış sınıflandırılan örnekler üzerinde bazı hesaplamalar yapılır.

Karmaşıklık matrisi (Confusion matrix)

Karmaşıklık matrisi, sınıflayıcı tarafından karar verilen gerçek ve hesaplanan örnek sayılarının bulunduğu matristir (Liu, 2011).

Pozitif örnek, karmaşıklık matrisi hesaplanan sınıftır. Negatif örnek ise karmaşıklık matrisi hesaplanan sınıfın dışındaki tüm sınıflardır.

İki-sınıflı sınıflandırmada karmaşıklık matrisi Çizelge 2.6'da gösterildiği gibi olmaktadır.

Çizelge 2.6 : Karmaşıklık matrisi.

		Hesaplanan	
		Pozitif	Negatif
Gerçekte olan	Pozitif	DP	YN
	Negatif	YP	DN

- DP (Doğru Pozitif - True Positive): Doğru sınıflandırılan pozitif örnek sayısıdır.
- DN (Doğru Negatif - True Negative): Doğru sınıflandırılan negatif örnek sayısıdır.
- YP (Yanlış Pozitif - False Positive): Pozitif olarak sınıflandırılan ancak gerçekte negatif olması gereken örnek sayısıdır.
- YN (Yanlış Negatif - False Negative): Negatif olarak sınıflandırılan ancak gerçekte pozitif olması gereken örnek sayısıdır.

Çok-sınıflı sınıflandırmada ise, örneğin sınıf-2 için Çizelge 2.7'de gösterilen karmaşıklık matrisi elde edilir (Rigutini & Maggini, 2004).

Çizelge 2.7 : Sınıf-2 için genelleştirilmiş çok-sınıflı karmaşıklık matrisi.

		Hesaplanan				
		Sınıf-1	Sınıf-2	Sınıf-3	Sınıf-4	Sınıf-5
Gerçekte olan	Sınıf-1	DN	YP	DN	DN	DN
	Sınıf-2	YN	**DP**	YN	YN	YN
	Sınıf-3	DN	YP	DN	DN	DN
	Sınıf-4	DN	YP	DN	DN	DN
	Sınıf-5	DN	YP	DN	DN	DN

Çok-sınıflı sınıflandırma işlemlerinde mikro-ortalama (micro-averaging) ve makro-ortalama (macro-averaging) ölçümleri kullanılmaktadır. Mikro-ortalamada yapılan ölçüm tüm sınıflar üzerinde aynı anda yapılırken makro-ortalamada ise her sınıf ayrı değerlendirilerek ilgili ölçümün ortalaması alınır (Sokolova & Lapalme, 2009). Mikro-ortalamalar μ, makro-ortalamalar M ile gösterilmektedir.

Sınıf sayısının dengesiz dağılım gösterdiği çalışmalarda makro-ortalama hesaplamalarına benzer şekilde sınıflardaki örnek sayılarıyla birlikte değerler ağırlıklandırılarak hesaplanır. Bu hesaplamaya ağırlıklı ortalama (Weighted Average - WA) denilmektedir (Zhou & Yao, 2010).

Hassasiyet (Precision)

İki-sınıflı sınıflandırmalarda hassasiyet denklem (2.17)'de gösterilmiştir.

$$Precision = \frac{DP}{DP + YP} \tag{2.17}$$

Çok-sınıflı sınıflandırmalarda ağırlıklı ortalama, mikro-ortalama ve makro-ortalama ölçümlerine göre hassasiyet hesaplaması *n* adet sınıfta her *i* sınıfı için o sınıftaki örnek sayısı k_i ve tüm örneklerin sayısı *k* olacak şekilde denklem (2.18)'de gösterilmiştir.

$$Precision_{wa} = \frac{1}{k}\sum_{i=1}^{n}(k_i\ Precision)$$

$$Precision_{\mu} = \frac{\sum_{i=1}^{n} DP_i}{\sum_{i=1}^{n}(DP_i + YP_i)} \quad \textbf{(2.18)}$$

$$Precision_M = \frac{\sum_{i=1}^{n} \frac{DP_i}{DP_i + YP_i}}{n}$$

Duyarlılık (Recall - Sensitivity)

İki-sınıflı sınıflandırmalarda duyarlılık denklem (2.19)'da gösterilmiştir.

$$Recall = \frac{DP}{DP + YN} \quad \textbf{(2.19)}$$

Çok-sınıflı sınıflandırmalarda ağırlıklı ortalama, mikro-ortalama ve makro-ortalama ölçümlerine göre duyarlılık hesaplaması denklem (2.20)'de gösterilmiştir (Sokolova & Lapalme, 2009).

$$Recall_{wa} = \frac{1}{k}\sum_{i=1}^{n}(k_i\ Recall)$$

$$Recall_{\mu} = \frac{\sum_{i=1}^{n} DP_i}{\sum_{i=1}^{n}(DP_i + YN_i)} \quad \textbf{(2.20)}$$

$$Recall_M = \frac{\sum_{i=1}^{n} \frac{DP_i}{DP_i + YN_i}}{n}$$

F-değeri (f-score)

İki-sınıflı sınıflandırmalarda f-değeri hesaplaması denklem (2.21)'de gösterilmiştir.

$$Fscore = \frac{2\ Precision\ Recall}{Precision + Recall} \quad \textbf{(2.21)}$$

Çok sınıflı sınıflandırmalarda ağırlıklı f-değeri hesaplaması denklem (2.22)'de gösterilmiştir.

$$Fscore_{wa} = \frac{1}{k}\sum_{i=1}^{n}(k_i\ Fscore) \quad \textbf{(2.22)}$$

Ortalama doğruluk (Average accuracy)

İki-sınıflı sınıflandırmalarda doğruluk değeri denklem (2.23)'te gösterilmiştir.

$$Accuracy = \frac{DP + DN}{DP + YN + YP + DN} \quad \textbf{(2.23)}$$

Çok-sınıflı sınıflandırmalarda yalnızca DP değerleri dikkate alınır (Li ve diğ, 2015). Her bir sınıfın DP değerleri toplamı, eğitim verisetindeki toplam örnek sayısına bölünerek denklem (2.24)'te gösterildiği gibi hesaplanır.

$$Accuracy = \frac{1}{k}\sum_{i=1}^{n} DP_i \quad \textbf{(2.24)}$$

Örneğin *Ankara, İstanbul, İzmir* ve *Van* olmak üzere dört sınıftan oluşan eğitim seti ile eğitilen sistem, sekiz adet veriden oluşan test verisi ile test edilmektedir. *Ankara* sınıfına ait iki doküman doğru sınıflandırılırken bir doküman yanlış (*İzmir olarak*) sınıflandırılmıştır. *İzmir* sınıfına ait iki doküman doğru sınıflandırılırken bir doküman yanlış (*İstanbul olarak*) sınıflandırılmıştır. *Van* sınıfına ait bir doküman doğru sınıflandırılırken bir doküman yanlış (*Ankara olarak*) sınıflandırılmıştır.

Her sınıf için hassasiyet (*P*), duyarlılık (*R*) ve f-değerleri (*F*), her sınıfın kendi karmaşıklık matrisi üzerinden hesaplanır. Bahsedilen örneğe göre sınıfların karmaşıklık matrisi aşağıdaki gibidir.

Çizelge 2.8 : Sınıfların karmaşıklık matrisleri.

Ankara

	+	-
+	2	1
-	1	4

$P = \frac{2}{2+1} = 0{,}667$

$R = \frac{2}{2+1} = 0{,}667$

$F = \frac{2\ P\ R}{P+R} = 0{,}667$

İstanbul

	+	-
+	0	0
-	1	7

$P = \frac{0}{0+1} = 0$

$R = \frac{0}{0} = 0$

$F = \frac{2\ P\ R}{P+R} = 0$

İzmir

	+	-
+	2	1
-	1	4

$P = \frac{2}{2+1} = 0{,}667$

$R = \frac{2}{2+1} = 0{,}667$

$F = \frac{2\ P\ R}{P+R} = 0{,}667$

Van

	+	-
+	1	1
-	0	6

$P = \frac{1}{1+0} = 1$

$R = \frac{1}{1+1} = 0{,}500$

$F = \frac{2\ P\ R}{P+R} = 0{,}667$

Başarı (Accuracy)

$$Accuracy = \frac{1}{8}(2 + 0 + 2 + 1) = 0{,}625 = 62{,}5\%$$

Mikro-ortalama hesaplamaları için tüm sınıfların karmaşıklık matrislerindeki DP, DN, YP, YN değerleri kendi aralarında toplanarak Çizelge 2.9'daki tüm sınıfların toplam karmaşıklık matrisi elde edilir.

Çizelge 2.9 : Tüm sınıflar için karmaşıklık matrislerinin toplamı.

		Hesaplanan	
		Pozitif	Negatif
Gerçekte olan	Pozitif	5	3
	Negatif	3	20

Makro-Ortalama Hesaplaması

$$Precision_M = \frac{0{,}667 + 0 + 0{,}667 + 1}{4} = 0{,}584$$

$$Recall_M = \frac{0{,}667 + 0 + 0{,}667 + 0{,}500}{4} = 0{,}459$$

$$Fscore_M = \frac{2\ x\ 0{,}584\ x\ 0{,}459}{0{,}584 + 0{,}459} = 0{,}514$$

Mikro-Ortalama Hesaplaması

$$Precision_\mu = \frac{5}{5 + 3} = 0{,}625$$

$$Recall_\mu = \frac{5}{5 + 3} = 0{,}625$$

$$Fscore_\mu = \frac{2\ x\ 0{,}625\ x\ 0{,}625}{0{,}625 + 0{,}625} = 0{,}625$$

Ağırlıklı Ortalama (Weighted Average)

$$Precision_{wa} = \frac{(0{,}667\ x\ 3) + (0\ x\ 0) + (0{,}667\ x\ 3) + (1\ x\ 2)}{8} = 0{,}750$$

$$Recall_{wa} = \frac{(0{,}667\ x\ 3) + (0\ x\ 0) + (0{,}667\ x\ 3) + (0{,}500\ x\ 2)}{8} = 0{,}625$$

$$Fscore_{wa} = \frac{(0{,}667\ x\ 3) + (0\ x\ 0) + (0{,}667\ x\ 3) + (0{,}667\ x\ 2)}{8} = 0{,}667$$

3. VERİSETİNİN OLUŞTURULMASI VE ÖZELLİKLERİ

Web sayfalarının sınıflandırılması amacıyla gerçek verilerden oluşan ve oldukça geniş bir konu alanını kapsayan DMOZ verileri ile çalışılmasına karar verilmiştir.

DMOZ (Directory Mozilla - The Open Directory Project), alanında uzman kişiler tarafından düzenlenmiş geniş kapsamlı web dizinidir. 15/10/2016 tarihi itibariyle DMOZ üzerinde 91.725 editör, 1.031.499 kategori içerisinde 3.890.990 web sayfası konularına uygun bir şekilde yerleştirilmiştir (DMOZ, 2017).

Tez kapsamında çalışılan Türkçe kategorilerin içerdiği web sayfaları hakkında DMOZ editörlerinin yazdıkları açıklamalar ve başlık bilgisinin yanında bu web sayfalarını işaret eden adreslerdeki içerik bilgileri alınarak 22 bin kayıttan oluşan geniş kapsamlı, Türkçe sınıflandırmaya uygun bir derlem oluşturulmuştur. Bu derlem, Türkçe metinler üzerinde yapılacak sınıflandırma çalışmalarına katkı sağlayabilecek bir kaynak niteliğindedir.

DMOZ, verilerini RDF (Resource Definition Framework – Kaynak Tanımlama Çerçevesi) veri tipinde açık kaynak olarak sunmaktadır. Kaynak tanımlama çerçevesi, web üzerindeki bilgiyi tasvir etmek amacıyla kullanılan ve W3C (World Wide Web Consortium) tarafından standartları belirlenen bir doküman modelidir. Veri tanımlamaları XML (eXtensible Markup Language) şemasına uygun bir şekilde yapılmaktadır (Klyne ve diğ, 2014; Beckett, 2014).

Bu çalışma yürütülürken 17/03/2017 tarihinde DMOZ'un çalışmalarına Curlie (curlie.org) üzerinde devam edeceği duyurulmuştur (DMOZ, 2017; Resource-Zone, 2017).

3.1 Verisetinin Oluşturulması

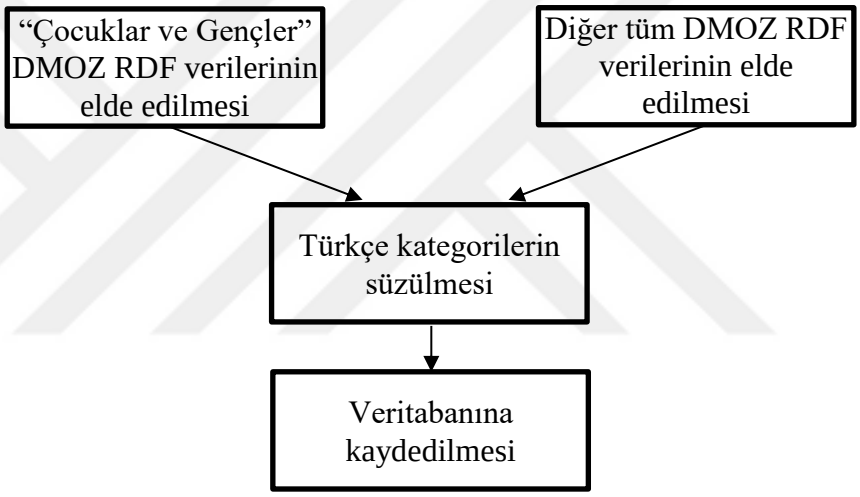
Veriseti, DMOZ tarafından verilen RDF tipinde dosyadan faydalanılarak aşağıdaki adımlar gerçekleştirilmiştir.

3.1.1 Türkçe kategorisindeki sayfaların elde edilmesi

DMOZ'dan elde edilen RDF düzenindeki veriler Java platformu kullanılarak MySQL veritabanına aktarılmıştır. Türkçe sayfalar üzerinde çalışılacağından veritabanı üzerinden Türkçe kategorisinde bulunmayan tüm sayfalar elenmiştir.

Türkçe kategorisine giren 54.609 adet kayıt bulunmaktadır. Sayfaların kategorilerine göre dağılımından başlık 3.3 altında ayrıntılı olarak bahsedilmiştir.

DMOZ üzerinde "Çocuklar ve Gençler (Kids and Teens)" kategorisi ayrı bir RDF dosyasında sunulmaktadır. Bu yüzden iki ayrı RDF dosyası birleştirilerek veriseti oluşturulmuştur. DMOZ verilerinin elde edilmesi ve üzerinde çalışabilmesi için gerekli verilerin veritabanına eklenme işlemleri Şekil 3.1'de gösterilmiştir.

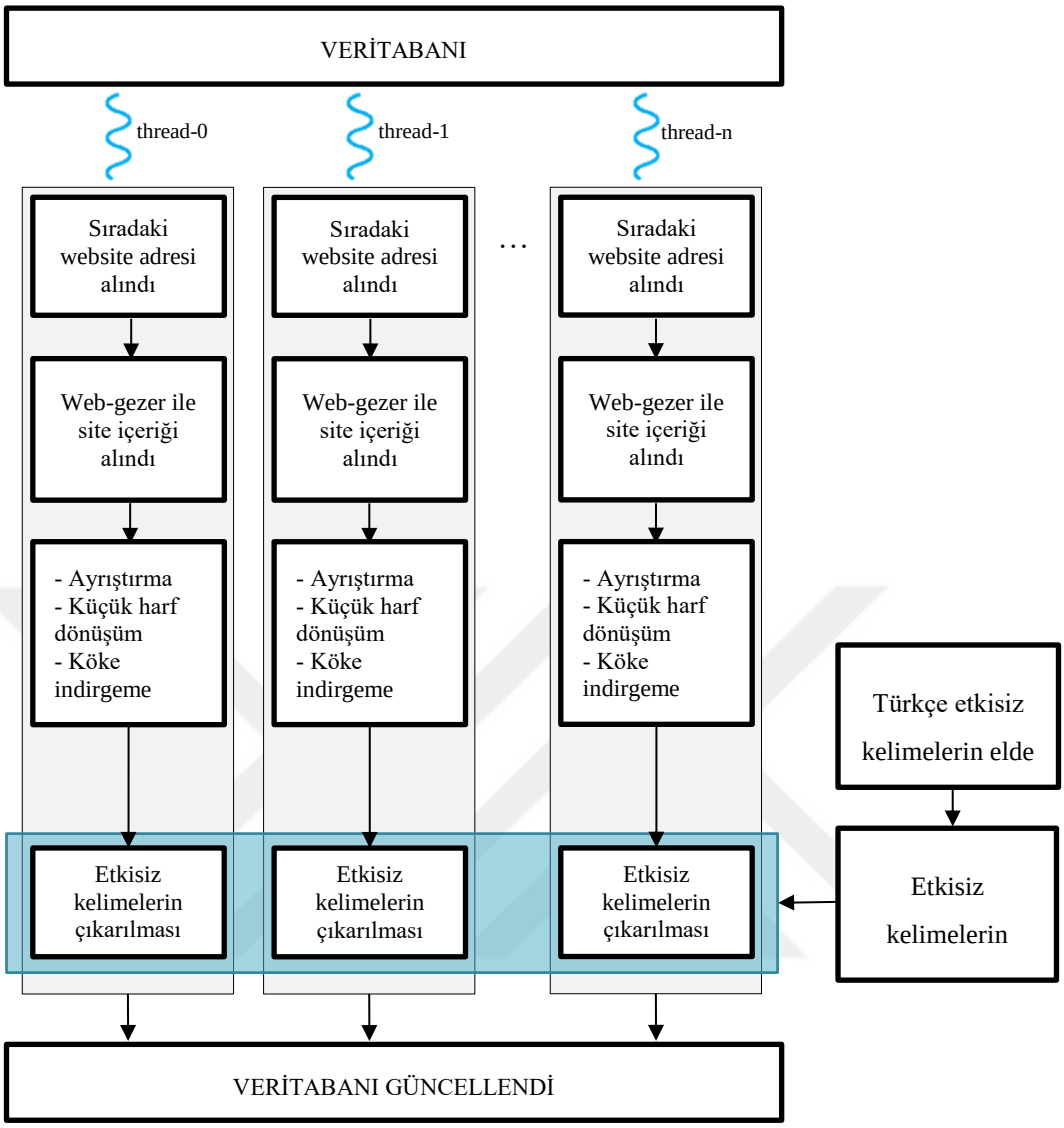


Şekil 3.1 : Türkçe verisetinin oluşturulma akış şeması.

3.1.2 Sayfaların dolaşımı

DMOZ verisetinden elde edilen Türkçe sayfaların gerçek site verilerine ulaşılması için bir web-gezer (crawler) tasarlanmıştır. Verisetindeki kayıt fazlalığından dolayı çoklu-kanal yöntemiyle (multithread) çalışılmıştır. Bu sayede web sayfalarının verileri daha hızlı elde edilmiştir.

Çoklu-kanal yöntemiyle veritabanındaki her web adresi ziyaret edilmiş ve içerikleri ön işlemlerden geçirilerek veritabanı güncellenmiştir. Yapılan işlemlerin akış diyagramı Şekil 3.2'de gösterilmiştir.



Şekil 3.2 : Sayfa dolaşımları, verisetinin elde edilmesi ve metin tabanlı ön işlemlerin çoklu-kanal ile gerçekleştirilmesi.

Sayfaların dolaşımından meta açıklamalar, meta kelimeler, sayfa başlığı ve içerik bilgileri elde edilmiştir. Elde edilen bu veriler veritabanına kaydedilmiştir.

Veriseti oluşturulurken kullanılan özellikler:

- **DMOZ Başlık ve Açıklaması:** DMOZ editörlerinin sayfalar hakkında uygun gördükleri başlık ve açıklamalardır.
- **İçerik:** *<BODY> ... </BODY>* etiketi içinde yer alan içeriğin tüm HTML etiketlerinden arındırılmış kelime topluluğudur.
- **Başlık:** *<TITLE> ... </TITLE>* etiketi içinde yer alan metindir.

- **Meta Veriler**: Açıklama için *<META name="description" content="...">* etiketinin *content* parametresinin değeri alınmıştır. Anahtar kelimler için *<META name="keywords" content="..." />* etiketinin *content* parametresinin değeri alınmıştır.

DMOZ tarafından kategorilendirilmiş bazı sayfalara dolaşım aşamasında ulaşılamamıştır. Ulaşılamayan sayfaların kullanımda olmadığı görülmüştür. DMOZ Türkçe kategorisine giren sayfaların geçerliliğinin daha sık aralıklarla tespit edilip güncellenmesi, sitenin amacı açısından daha yararlı olacaktır.

3.2 Verisetinin İşlenmesi

DMOZ'dan ve sayfa dolaşımlarından elde edilen sayfa içerikleri üzerinde sınıflandırma işlemi yapmadan önce ön işlemler uygulanır. Yapılan önişlem adımları aşağıda anlatılmıştır.

Ayrıştırma

Türkçe metinler üzerinde yapılan işlemlerde Zemberek 2.1 kütüphanesi kullanılmıştır. Java platformu üzerinde Zemberek kütüphanesinden yararlanılarak verisetindeki metinler üzerinde aşağıdaki işlemler gerçekleştirilmiştir.

Web sayfalarından toplanan veriler içinde sınıflandırmada yararı olmayacak noktalama işaretleri çıkarılmıştır. Sonraki adımlarda kelimelerin yapım ve çekim eklerinden temizlenerek kökleri ile işlem yapılabilmesi için veriseti kelimelere ayrılmıştır. Bunların yanında web sayfalarının doğası gereği içerdikleri HTML etiketleri, JavaScript betikleri ve CSS kodları gibi metin sınıflandırmaya etkisi olmayacak kodlamalar bu aşamada verisetinden çıkarılmıştır.

Tüm karakterlerin küçük harflere çevrilmesi

Site başlığı, meta kelime, meta açıklama ve sitelerin gerçek içeriğindeki tüm metinler, DMOZ verisetinden alınan verilerdeki açıklama ve başlıklarla birlikte küçük harflere çevrilmiştir.

Köke indirgeme

Sınıflandırma öncesi kelimelerde bulunan yapım ve çekim ekleri atılarak tüm kelimler kök haline indirgenmiştir. Zemberek'te bulunmayan yabancı kelimeler ile

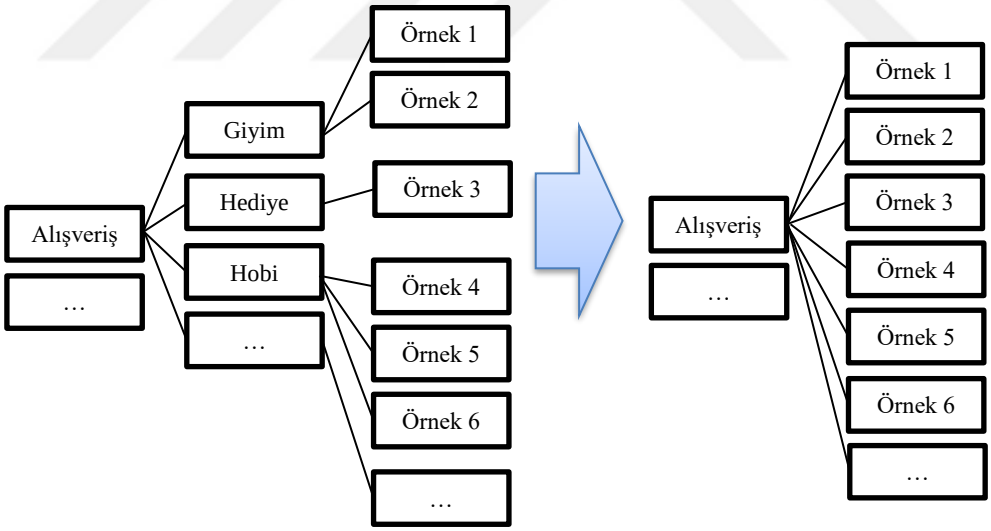
yanlış veya eksik yazılmış kelimeler verisetinden çıkarılmıştır. DMOZ üzerindeki başlık ve sitenin kendi başlık bilgisi bu kapsam dışında tutulmuştur. Çünkü birçok site adı "hepsiburada.com", "YÖK", "BİMER" örneklerinde olduğu gibi, doğrudan Türkçe sözlükte bulunmayan kelimeleri içermektedir.

Etkisiz kelimelerinin çıkarılması

Türkçe etkisiz kelimeler, verisetinden çıkarılmadan önce köke indirgenmiştir. Bu sayede, örneğin "hiçbir" kelimesinin "hiçbiri", "hiçbirimiz", "hiçbiriniz" gibi tüm çekimlerinin düşünülme zorunluluğu ortadan kaldırılmıştır. Kök haline indirgenmiş etkisiz kelimeler, kelimeleri önceden kök haline indirgenmiş verisetinden çıkarılmıştır.

3.3 Verisetinin Özellikleri

DMOZ üzerindeki ana kategoriler ile çalışılmıştır. Ana kategoriler altındaki alt kategoriler altındaki siteler, doğrudan ilgili ana kategori altında Şekil 3.3'te gösterildiği gibi toplanmıştır. Alt kategori etiketleri kullanılmamıştır.



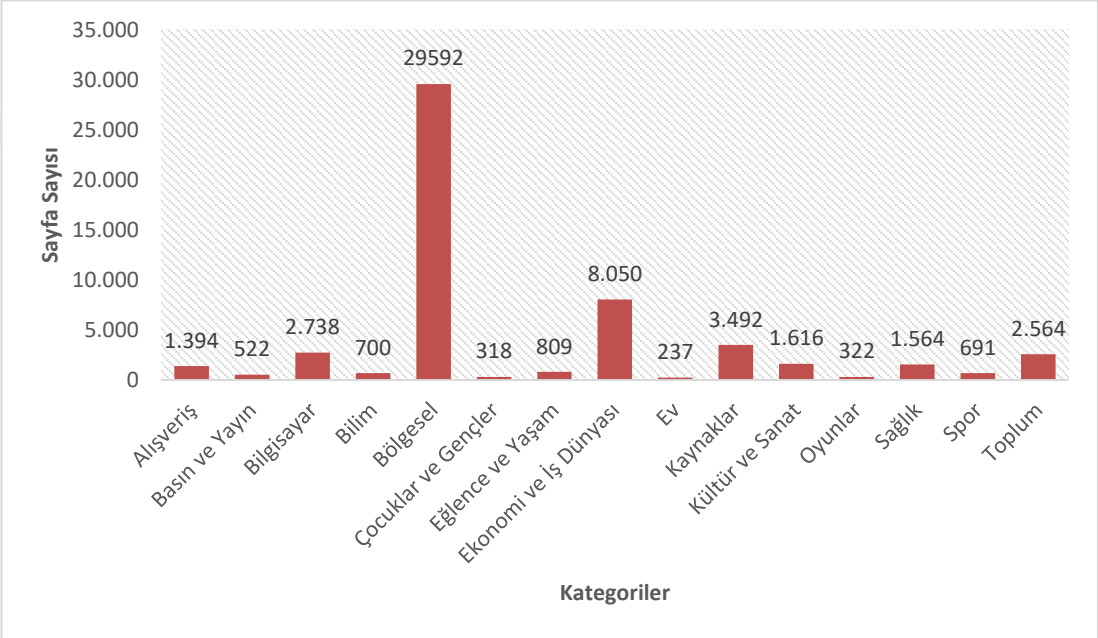
Şekil 3.3 : Alt kategorilerin ait olduğu ana kategoriye dahil edilmesi.

DMOZ tarafından tüm sayfalar 15 ana kategori altında toplanmıştır. Bu kategoriler Çizelge 3.1'de gösterilmiştir.

Çizelge 3.1 : DMOZ Kategorileri.

Alışveriş	Çocuklar ve Gençler	Kültür ve Sanat
Basın ve Yayın	Eğlence ve Yaşam	Oyunlar
Bilgisayar	Ekonomi ve İş Dünyası	Sağlık
Bilim	Ev	Spor
Bölgesel	Kaynaklar	Toplum

Çalışmanın konusu olan Türkçe web sayfaları ve kategorileri, DMOZ üzerinde "World/Türkçe" konumu ve "Kids_and_Teens/International/Türkçe" konumu altında bulunmaktadır. Tüm Türkçe sitelerin sayısı 15/10/2016 tarihi itibariyle 54.609 olarak tespit edilmiştir. Verisetindeki sayfaların dağılımı Şekil 3.4'te gösterilmiştir.

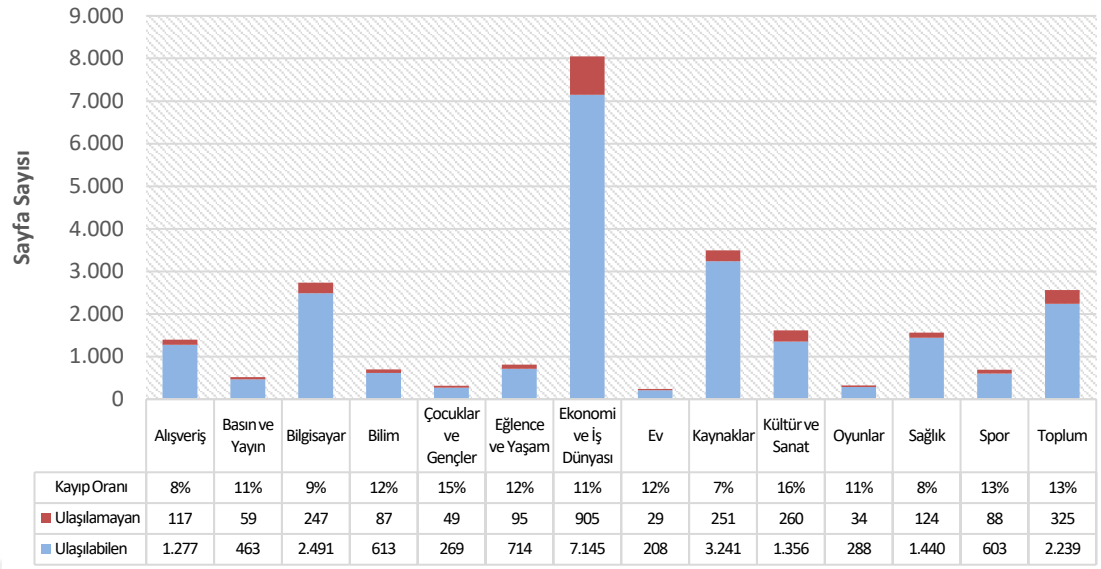


Şekil 3.4 : DMOZ'dan alınan Türkçe sayfaların kategorilerine göre dağılım grafiği.

"Bölgesel" kategorisinin altındaki alt kategoriler, DMOZ'un ana kategorileri ile benzer konu başlıklarına sahiptir. Bunun yanında bu kategoride, toplam site sayısının yarısından fazlasının olması (%54) sınıflandırma performansında ve amacı açısından sorun oluşturacağından bu kategori çalışma dışında bırakılmıştır. "Bölgesel" kategorisinde 15/10/2016 tarihi itibariyle 29.592 adet site bulunmaktadır.

Sayfaların kategorilere göre dağılımı

Verisetinde bulunan 2.670 adet sayfaya ulaşılamadığından çalışılacak olan toplam örnek sayısı 22.347'ye inmiştir. Web-gezer ile ulaşılabilen sayfaların DMOZ kategorilerine göre dağılımları Şekil 3.5'te gösterilmiştir.



	Alışveriş	Basın ve Yayın	Bilgisayar	Bilim	Çocuklar ve Gençler	Eğlence ve Yaşam	Ekonomi ve İş Dünyası	Ev	Kaynaklar	Kültür ve Sanat	Oyunlar	Sağlık	Spor	Toplum
Kayıp Oranı	8%	11%	9%	12%	15%	12%	11%	12%	7%	16%	11%	8%	13%	13%
Ulaşılamayan	117	59	247	87	49	95	905	29	251	260	34	124	88	325
Ulaşılabilen	1.277	463	2.491	613	269	714	7.145	208	3.241	1.356	288	1.440	603	2.239

Şekil 3.5 : Bölgesel kategorisi haricinde ulaşılabilen ve ulaşılamayan sayfaların DMOZ kategorilerine göre dağılım grafiği.

3.4 Sayfaların Özelliklerine Göre Deney Verisetlerinin Oluşturulması

DMOZ verisetinden, birbirinden farklı altı adet deney veriseti oluşturulmuştur. Bu çalışmadaki amaç; kategori sayısı, eksiksiz ve eğitimde kullanılan içeriğin sınıflandırma başarısına etkisinin daha net izlenebilmesidir. Deney verisetleri oluşturulurken eğitim verilerine dahil edilen içerik Çizelge 3.2'de gösterilmiştir. Farklı sayfa içeriklerinin sınıflandırma başarısına etkisinin daha net gözlenebilmesi için 3, 4, ve 5. deney verisetleri oluşturulmuştur. Bu sayede DMOZ'daki başlık ve açıklamaların sitenin kendisinden elde edilen verilerle birlikte değerlendirildiğinde sınıflandırma başarısına katkısı incelenmiştir.

Çizelge 3.2 : Oluşturulan deney verisetlerinin eğitim verilerinde kullanılan özellikler.

DENEY VERİSETLERİNİN EĞİTİM VERİLERİ	DMOZ Başlık ve Açıklama	İçerik	Başlık	META veriler
1	+	+	+	+
2	+	+	+	+
3	+	-	-	-
4	-	+	+	+
5	+	+	+	+
6	+	+	+	+

Herhangi bir DMOZ kategorisinde bulunmayan web sayfalarının sınıflandırılması üzerinde durulduğundan oluşturulan tüm deney verisetlerinin test verilerinde sayfaların kendi içerikleri kullanılmıştır. DMOZ'dan elde edilen başlık ve açıklamalar test verilerine dahil edilmemiştir. Test verilerinin özellikleri Çizelge 3.3'te gösterilmiştir.

Çizelge 3.3 : Oluşturulan deney verisetlerinin test verilerinde kullanılan özellikler.

DENEY VERİSETLERİNİN TEST VERİLERİ	DMOZ Başlık ve Açıklama	İçerik	Başlık	META veriler
1, 2, 3, 4, 5, 6	**-**	**+**		

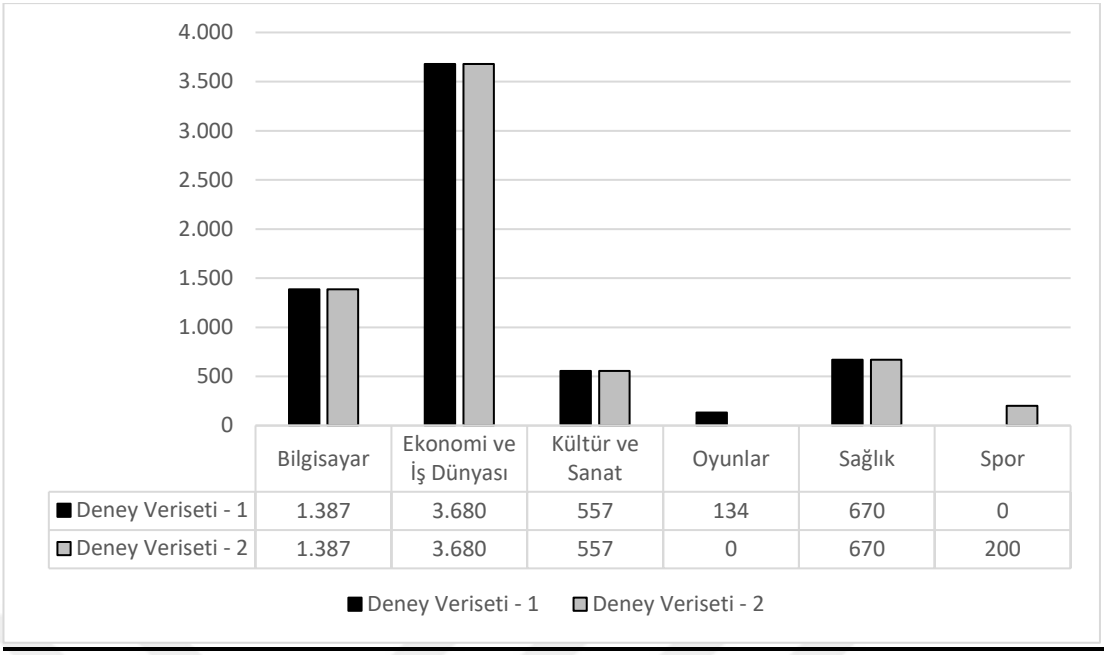
Deney verisetlerindeki örnek sayıları, eksiksiz veriler ile çalışılıp çalışılmadığına ve kategori sayılarına göre değişmektedir. Çizelge 3.4'te gösterilen 1, 2 ve 6. deney verisetlerinde, site başlık ve meta verileri tam olan sayfalar ile çalışıldığından bu deney verisetlerinin kategorilerindeki örnek sayıları, Şekil 3.5'te gösterilen örnek sayılarına göre daha azdır.

Çizelge 3.4 : Oluşturulan deney verisetlerindeki sayfaların kategorilere göre dağılımları.

Kategori	VERİSETLERİNDEKİ SAYFA SAYILARI					
	1	**2**	**3**	**4**	**5**	**6**
Alışveriş	-	-	1.277			876
Basın ve Yayın	-	-	463			232
Bilgisayar	1.387	1.387	2.491			1.387
Bilim	-	-	613			188
Çocuklar ve Gençler	-	-	269			114
Eğlence ve Yaşam	-	-	714			350
Ekonomi ve İş Dünyası	3.680	3.680	7.145			3.680
Ev	-	-	208			101
Kaynaklar	-	-	3.241			825
Kültür ve Sanat	557	557	1.356			557
Oyunlar	134	-	288			134
Sağlık	670	670	1.440			670
Spor	-	200	603			200
Toplum	-	-	2.239			927
TOPLAM	**6.428**	**6.494**	**22.347**			**10.241**

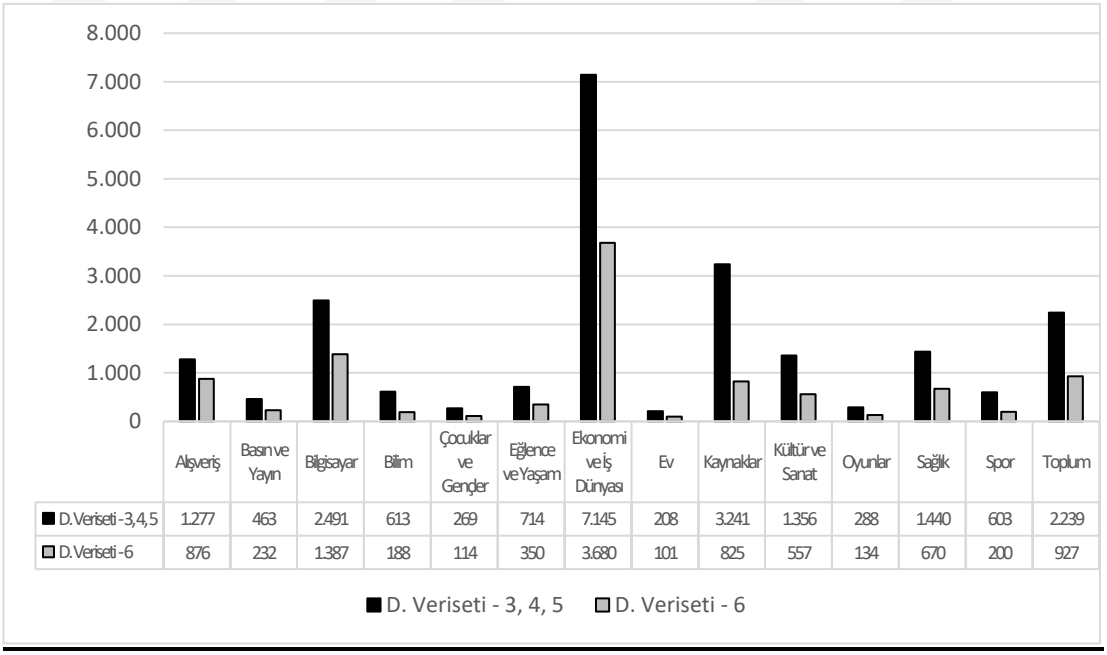
Deney veriseti - 1 ve 2:

Bu deney verisetlerinde beşer kategori ile çalışılmıştır. Deney veriseti 1 ve 2'nin karşılaştırması Şekil 3.6'da gösterilmiştir.



Şekil 3.6 : Deney veriseti 1 ve 2'deki sayfaların kategorilere dağılımı ve karşılaştırılma grafiği.

Deney veriseti – 3, 4 ve 5'te sayfa içeriğinde eksiklikleri bulunan örnekler de dahil edilirken deney veriseti - 6'da içeriği eksiksiz örnekler seçilmiştir. Örneklerin kategorilere göre dağılımı ve karşılaştırılması Şekil 3.7'de verilmiştir.



Şekil 3.7 : Deney veriseti 3, 4, 5 ile deney veriseti 6'daki örneklerin kategorilere göre dağılımı ve karşılaştırılma grafiği.

3.5 Veriseti Oluşturmada Yaşanan Zorluklar

Veriseti oluşturulma aşamasında birtakım zorluklar yaşanmıştır. Yaşanan zorluklar aşağıdaki başlıklarda anlatılmıştır.

Çalışılan sayfa sayısının fazlalığı

Çalışılan sayfa sayısının fazlalığından dolayı bu sayfaların gezilerek site verilerinin elde edilmesi ciddi bir zaman kaybına sebep olmaktadır. Bu sorunun çözümü için çok-kanallı çalışan bir web-gezer tasarlanmıştır. Böylece toplamda 25.017 sayfa dolaşımından kaynaklı zaman kaybının önüne geçilmiştir.

Farklı RDF dosyaları

"Çocuklar ve Gençler" kategorisi, DMOZ tarafından verilen genel RDF verisetinde bulunmamaktadır. Bu kategori için yine DMOZ üzerinde ayrı bir RDF dosyası bulunmaktadır. Bu sebeple, "Çocuklar ve Gençler" kategorisi veritabanına ayrıca eklenmiştir.

Ulaşılamayan sayfalar

DMOZ kategorileri içinde, artık kullanımda olmayan sayfalar da olduğundan web-gezer ile içeriklerine ulaşılamayan sayfalar bulunmaktadır. Ulaşılamayan sayfalar nedeniyle bu örnekler, eğitim ve test verilerinde çalışma kapsamına alınmamıştır.

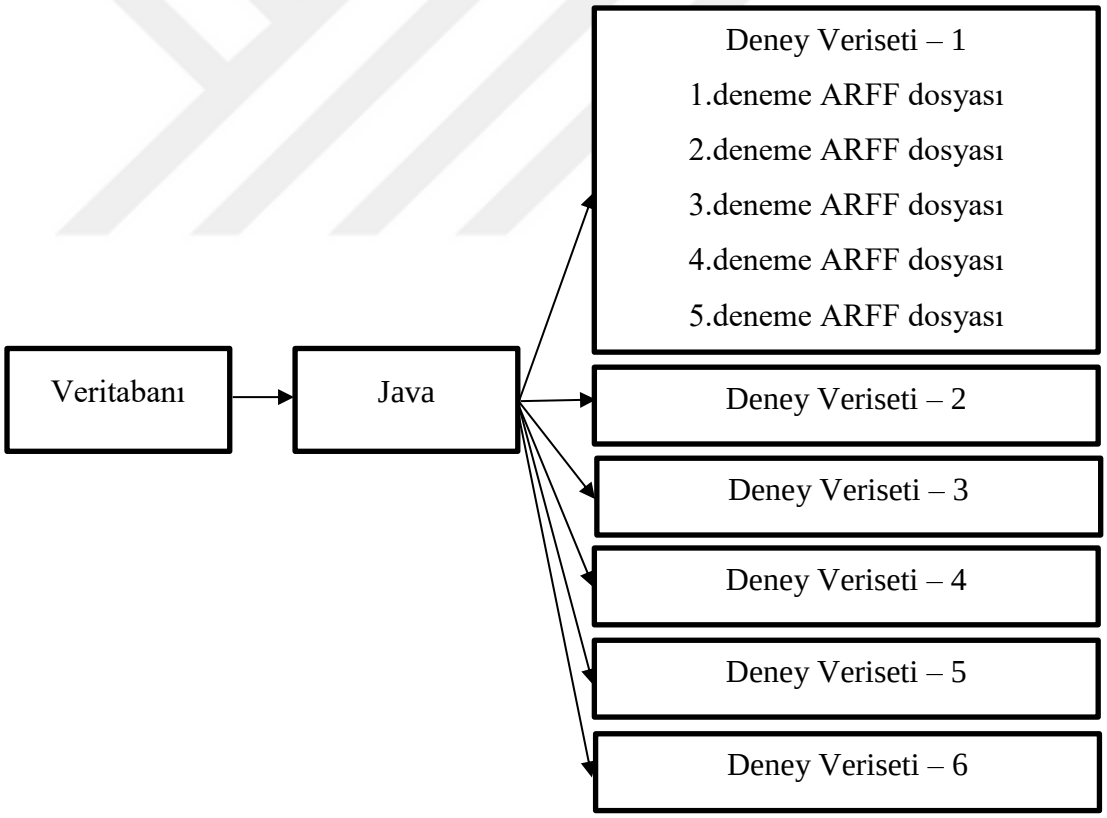
Dolaşılan Sayfalardaki Eksik Veriler

Tez kapsamındaki denemelerde sayfaların başlık, içerik, meta kelimeler ve meta açıklamalar etiketlerinin içindeki veriler kullanılmıştır. Sayfaların HTML kodlamasındaki eksiklikler nedeniyle sayfa hakkında edinilen bilgi konusunda olumsuz etki oluşmaktadır. Sayfaya ait özelliklerin eksik olması, sınıflandırmayı olumsuz etkilemektedir.

4. UYGULAMA

Java platformunda, RDF verilerinin yorumlanması için RDF'ten MySQL veritabanına aktaran ve WEKA'da kullanılmak üzere ARFF dosyası oluşturan metin-ayrıştırıcı ile sayfaların güncel bilgilerini elde eden bir web-gezer yazılmıştır. Veriseti, veri madenciliği yöntemleri kullanılarak WEKA programı yardımıyla test edilmiştir.

Java platformu ile veritabanından alınan verisetinden altı adet deney veriseti ve her deney veriseti için Şekil 4.1'de gösterilen yapıda farklı ARFF dosyaları oluşturulmuştur.



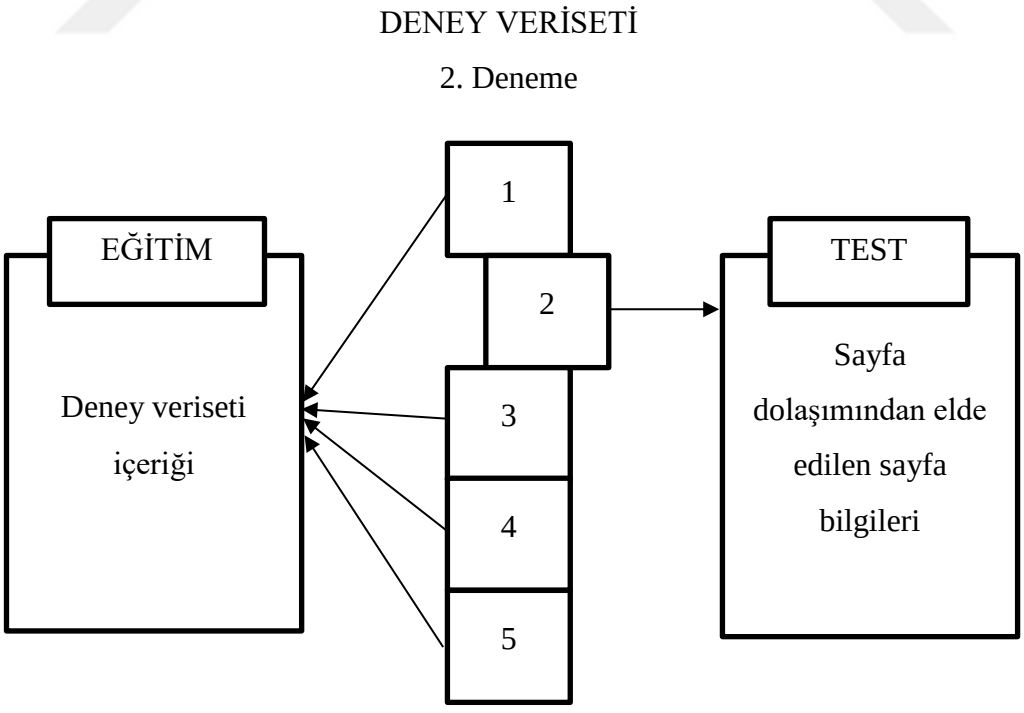
Şekil 4.1 : Altı adet deney verisetinin oluşturulması.

Sınıflandırma işleminde dikkate alınacak kelimeler üzerinde sınıflandırma için uygulanması gereken metin tabanlı ön işlemler dışında hiçbir müdahale olmadan veriseti, DMOZ ve sayfa içeriklerinden elde edilen kelimeler ile hazırlanmıştır.

4.1 Doğrulama Yöntemi

Deney verisetlerinde dışarı bırakma (hold-out) yöntemi kullanılmıştır. Eğitim ve test deney verisetlerinde kullanılan özellikler farklı olduğu için eğitim ve test verileri iki farklı dosya şeklinde hazırlanmıştır. WEKA platformuna eğitim ve test verileri ayrı ayrı verilerek sınıflandırma işlemleri yapılmıştır.

Her deney veriseti için 5 farklı eğitim ve test veriseti oluşturulmuştur. Böylece her bir sayfanın, mutlaka bir test verisetinde yer alması sağlanmıştır. Şekil 4.2'de herhangi bir deney veriseti için, örneğin 2. deneme aşamasında oluşturulan eğitim ve test verilerinin nasıl seçildiği gösterilmektedir. Bu işlem, her deney veriseti için beş kez tekrar etmektedir. Bir deney seti için 5'er adet eğitim ve test verisi bulunmaktadır. Altı deney verisetinde, beş adet deneme ile *6 x 5 = 30* adet eğitim ve test verisiyle çalışılmıştır.



Şekil 4.2 : Deney verisetleri için eğitim ve test verisetlerinin oluşturulma yöntemi.

Çizelge 4.1'de gösterildiği gibi tez kapsamında iki farklı algoritma, üç adet özellik vektörü ve bilgi kazanım oranı kullanılıp kullanılmamasına göre *30 x 2 x 3 x 2 = 360* adet test yapılmıştır. Yapılan tüm işlemler farklı test verileri ile test edilmiş olup ortalamaları sonuç olarak yansıtılmıştır. Bu şekilde çalışılmasındaki amaç gerçek bir test ortamı oluşturmak için test verilerinde DMOZ verilerinin bulunmamasını sağlamaktır. Eğitim ve test verilerindeki sayfalar, içerikleri bakımından farklılık gösterdiği için her deney veriseti için beş adet eğitim ve test verisi ayrı dosyalarda oluşturulmuştur.

Çizelge 4.1 : Verisetleri ile yapılan deney sayıları.

HER DENEY VERİSETİ İÇİN	**Kelime vektörü**	**ALGORİTMA**			
		M-NB	**DVM**	**Bilgi kazanım oranlı M-NB**	**Bilgi kazanım oranlı DVM**
	1-gram	5	5	5	5
	2-gram	5	5	5	5
	1..2-gram	5	5	5	5
Bir deney veriseti için yapılan toplam deney sayısı : 60					
Altı farklı deney seti için yapılan toplam deney sayısı : 360					

4.2 Kelime N-Gram Özellik Vektörü Çıkarımı

Vektör uzayını küçültmek ve başarıyı artırmak adına en yüksek frekansa sahip ilk 10.000 adet kelime ile çalışılmıştır. Sınıflandırma işlemlerinde kelime n-gram özellik çıkarım yöntemiyle çalışılmaya karar verildiğinden n-gram denemeleri sonucunda başarı sonuçlarına bağlı olarak bu çalışmada kelime 1-gram, 2-gram ve 1..2-gram ile çalışmaya karar verilmiştir.

Örneğin metin sınıflandırmada yapılması gereken ön işlemlerden geçirilmiş "bilgisayar ata say hesap aygıt abaküs" ifadesi üzerinde 1-gram, 2-gram ve 1..2-gram kelime vektörü kullanımı Çizelge 4.2'de gösterilmektedir.

Çizelge 4.2 : Kelime 1..2-gram yönteminin örneklendirmesi.

1-gram	bilgisayar, ata, say, hesap, aygıt, abaküs
2-gram	bilgisayar ata, ata say, say hesap, hesap aygıt, aygıt abaküs
1..2-gram	bilgisayar, ata, say, hesap, aygıt, abaküs, bilgisayar ata, ata say, say hesap, hesap aygıt, aygıt abaküs

4.3 Test Sonuçları

Çalışılan deney verisetlerinin özellikleri ve kategorilerine göre sayfa dağılımları başlık 3.3'te anlatılmıştır. Verisetleri; kategori sayısı, farklı içerikle eğitilen sistem ve içerik kalitesi bakımından incelenmiştir. Her bir başlık kendi içinde özellik vektörü ve bilgi kazanım oranının sınıflandırmaya katkısı ile detaylandırılmıştır.

Kategori sayısı farklı deney verisetlerinin sınıflandırma başarılarının gözlenmesi için 1, 2 ve 6. deney verisetleri ile çalışılmıştır. 1 ve 2. deney verisetleri 5 kategoriden, 6. deney veriseti 14 kategoriden oluşmuştur. Örneklerin kategorilere göre dağılımı Çizelge 3.4'te gösterilmiştir. 1, 2 ve 6. deney verisetlerinin birlikte karşılaştırılmalarının sebebi, eğitim verileri oluşturulurken kullanılan içerik bilgilerinin aynı olması ve içerik bilgilerinde eksiksiz verilerin kullanılmasıdır.

Farklı içerik ile eğitilen verisetlerinin sınıflandırma başarılarının gözlenmesi için 3, 4 ve 5. deney verisetleri ile çalışılmıştır. Bu deney verisetlerinin test edilmesindeki amaç sistemin eğitiminde kullanılan içeriklerin farklılıklarının sınıflandırma başarısına etkilerinin gözlenmek istenmesidir. 3, 4 ve 5. deney verisetleri 14 kategoriden oluşmaktadır. Deney verisetleri oluşturulurken kullanılan özellikler Çizelge 3.2'de gösterilmiştir. Kullanılan özelliklerin ayrıntıları başlık 3.3 altında anlatılmıştır.

Deney verisetleri içerisinde sayfa içeriklerinden elde edilen <TITLE> etiketinden alınan başlık bilgisi, <META> etiketinden alınan açıklama ve anahtar kelime bilgileri eksik olan sayfaların bulunmaması durumunda sınıflandırma başarısına etkisini görebilmek için 5 ve 6. deney verisetleri ile çalışılmıştır. 5 ve 6. deney verisetlerinde eğitim verilerinde kullanılan içerik bilgileri aynıdır. 6. Deney veriseti oluşturulurken eksik içerikli sayfalar olmamasına dikkat edilmiştir.

4.3.1 Kategori sayısı farklı verisetleri için test sonuçları

Kategori sayısı farklı olan 1, 2 ve 6. deney verisetleri için M-NB ve DVM algoritmalarında kelime vektörü seçiminin ve BKO yaklaşımının kullanıldığı sınıflandırma sonuçlarının doğruluk ve f-değerleri Çizelge 4.3'te gösterilmektedir.

Çizelge 4.3 : Kategori sayısına göre kelime n-gram özellik vektörü seçimi ve BKO'nun sınıflandırma başarısına etkisi.

Kategori Sayısı	Örnek Sayısı	Veriseti	Algoritma	1-gram		2-gram		1..2-gram	
				Doğruluk	F ölçümü	Doğruluk	F ölçümü	Doğruluk	F ölçümü
5	6,428	1	M-NB	89,98% σ = 0,009	0,902	**92,61%** σ = 0,006	0,926	91,97% σ = 0,009	0,921
			DVM	**85,37%** σ = 0,201	0,849	80,26% σ = 1,487	0,788	83,87% σ = 1,103	0,830
			BKO + M-NB	90,04% σ = 0,010	0,903	**91,20%** σ = 0,010	0,913	90,83% σ = 0,010	0,910
			BKO + DVM	**86,94%** σ = 3,289	0,868	85,45% σ = 5,513	0,851	86,92% σ = 4,257	0,866
5	6,494	2	M-NB	89,80% σ = 0,008	0,900	**92,39%** σ = 0,009	0,924	91,60% σ = 0,008	0,917
			DVM	**85,18%** σ = 0,802	0,847	79,71% σ = 1,325	0,781	83,14% σ = 0,744	0,821
			BKO + M-NB	89,94% σ = 0,007	0,901	**90,05%** σ = 0,023	0,902	89,91% σ = 0,018	0,901
			BKO + DVM	89,83% σ = 0,691	0,900	**90,89%** σ = 1,060	0,910	90,59% σ = 0,907	0,910
14	10,241	6	M-NB	77,07% σ = 0,009	0,770	**79,45%** σ = 0,007	0,786	78,72% σ = 0,008	0,778
			DVM	**70,31%** σ = 0,789	0,693	65,36% σ = 0,761	0,629	69,06% σ = 0,531	0,672
			BKO + M-NB	76,79% σ = 0,007	0,770	76,67% σ = 0,004	0,762	**77,90%** σ = 0,006	0,781
			BKO + DVM	69,79% σ = 0,865	0,688	66,77% σ = 0,695	0,652	**72,04%** σ = 0,931	0,711

Çizelge 4.3 incelendiğinde, uygulanan tüm algoritmalarla kategori sayısı az olan verisetlerinde daha yüksek doğruluk ve f-değerlerine ulaşılmıştır. Kategori sayısı arttığında elde edilen sınıflandırma başarılarının düştüğü gözlenmiştir.

Az kategorili deney verisetlerinde en yüksek doğruluk oranı M-NB algoritması ile elde edilen %92,61'dir. Her bir sınıf için kelime ağırlıkları, algoritmaların doğası gereği otomatik bir şekilde yapılmıştır. İnsan müdahalesiyle kategoriler için en iyi anahtar kelime seçim işlemi yapılmamıştır. Kategoriler için önemli olabilecek anahtar kelimelerin el ile seçilmesiyle otomatik sınıflandırmanın dışına çıkılarak daha yüksek doğruluk oranlarına ulaşılabildiği görülmektedir (Gürcan, 2009).

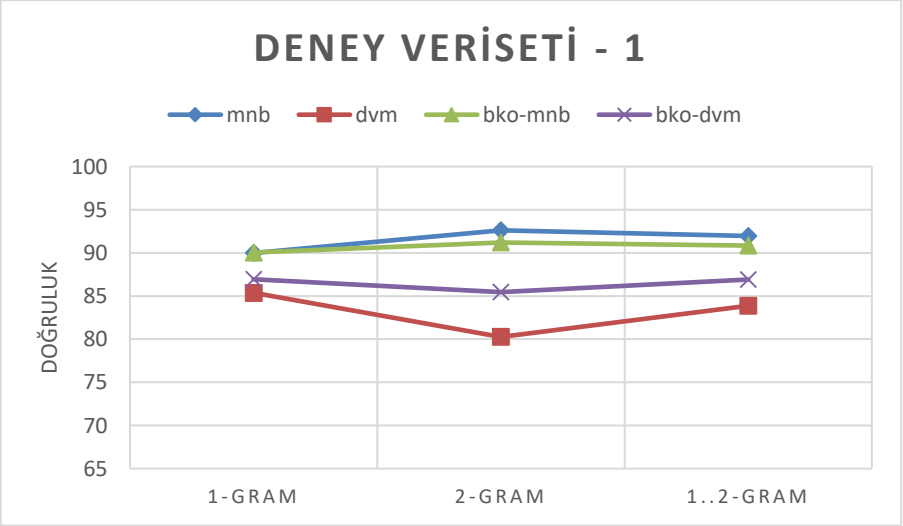
BKO yaklaşımının, kelime 1-gram için doğruluk ve f-değeri sonuçlarına bakıldığında az kategorili 1. ve 2. deney verisetlerinde M-NB algoritmasının performansında ciddi bir artışa sebep olmadığı gözlenmiştir. Aynı şekilde çok kategorili 6. deney

verisetinde de M-NB algoritmasının başarısına katkısında ciddi bir etkisi görülmemiştir. Kategori sayısı az olan deney verisetlerinde 1-gram için BKO yaklaşımının, DVM algoritmasının performansına olumlu katkı sağlarken çok kategorili verisetinde başarıya ciddi bir etkisi görülmemektedir.

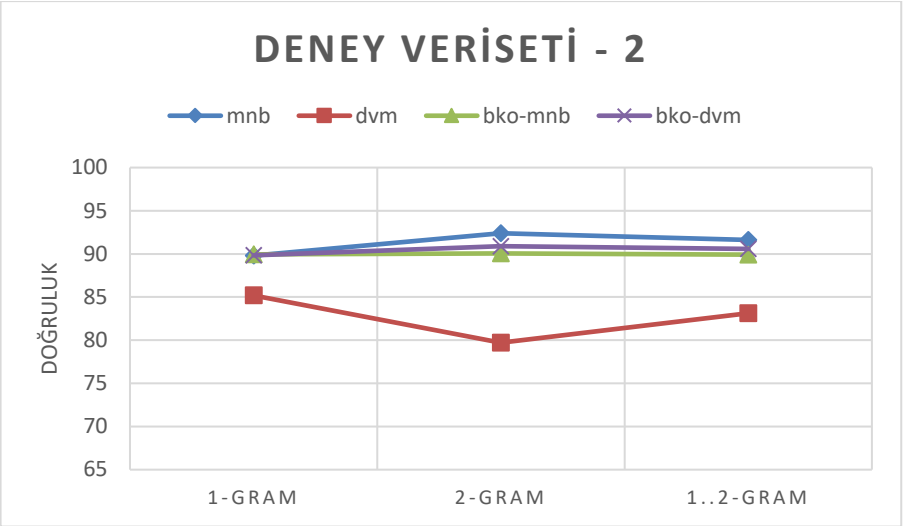
BKO yaklaşımının 1, 2 ve 6. deney verisetlerinde 2-gram özellik vektörü için doğruluk ve f-değerlerine bakıldığında M-NB algoritmasının performansına etkisinin olumsuz yönde %1,5 - %3 oranında olduğu görülmektedir. 2-gram için aynı deney verisetlerinde BKO yaklaşımının, DVM algoritmasının performansına olumlu yönde katkı sağladığı görülmüştür. Az kategoriye sahip 1 ve 2. deney verisetlerinde %5'ten fazla artış gözlenirken çok kategorili 6. deney verisetinde yaklaşık %1,5 değerinde bir artış gözlenmiştir.

BKO yaklaşımının 1..2-gram için 1, 2 ve 6. deney verisetleri incelendiğinde M-NB algoritma performansının olumsuz yönde etkilendiği görülmektedir. 1..2-gram için kategori sayısı az olan deney verisetlerinde BKO yaklaşımının DVM algoritmasının performansına %3'ten fazla olumlu katkı sağlarken 6. deney verisetinde yaklaşık %3 civarında olumlu katkı sağladığı görülmüştür.

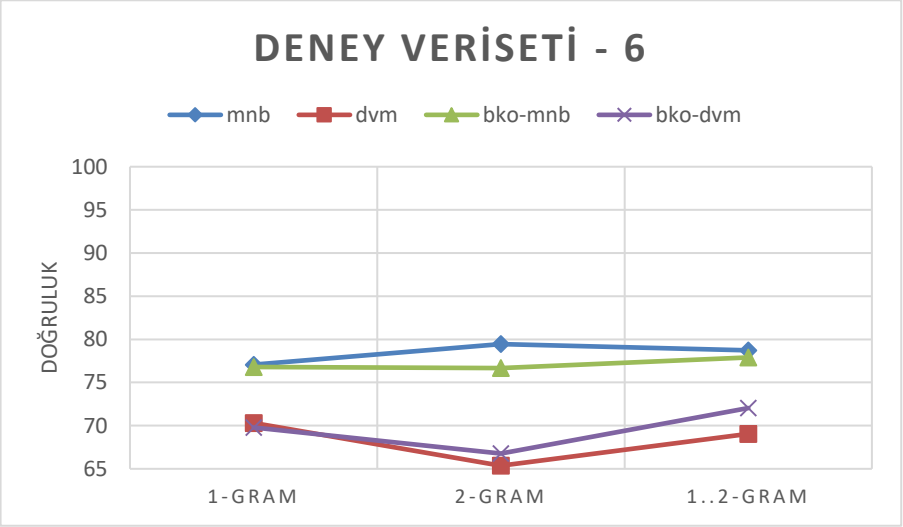
Kelime vektörü seçiminin uygulandığı 1, 2 ve 6. deney verisetlerinde M-NB, DVM, BKO+M-NB ve BKO+DVM algoritmalarının doğruluk değerleri değişimleri Şekil 4.3, Şekil 4.4 ve Şekil 4.5'te gösterilmiştir.



Şekil 4.3 : 1. deney verisetinin kelime vektörü seçimine göre algoritmaların doğruluk değerleri grafiği.



Şekil 4.4 : 2. deney verisetinin kelime vektörü seçimine göre algoritmaların doğruluk değerleri grafiği.



Şekil 4.5 : 6. deney verisetinin kelime vektörü seçimine göre algoritmaların doğruluk değerleri grafiği.

Çizelge 4.3'teki test sonuçlarına göre kelime n-gram özellik vektörü seçiminin sınıflandırma başarısına etkisi incelendiğinde aşağıdaki sonuçlar görülmektedir.

M-NB algoritmasının 1, 2 ve 6. deney verisetlerinde 1-gram referans alındığında doğruluk ve f-değerlerine bakıldığında 2-gram ve 1..2-gram özellik vektörü seçimlerinin daha iyi sonuçlar verdiği gözlenmiştir. Aynı deney verisetlerinde M-NB algoritmasının en yüksek başarısının kelime 2-gram özellik vektöründe olduğu gözlenmiştir. BKO yaklaşımlı M-NB algoritmasının da doğruluk ve f-değerleri

incelendiğinde kelime 2-gram özellik vektörü seçiminin diğerlerine oranla daha iyi sonuçlar verdiği gözlenmiştir.

DVM algoritmasında 1, 2 ve 6. deney verisetlerinde kelime 1-gram özellik vektörü seçiminin daha iyi sonuçlar verdiği gözlenmiştir. 1..2-gram özellik vektörünün doğruluk ve f-değerlerindeki artışta 1-gram'ın olumlu etkisi olduğu görülmektedir. BKO yaklaşımlı DVM algoritmasında kelime 1..2-gram özellik vektörü yaklaşımının diğerlerine oranla daha yüksek doğruluk ve f-değerlerine sahip olduğu gözlenmiştir.

Uygulanan algoritmaların 1, 2 ve 6. deney verisetlerindeki sonuçlarının derecelendirilmesi Çizelge 4.4'te gösterilmiştir. Çizelge üzerinde en başarılı algoritma 1'den başlayarak derecelendirilmiştir.

Çizelge 4.4 : 1, 2 ve 6. deney verisetleri için algoritma sonuçlarının derecelendirilmesi.

Veriseti	**Özellik Vektörü**	**Algoritmalar**			
		M-NB	**DVM**	**BKO+M-NB**	**BKO+DVM**
1	1-gram	2	4	1	3
	2-gram	1	4	2	3
	1..2-gram	1	4	2	3
2	1-gram	3	4	1	2
	2-gram	1	4	3	2
	1..2-gram	1	4	3	2
6	1-gram	1	3	2	4
	2-gram	1	4	2	3
	1..2-gram	1	4	2	3
Toplam		**12**	**35**	**18**	**25**
1-gram için toplam		**6**	**11**	**4**	**9**
2-gram için toplam		**3**	**12**	**7**	**8**
1..2-gram için toplam		**3**	**12**	**7**	**8**

Çizelge 4.4'teki derecelendirme sonuçlarında toplam değeri en düşük olan algoritma en başarılıdır. Bu deney verisetlerinde en başarılı sonuçların M-NB algoritması ile elde edildiği görülmüştür. BKO yaklaşımı M-NB algoritmasına olumsuz etki gösterirken DVM algoritmasına olumlu etki göstermektedir.

4.3.2 Farklı içerik ile eğitilen verisetlerinin test sonuçları

Farklı içerikli eğitim verilerine sahip olan 3, 4 ve 5. deney verisetleri için M-NB ve DVM algoritmalarında kelime vektörü seçiminin ve BKO yaklaşımının kullanıldığı sınıflandırma sonuçlarının doğruluk ve f-değerleri Çizelge 4.5'te gösterilmektedir.

Çizelge 4.5 : Seçilen özelliklere göre kelime n-gram özellik vektörü seçimi ve bilgi kazanım oranının sınıflandırma başarısına etkisi.

Veriseti Özellikleri	Veriseti	Algoritma	1-gram		2-gram		1..2-gram	
			Doğruluk	F ölçümü	Doğruluk	F ölçümü	Doğruluk	F ölçümü
DMOZ	3	M-NB	**73,15%** σ = 0,004	0,719	73,12% σ = 0,004	0,719	69,79% σ = 0,005	0,663
		DVM	65,18% σ = 0,438	0,673	**67,42%** σ = 0,603	0,676	66,38% σ = 0,398	0,684
		BKO + M-NB	73,15% σ = 0,004	0,719	63,91% σ = 0,008	0,624	**73,62%** σ = 0,003	0,721
		BKO + DVM	65,19% σ = 0,417	0,673	56,28% σ = 1,000	0,559	**66,34%** σ = 0,477	0,678
Site İçeriği	4	M-NB	74,29% σ = 0,006	0,743	**76,51%** σ = 0,002	0,758	76,35% σ = 0,004	0,760
		DVM	68,68% σ = 0,669	0,676	65,79% σ = 0,819	0,644	**68,83%** σ = 0,006	0,678
		BKO + M-NB	**73,65%** σ = 0,005	0,739	67,62% σ = 0,008	0,674	69,06% σ = 0,009	0,690
		BKO + DVM	**68,76%** σ = 0,547	0,676	63,32% σ = 0,619	0,616	65,64% σ = 0,544	0,642
DMOZ + Site İçeriği	5	M-NB	74,40% σ = 0,006	0,744	**76,84%** σ = 0,004	0,762	76,52% σ = 0,004	0,763
		DVM	68,50% σ = 0,595	0,673	65,95% σ = 1,881	0,645	**68,62%** σ = 0,736	0,675
		BKO + M-NB	**73,78%** σ = 0,005	0,740	67,94% σ = 0,009	0,677	71,44% σ = 0,010	0,713
		BKO + DVM	**68,92%** σ = 0,499	0,678	62,67% σ = 0,551	0,609	64,96% σ = 0,319	0,634

Çizelge 4.5 incelendiğinde eğitimde kullanılan sayfa içeriklerinin sadece DMOZ verileri ile eğitilen modele göre sınıflandırma başarısına daha fazla katkı sağladığı görülmüştür. Sayfa içerikleri ile eğitilen sisteme DMOZ verileri de eklendiğinde sınıflandırma performansına olumlu etki gözlense de anlamlı bir fark ortaya çıkmamıştır.

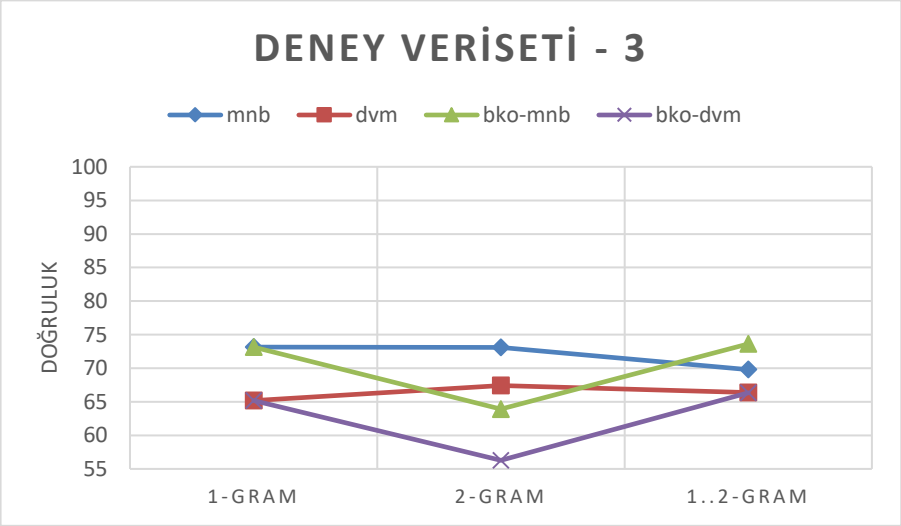
Çizelge 4.5'teki test sonuçlarına göre BKO yaklaşımının sınıflandırma başarısına katkısı incelendiğinde aşağıdaki sonuçlar görülmektedir.

Kelime 1-gram için doğruluk ve f-değeri sonuçlarına bakıldığında 3, 4 ve 5. deney verisetlerinde BKO yaklaşımının M-NB algoritmasının performansına ciddi derecede bir katkısı gözlenmemiştir. Kelime 1-gram için BKO yaklaşımının DVM algoritmasına katkısı incelendiğinde 14 kategoriden oluşan deney verisetlerinde doğruluk ve f-değerlerine bakıldığında ciddi bir katkı sağlamadığı görülmektedir.

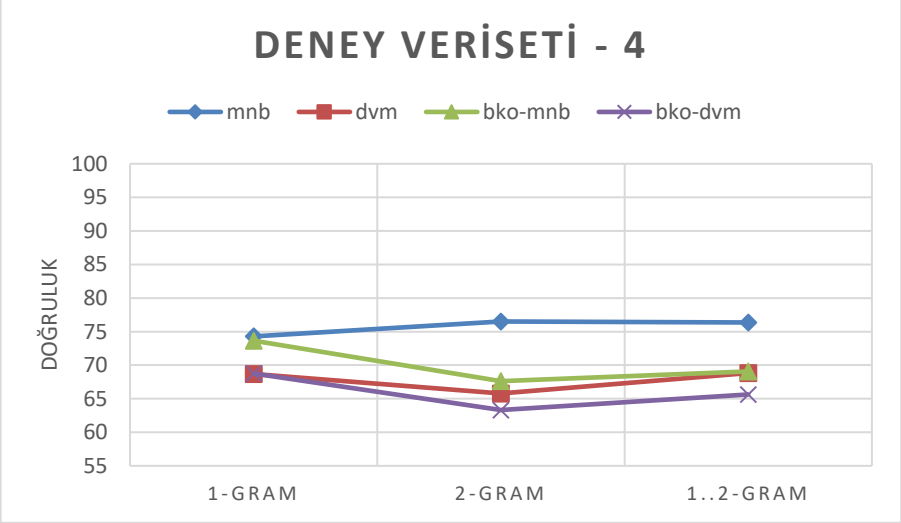
BKO yaklaşımının aynı deney verisetlerinde 2-gram özellik vektörü için M-NB algoritmasının performansına etkisinin %9 civarında olumsuz olduğu gözlenmiştir. DVM algoritmasında da BKO yaklaşımının performansa olumsuz etkisi gözlenmiştir.

BKO yaklaşımının 1..2-gram için doğruluk ve f-değerleri 3, 4 ve 5. deney verisetleri için incelendiğinde M-NB ve DVM algoritmalarının performansına olumlu yönde katkı sağlamadığı gözlenmiştir.

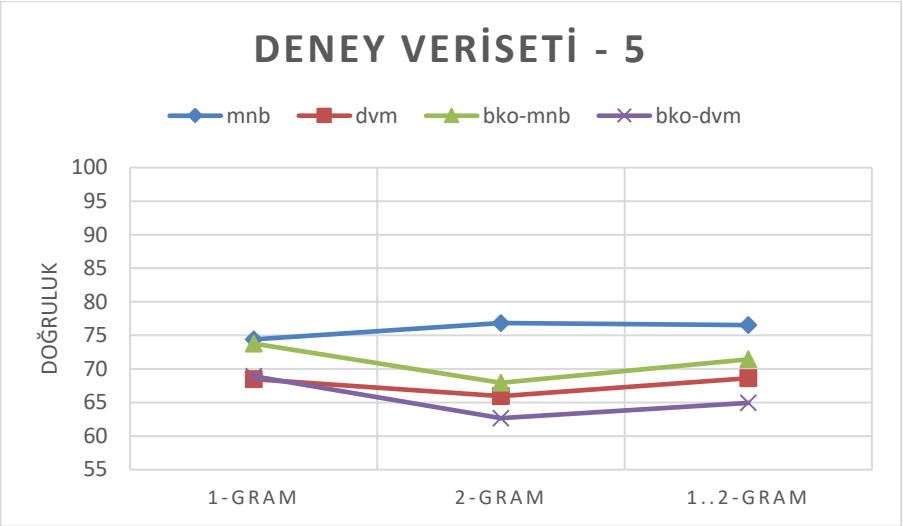
Kelime vektörü seçiminin uygulandığı 3, 4 ve 5. deney verisetlerinde M-NB, DVM, BKO+M-NB ve BKO+DVM algoritmalarının doğruluk değerleri değişimleri Şekil 4.6, Şekil 4.7 ve Şekil 4.8'de gösterilmiştir.



Şekil 4.6 : 3. deney verisetinin kelime vektörü seçimine göre algoritmaların doğruluk değerleri grafiği.



Şekil 4.7 : 4. deney verisetinin kelime vektörü seçimine göre algoritmaların doğruluk değerleri grafiği.



Şekil 4.8 : 5. deney verisetinin kelime vektörü seçimine göre algoritmaların doğruluk değerleri grafiği.

BKO yaklaşımının incelenen tüm n-gram kelime vektörleri için 3, 4 ve 5. deney verisetlerinde M-NB ve DVM algoritmalarının performansında ciddi bir değişim görülmediğinden bu başlıkta özellik vektörü değerlendirmelerinde BKO yaklaşımlı algoritmalar incelenmemiştir. Çizelge 4.5'teki test sonuçlarına göre kelime n-gram özellik vektörü seçiminin sınıflandırma başarısına etkisi incelendiğinde aşağıdaki sonuçlar görülmektedir.

M-NB algoritmasının 3, 4 ve 5. deney verisetlerinde 1-gram referans alındığında 2-gram özellik vektörü seçiminin daha iyi sonuçlar verdiği gözlenmiştir. Aynı deney verisetlerinde DVM algoritmasının kelime 1-gram özellik vektörü seçiminin daha iyi sonuçlar verdiği görülmektedir. 1..2-gram özellik vektörünün doğruluk ve f-değerlerindeki yükselişte 1-gram'ın olumlu etkisi olduğu gözlenmiştir.

Uygulanan algoritmaların 3, 4 ve 5. deney verisetlerindeki sonuçlarının derecelendirilmesi Çizelge 4.6'da gösterilmiştir. Çizelge üzerinde en başarılı algoritma 1'den başlayarak derecelendirilmiştir. 3. deney verisetinde 1-gram özellik vektöründe M-NB ile BKO+M-NB algoritmalarının doğruluk ve f-değerleri eşit olduğundan dolayı iki algoritmanın derecesi aynıdır.

Çizelge 4.6 : 3, 4 ve 5. deney verisetleri için algoritma sonuçlarının derecelendirmesi.

Veriseti	Özellik Vektörü	Algoritmalar			
		M-NB	DVM	BKO+M-NB	BKO+DVM
3	1-gram	1	3	1	2
	2-gram	1	2	3	4
	1..2-gram	2	3	1	4
4	1-gram	1	4	2	3
	2-gram	1	3	2	4
	1..2-gram	1	3	2	4
5	1-gram	1	4	2	3
	2-gram	1	3	2	4
	1..2-gram	1	3	2	4
Toplam		**10**	**28**	**17**	**32**
1-gram için toplam		**3**	**11**	**5**	**8**
2-gram için toplam		**3**	**8**	**7**	**12**
1..2-gram için toplam		**4**	**9**	**5**	**12**

Çizelge 4.6'daki derecelendirme sonuçlarında toplam değeri en düşük olan algoritma en başarılıdır. Bu deney verisetlerinde en başarılı sonuçların M-NB algoritması ile elde edildiği görülmüştür. BKO yaklaşımının M-NB ve DVM algoritmalarının performansına ciddi bir etkisi gözlenmemiştir.

4.3.3 İçerik kalitesi farklı deney verisetleri için test sonuçları

İçerik kalitesi açısından farklı olan 5 ve 6. deney verisetleri için M-NB ve DVM algoritmalarında kelime vektörü seçiminin ve BKO yaklaşımının kullanıldığı sınıflandırma sonuçlarının doğruluk ve f-değerleri Çizelge 4.7'de gösterilmektedir.

Çizelge 4.7 : Verisetindeki eksik özelliklerin, kelime n-gram özellik vektörü seçimi ve bilgi kazanım oranının sınıflandırma başarısına etkisi.

Veriseti Özellikleri	Örnek Sayısı	Veriseti	Algoritma	1-gram		2-gram		1..2-gram	
				Doğruluk	F ölçümü	Doğruluk	F ölçümü	Doğruluk	F ölçümü
Eksik içerikli sayfalar dahil edildi,	**22,347**	**5**	**M-NB**	74,40% σ = 0,006	0,744	**76,84%** σ = 0,004	0,762	76,52% σ = 0,004	0,763
			DVM	68,50% σ = 0,595	0,673	65,95% σ = 1,881	0,645	**68,62%** σ = 0,736	0,675
			BKO + M-NB	**73,78%** σ = 0,005	0,740	67,94% σ = 0,009	0,677	71,44% σ = 0,010	0,713
			BKO + DVM	**68,92%** σ = 0,499	0,678	62,67% σ = 0,551	0,609	64,96% σ = 0,319	0,634
Eksik içerikli sayfalar çıkarıldı,	**10,241**	**6**	**M-NB**	77,07% σ = 0,009	0,770	**79,45%** σ = 0,007	0,786	78,72% σ = 0,008	0,778
			DVM	**70,31%** σ = 0,789	0,693	65,36% σ = 0,761	0,629	69,06% σ = 0,531	0,672
			BKO + M-NB	76,79% σ = 0,007	0,770	76,67% σ = 0,004	0,762	**77,90%** σ = 0,006	0,781
			BKO + DVM	69,79% σ = 0,865	0,688	66,77% σ = 0,695	0,652	**72,04%** σ = 0,931	0,711

Çizelge 4.7 incelendiğinde sayfa içeriklerinden elde edilen site başlık, meta açıklama ve meta kelimeleri eksik olmayan sayfalarla çalışıldığında sınıflandırma başarısının olumlu şekilde arttığı gözlenmiştir. Sayfa içeriklerinden alınan site başlık, meta açıklama ve meta kelimelerin sayfaların birbirinden ayırt edilmesinde olumlu etki sağladıkları görülmektedir.

Çizelge 4.7'deki test sonuçlarına göre BKO yaklaşımının sınıflandırma başarısına katkısı incelendiğinde aşağıdaki sonuçlar görülmektedir. 5 ve 6. deney verisetlerinin kelime vektörü seçimine göre algoritmaların doğruluk değerleri Şekil 4.5 ve Şekil 4.8'de gösterilmiştir.

BKO yaklaşımının 5 ve 6. deney verisetlerinde kelime 1-gram için doğruluk ve f-değerlerine bakıldığında M-NB ve DVM algoritmalarının performansına ciddi bir etkisi olmadığı gözlenmiştir.

2-gram özellik vektörü için BKO yaklaşımının eksik içerikli sayfaların dahil olduğu 5. deney verisetinde M-NB algoritmasının performansında yaklaşık %9 oranında olumsuz etkisi görülürken eksik içeriklerin dahil edilmediği 6. deney verisetinde de yaklaşık %3 oranında olumsuz etkisi görülmektedir. Aynı özellik vektörü için eksik içerikli sayfaların dahil olduğu verisetinde BKO yaklaşımının DVM algoritmasının performansında yaklaşık %3 oranında olumsuz etki gösterirken eksik içerikli sayfaların dahil edilmediği verisetinde %1 oranında olumlu etkisi görülmektedir.

1..2-gram özellik vektöründe BKO yaklaşımının eksik içeriğe sahip olan sayfaların da bulunduğu 5. deney verisetinde M-NB algoritmasının performansına olumsuz etki gösterirken, eksik içeriklerin dahil edilmediği 6. deney verisetinde M-NB algoritmasına ciddi bir katkısı görülmemiştir. Aynı özellik vektöründe BKO yaklaşımının eksik içeriğe sahip olan sayfaların dahil edildiği verisetinde DVM algoritmasının performansına olumsuz etki gösterirken eksik içeriklerin dahil edilmediği verisetinde DVM algoritmasının performansına %3 oranında olumlu katkı sağladığı gözlenmiştir.

Çizelge 4.7'deki test sonuçlarına göre kelime n-gram özellik vektörü seçiminin sınıflandırma başarısına etkisi incelendiğinde aşağıdaki sonuçlar görülmektedir.

M-NB algoritmasının 5 ve 6. deney verisetlerinde 1-gram özellik vektörü referans alındığında doğruluk ve f-değerlerine bakıldığında 2-gram özellik vektörünün

yaklaşık %2 oranında olumlu etkisi gözlenmiştir. M-NB algoritmasının 5 ve 6. deney verisetlerinde en yüksek başarısının 2-gram özellik vektöründe elde edildiği görülmüştür. BKO yaklaşımlı M-NB algoritmasının 1-gram özellik vektöründe daha iyi sonuçlar verdiği gözlenmiştir.

DVM algoritmasının 5 ve 6. deney verisetlerinde 2-gram ve 1..2-gram özellik vektörlerinin olumlu bir katkı sağlamadığı görülmektedir. DVM algoritmasının aynı deney verisetlerinde 1-gram özellik vektöründe daha başarılı sonuçlar verdiği ve BKO yaklaşımlı DVM algoritmasının 1-gram ve 1..2-gram özellik vektörü seçimlerinde daha iyi sonuçlar verdiği görülmektedir.

Uygulanan algoritmaların 5 ve 6. deney verisetlerindeki sonuçlarının derecelendirilmesi Çizelge 4.8'de gösterilmiştir. Çizelge üzerinde en başarılı algoritma 1'den başlayarak derecelendirilmiştir.

Çizelge 4.8 : 5 ve 6. deney verisetleri için algoritma sonuçlarının derecelendirmesi.

Veriseti	**Özellik Vektörü**	**Algoritmalar**			
		M-NB	**DVM**	**BKO+M-NB**	**BKO+DVM**
5	1-gram	1	4	2	3
	2-gram	1	3	2	4
	1..2-gram	1	3	2	4
6	1-gram	1	3	2	4
	2-gram	1	4	2	3
	1..2-gram	1	4	2	3
Toplam		**6**	**21**	**12**	**21**
1-gram için toplam		**2**	**7**	**4**	**7**
2-gram için toplam		**2**	**7**	**4**	**7**
1..2-gram için toplam		**2**	**7**	**4**	**7**

Çizelge 4.8'deki derecelendirme sonuçlarında toplam değeri en düşük olan algoritma en başarılıdır. Bu deney verisetlerinde en başarılı sonuçların M-NB algoritması ile elde edildiği görülmektedir. BKO yaklaşımının M-NB ve DVM algoritmalarının performansına ciddi bir etkisi gözlenmemiştir.

4.4 Tüm Verisetlerin Test Sonuçları

Tüm deney verisetlerinde M-NB ve DVM algoritmaları ile yapılan sınıflandırmaya kelime n-gram özellik vektörü ve BKO yaklaşımının etkisi Çizelge 4.9'da gösterilmiştir.

Çizelge 4.9 : Bilgi kazanım oranı yaklaşımı ve kelime n-gram özellik vektörü seçimlerinin tüm deney verisetleri için M-NB ve DVM algoritmalarına etkisi.

Veriseti	Algoritma	1-gram		2-gram		1..2-gram	
		Doğruluk	F ölçümü	Doğruluk	F ölçümü	Doğruluk	F ölçümü
1	M-NB	89.98% σ = 0,009	0,902	**92.61%** σ = 0,006	0,926	91.97% σ = 0,009	0,921
	DVM	**85.37%** σ = 0,201	0,849	80.26% σ = 1,487	0,788	83.87% σ = 1,103	0,830
	BKO + M-NB	90.04% σ = 0,010	0,903	**91.20%** σ = 0,010	0,913	90.83% σ = 0,010	0,910
	BKO + DVM	**86.94%** σ = 3,289	0,868	85.45% σ = 5,513	0,851	86.92% σ = 4,257	0,866
2	M-NB	89.80% σ = 0,008	0,900	**92.39%** σ = 0,009	0,924	91.60% σ = 0,008	0,917
	DVM	**85.18%** σ = 0,802	0,847	79.71% σ = 1,325	0,781	83.14% σ = 0,744	0,821
	BKO + M-NB	89.94% σ = 0,007	0,901	**90.05%** σ = 0,023	0,902	89.91% σ = 0,018	0,901
	BKO + DVM	89.83% σ = 0,691	0,900	**90.89%** σ = 1,060	0,910	90.59% σ = 0,907	0,910
3	M-NB	**73.15%** σ = 0,004	0,719	73.12% σ = 0,004	0,719	69.79% σ = 0,005	0,663
	DVM	65.18% σ = 0,438	0,673	**67.42%** σ = 0,603	0,676	66.38% σ = 0,398	0,684
	BKO + M-NB	73.15% σ = 0,004	0,719	63.91% σ = 0,008	0,624	**73.62%** σ = 0,003	0,721
	BKO + DVM	65.19% σ = 0,417	0,673	56.28% σ = 1,000	0,559	**66.34%** σ = 0,477	0,678
4	M-NB	74.29% σ = 0,006	0,743	**76.51%** σ = 0,002	0,758	76.35% σ = 0,004	0,760
	DVM	68.68% σ = 0,669	0,676	65.79% σ = 0,819	0,644	**68.83%** σ = 0,006	0,678
	BKO + M-NB	**73.65%** σ = 0,005	0,739	67.62% σ = 0,008	0,674	69.06% σ = 0,009	0,690
	BKO + DVM	**68.76%** σ = 0,547	0,676	63.32% σ = 0,619	0,616	65.64% σ = 0,544	0,642
5	M-NB	74.40% σ = 0,006	0,744	**76.84%** σ = 0,004	0,762	76.52% σ = 0,004	0,763
	DVM	68.50% σ = 0,595	0,673	65.95% σ = 1,881	0,645	**68.62%** σ = 0,736	0,675
	BKO + M-NB	**73.78%** σ = 0,005	0,740	67.94% σ = 0,009	0,677	71.44% σ = 0,010	0,713
	BKO + DVM	**68.92%** σ = 0,499	0,678	62.67% σ = 0,551	0,609	64.96% σ = 0,319	0,634
6	M-NB	77.07% σ = 0,009	0,770	**79.45%** σ = 0,007	0,786	78.72% σ = 0,008	0,778
	DVM	**70.31%** σ = 0,789	0,693	65.36% σ = 0,761	0,629	69.06% σ = 0,531	0,672
	BKO + M-NB	76.79% σ = 0,007	0,770	76.67% σ = 0,004	0,762	**77.90%** σ = 0,006	0,781
	BKO + DVM	69.79% σ = 0,865	0,688	66.77% σ = 0,695	0,652	**72.04%** σ = 0,931	0,711

Uygulanan algoritmaların tüm deney verisetlerindeki sonuçlarının derecelendirilmesi Çizelge 4.10'da gösterilmiştir.

Tüm deney verisetlerinde en iyi sonuçların M-NB algoritması ile elde edildiği görülmektedir. BKO yaklaşımının M-NB algoritmasının performansına ciddi derecede katkı sağlamazken DVM algoritmasına çoğunlukla olumlu yönde katkı sağladığı gözlenmiştir.

Çizelge 4.10 : Tüm deney verisetleri için algoritma sonuçlarının derecelendirmesi.

Veriseti	**Özellik Vektörü**	**Algoritmalar**			
		M-NB	**DVM**	**BKO+M-NB**	**BKO+DVM**
1	1-gram	2	4	1	3
	2-gram	1	4	2	3
	1..2-gram	1	4	2	3
2	1-gram	3	4	1	2
	2-gram	1	4	3	2
	1..2-gram	1	4	3	2
3	1-gram	1	3	1	2
	2-gram	1	2	3	4
	1..2-gram	2	3	1	4
4	1-gram	1	4	2	3
	2-gram	1	3	2	4
	1..2-gram	1	3	2	4
5	1-gram	1	4	2	3
	2-gram	1	3	2	4
	1..2-gram	1	3	2	4
6	1-gram	1	3	2	4
	2-gram	1	4	2	3
	1..2-gram	1	4	2	3
Toplam		**22**	**63**	**35**	**57**
1-gram için toplam		**9**	**22**	**9**	**17**
2-gram için toplam		**6**	**20**	**14**	**20**
1..2-gram için toplam		**7**	**21**	**12**	**20**

5. SONUÇ VE ÖNERİLER

Metin sınıflandırma işlemlerinde kategori sayısı az olduğunda daha yüksek doğruluk oranları gözlenmiştir. Kategori sayısının arttığı durumlarda doğruluk oranlarında düşme görülmektedir. M-NB algoritmasının kategori değişiminden az etkilendiği ve yüksek doğruluk oranlarına sahip olduğu gözlenmiştir.

M-NB'nin, az sınıflı verisetlerindeki en yüksek doğruluk oranı %92,61 iken çok sınıflı verisetlerindeki en yüksek doğruluk oranı %79,45'tir. Her bir sınıf için kelime ağırlıkları otomatik bir şekilde hesaplanmıştır ve verisetlerine herhangi bir müdahalede bulunulmamıştır.

M-NB algoritmasının performansına en iyi katkıyı kelime 2-gram özellik vektörü seçiminin sağladığı görülmektedir. BKO yaklaşımının M-NB algoritmasına önemli bir katkı sağlamadığı görülmüştür.

DVM'de kelime n-gram özellik vektörü seçiminin önemli bir etkisi görülmediğinden 1-gram kelime vektörü kullanılarak BKO yaklaşımının, algoritmanın performansına daha fazla katkı sağladığı görülmüştür. Ayrıca BKO yaklaşımı sınıf sayısı az olan deney verisetlerinde başarıyı artırmada daha fazla katkı sağladığı gözlenmiştir.

Sayfa içerikleriyle eğitilen sisteme DMOZ editörlerinin yaptığı açıklamalar eklendiğinde olumlu katkı sağlasa da anlamlı bir fark ortaya çıkmamıştır. Sayfa başlığı, meta anahtar kelime ve açıklamaları eksiksiz olan sayfalar ile çalışıldığında örnek sayısı yarı yarıya azalsa da en yüksek doğruluk oranına ulaşıldığı görülmektedir. Dolayısıyla iyi bir model oluşturmak için çok sayıda örnek yerine içerik kalitesinin artırılması sınıflandırma başarısını olumlu yönde etkilemektedir.

Çok sınıflı metin sınıflandırma işlemlerinde yüksek doğruluk oranlarına ulaşabilmek için mümkünse birbirine yakın kategoriler birleştirilerek kategori sayısı azaltılabilir, özellikle eğitim için kullanılan web sayfalarındaki meta bilgi ve başlıkların eksiksiz olmasına dikkat edilebilir. Çok sınıflı metin sınıflandırma işlemlerinde M-NB

algoritmasının kullanılmasıyla performans, kullanım kolaylığı ve zaman açısından avantaj sağlanmaktadır.

Köke indirgeme işlemi sırasında karşılaşılan hatalı durumların önlenmesi için metnin anlamına da dikkat edilmesiyle en uygun kökün bulunarak sınıflandırma başarısının yükseltilmesi ve hiyerarşik yapıda bir derlem oluşturulması ileride yapılması düşünülen çalışmalar arasındadır.

KAYNAKÇA

Aggarwal, C. C., & Zhai, C. (2012). Mining Text Data. *A survey of text classification algorithms* (s. 164-165). içinde Springer US.

Amasyalı, M., & Beken, A. (2009). Türkçe Kelimelerin Anlamsal Benzerliklerinin Ölçülmesi Ve Metin Sınıflandırmada Kullanılması. *IEEE signal processing and communications applications conference, SİU.* Antalya.

Baig, Z. A., Shaheen, A. S., & AbdelAal, R. (2011). One-Dependence Estimators for Accurate Detection of Anomalous Network Traffic. *International Journal for Information Security Research, 1*(4), 202-210.

Baykal, A. (2006). Veri Madenciliği Uygulama Alanları. *DÜ Ziya Gökalp Eğitim Fakültesi Dergisi*(7), 95-107.

Beckett, D. (2014). *RDF 1.1 XML Syntax.* W3C Recommendation.

Bermejo, P., G´amez, J. A., & Puerta, J. M. (2010). *Improving the performance of Naive Bayes Multinomial in e-mail foldering by introducing distribution-based balance of datasets.* Albacete: Elsevier.

Değerli, O. (2012). *Naıve Bayes Yöntemi İle Blog İçeriklerinin Sınıflandırılması.* Ankara: Gazi Üniversitesi.

Dhillon, I. S., Mallela, S., & Kumar, R. (2002). *Enhanced Word Clustering for Hierarchical Text Classification.* Austin.

Dhillon, I. S., Mallela, S., & Kumar, R. (2003). A Divisive Information-Theoretic Feature Clustering Algorithm for Text Classification. *Journal of Machine Learning Research*, 1270-1271.

DMOZ. (2017, Nisan 15). http://www.dmoz.org/ adresinden alındı

Gürcan, F. (2009). *Web İçerik Madenciliği Ve Konu Sınıflandırılması.* Trabzon: Karadeniz Teknik Üniversitesi.

Kaliyeva, S. (2013). *Bilimsel Makalelerin Metin İşleme Yöntemleri İle Sınıflandırılması.* Ankara: Gazi Üniversitesi.

Kaşıkçı, T., & Gökçen, H. (2014). Metin Madenciliği ile E-Ticaret Sitelerinin Belirlenmesi. *Bilişim Teknolojileri Dergisi*, 25-32.

Kavzoğlu, T., & Çölkesen, İ. (2010). Destek Vektör Makineleri ile Uydu Görüntülerinin Sınıflandırılmasında Kernel Fonksiyonlarının Etkilerinin İncelenmesi. *Harita Dergisi*(144), 73-82.

Klyne, G., Carroll, J. J., & McBride, B. (2014). *RDF 1.1 Concepts and Abstract Syntax.* W3C Recommendation.

Kolyiğit, Ö. (2013). *Türkçe Dokümanlar İçin Yazar Tanıma.* Aydın: Adnan Menderes Üniversitesi.

Li, L., Wu, Y., & Ye, M. (2015). Experimental Comparisons of Multi-class Classifiers. *Informatica, 39*(1), 71-85.

Liu, B. (2011). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data.* Springer.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 1*, 142-150.

Mantaras, R. L. (1991). A Distance-Based Attribute Selection Measure for Decision Tree Induction. *Machine Learning, 6*(1), 81-92.

McCallum, A., & Nigam, K. (1998). *A Comparison of Event Models for Naive Bayes Text Classification.* Pittsburg.

Pilavcılar, İ. F. (2007). *Metin Madenciliği ile Metin Sınıflandırma.* İstanbul: Yıldız Teknik Üniversitesi.

Resource-Zone. (2017, Nisan 15). https://www.resource-zone.com/forum/t/dmoz-closure.53420/ adresinden alındı

Rigutini, L., & Maggini, M. (2004). *Automatic text processing: Machine learning techniques.* University of Siena.

Sandhya, M., Sarika, S., Anukriti, S., & Sushila, A. (2016). Automatic Text Categorization on News Articles. *International Journal of Engineering and Techniques, 2*(3), 33-38.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR), 34*(1), 1-47.

Silva, L. C. (2014). *Support Vector Machines.* Scholastic Tutors: http://scholastictutors.webs.com/Scholastic-Book-SupportVectorM-Part01-2014-01-26.pdf adresinden alındı

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management, 45*(4), 427-437.

Şekerci Hüsem, S. (2017, 06 23). GitHub: https://github.com/secilhusem/turkce-veriseti adresinden alındı

Tarımcı, A. B. (2009). *A Multithreaded Web Crawler and Text Search Engine.* İstanbul: Doğuş Üniversitesi.

Tunalı, V., & Bilgin, T. T. (2012). PRETO: A high-performance text mining tool for preprocessing Turkish texts. *Proceedings of the 13th International Conference on Computer Systems and Technologies*.

Yılmaz, R. (2013). *Türkçe Dokümanların Sınıflandırılması.* Aydın: Adnan Menderes Üniversitesi.

Yussouf Nahayo, S. A. (2016). Performance of SVM, k-NN and NBC classifiers for Text-Independent Speaker Identification With and Without Modelling Through Merging Models. *SAÜ Fen Bil Der, 20*(1), 1-6.

Zhou, B., & Yao, Y. (2010). Evaluating information retrieval system performance based on user preference. *Journal of Intelligent Information Systems,* 227-248.

EKLER

EK A: Çalışmada kullanılan Türkçe etkisiz kelimeler

EK B: Çalışmada kullanılan WEKA sürümü ve yapılan ayarlar

EK A: Çalışmada kullanılan Türkçe etkisiz kelimeler

acaba
altı
ama
ancak
artık
asla
aslında
az
bana
bazen
bazı
bazıları
bazısı
belki
ben
beni
benim
beş
bile
bir
birçoğu
birçok
birçokları
biri
birisi
birkaç
birkaçı
birşey
birşeyi
biz
bize
bizi
bizim
böyle
böylece
bu
buna
bunda
bundan
bunu
bunun
burada
bütün
çoğu
çoğuna
çoğunu
çok
çünkü
da
daha
de
değil
demek
diğer
diğeri
diğerleri
diye
dokuz
dolayı
dört
elbette
en
fakat
falan
felan
filan
gene
gibi
hâlâ
hangi
hangisi
hani
hatta
hem
henüz
hep
hepsi
hepsine
hepsini
her
her biri
herkes
herkese
herkesi
hiç
hiç kimse
hiçbiri
hiçbirine
hiçbirini
için
içinde
iki
ile
ise
işte
kaç
kadar
kendi
kendine
kendini
ki
kim
kime
kimi
kimin
kimisi
madem
mı
mi
mu
mü
mü
nasıl
ne
ne kadar
ne zaman
neden
nedir
nerde
nerede
nereden
nereye
nesi
neyse
niçin
niye
on
ona
ondan
onlar
onlara
onlardan
onların
onların
onu
onun
orada
oysa
oysaki
öbürü
ön
önce
ötürü
öyle
rağmen
sana
sekiz
sen
senden
seni
senin
siz
sizden
size
sizi
sizin
son
sonra
şayet
şey
şeyden
şeye
şeyi
şeyler
şimdi
şöyle
şu
şuna
şunda
şundan
şunlar
şunu
şunun
tabi
tamam
tüm
tümü
üç
üzere
var
ve
veya
veyahut
ya
ya da
yani
yedi
yerine
yine
yoksa
zaten
zira

EK B: Çalışmada kullanılan WEKA sürümü ve yapılan ayarlar

ALGORİTMA	CLASSIFIER	FILTER			
M-NB	NaiveBayesMultinomial	StringtoWordVector			
		wordsToKeep=10.000			
		tokenizer: NGramTokenizer			
		NGramMaxSize	1	2	2
		NGramMinSize	1	2	1
DVM	SMO	StringtoWordVector			
		wordsToKeep=10.000			
		tokenizer: NGramTokenizer			
		NGramMaxSize	1	2	2
		NGramMinSize	1	2	1
BKO + M-NB	NaiveBayesMultinomial	StringtoWordVector			
		wordsToKeep=10.000			
		tokenizer: NGramTokenizer			
	evaluator: GainRatioAttributeEval	NGramMaxSize	1	2	2
		NGramMinSize	1	2	1
BKO + DVM	SMO	StringtoWordVector			
		wordsToKeep=10.000			
		tokenizer: NGramTokenizer			
	evaluator: GainRatioAttributeEval	NGramMaxSize	1	2	2
		NGramMinSize	1	2	1
WEKA 3.8.1 sürümü kullanılmıştır. Örneğin 1-gram kelime vektörü için NGramMinSize=1 ve NGramMaxSize=1 seçenekleri kullanılmıştır.					

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER:

Ad-Soyad	: Seçil ŞEKERCİ HÜSEM
Doğum Yılı ve Yeri	: 1985 - ANKARA
E-posta	: secilsekerci@hotmail.com

İŞ DENEYİMİ:

Aralık 2010 tarihinden itibaren Milli Eğitim Bakanlığına bağlı Bilişim Teknolojileri Öğretmeni olarak görev yapmaktayım.

ÖĞRENİM DURUMU:

Lisans	: 2009, Gazi Üniversitesi, Endüstriyel Sanatlar Eğitim Fakültesi, Bilgisayar Öğretmenliği
Lisans (Mühendislik tamamlama)	: 2015, Namık Kemal Üniversitesi, Çorlu Mühendislik Fakültesi, Bilgisayar Mühendisliği