Metin Sınıflandırma
Text Classification

A. Cüneyd TANTUĞ

İTÜ Bilgisayar ve Bilişim Fakültesi
tantug@itu.edu.tr

Özetçe

Geçtiğimiz yirmi yıl göz önüne alındığında, bilgisayar ortamında üretilen belgelerin sayısının her geçen sene yükselen bir ivme ile artmakta olduğu görülmektedir. Kuşkusuz internetin ortaya çıkması, geniş kitleler tarafından kolayca ulaşılabilir ve kullanılabilir hâle gelmesi, kişisel bilgisayarların, akıllı telefonların, tabletlerin fiyatlarının ucuzlaması ve toplumun bilgisayar kullanımının artması gibi etkenler, elektronik ortamda oluşturulan belgelerin sayısının artmasının en önemli nedenleridir. Çok sayıda bilgi varlığının getirdiği sayısız fayda ile beraber ortaya çıkan bazı sorunların da çözülmesi gerekmektedir. Bu bağlamda ortaya çıkan sorunlardan bir tanesi de elektronik ortamdaki metinlerin sınıflandırılması sorunudur. Metin sınıflandırma sorunu, en genel anlamı ile eldeki bir metnin önceden belirlenen sınıflardan hangisine ya da hangilerine girdiğinin belirlenmesi demektir. Metin sınıflandırma için belge sınıflandırma, metin kategorilerinin belirlenmesi gibi farklı isimler de kullanılmaktadır.

Anahtar Sözcükler

metin sınıflandırma, belge sınıflandırma, metin kategorilerinin belirlenmesi

Abstract

The last two decades witnesses the proliferation of the number of electronically accessible documents. The emerge of the internet, easy access and usage of the internet with increasing coverage ratios, increased computer, smart phone and tablet usage triggered by decreased prices, increased computer usage ratios in the society and similar factors played important roles in this. Having lots of documents has some advantages, as well as some disadvantages that must be dealt with. A problem to work with the large number of documents is categorizing texts or electronic documents. In its broad sense, text categorization problem aims to determine which previously specified category or categories are suitable for a given text. This problem is addressed by different terms like text classification, document classification and document categorization in various contexts.

Keywords

text classification, document classification, text categorization,

1 Giriş

Geçtiğimiz yirmi yıl göz önüne alındığında, bilgisayar ortamında üretilen belgelerin sayısının her geçen sene yükselen bir ivme ile artmakta olduğu görülmektedir. Kuşkusuz internetin ortaya çıkması, geniş kitleler tarafından kolayca ulaşılabilir ve kullanılabilir hâle gelmesi, kişisel bilgisayarların fiyatlarının ucuzlaması ve toplumun bilgisayar kullanımının artması gibi etkenler, elektronik ortamda oluşturulan belgelerin sayısının artmasının en önemli nedenleridir. Çok sayıda bilgi varlığının getirdiği sayısız fayda ile beraber ortaya çıkan bazı sorunların da çözülmesi gerekmektedir. Bilgiye erişmenin (aranılan bilginin bulunabilirliğinin) kolay olması gereklidir. Bu bağlamda ortaya çıkan sorunlardan bir tanesi de elektronik ortamdaki metinlerin sınıflandırılması sorunudur. Metin sınıflandırma sorunu, en genel anlamı ile eldeki bir metnin önceden belirlenen sınıflardan hangisine ya da hangilerine girdiğinin belirlenmesi demektir. Metin sınıflandırma için belge sınıflandırma, metin kategorilerinin belirlenmesi gibi farklı isimler de kullanılmaktadır. İngilizcede ise bu sorun için sıklıkla "text classification", "document classification", "text categorization", "document categorization" gibi isimler kullanılmaktadır. Daha genel bir anlamı içerdiğinden bu bölümde metin sınıflandırma terimi tercih edilmiştir.

Metin sınıflandırmanın birçok farklı uygulama alanı vardır. Örneğin kütüphanede yeni gelen bir kitabın konusunun belirlenmesi ve benzer konulu kitaplar arasında uygun yerin belirlenmesi bir metin sınıflandırma sorunudur. Bu işlemin, insan emeği yerine bilgisayarla yapılması durumunda işleme, bilgisayarlı metin sınıflandırma adı verilmektedir. Yaramaz (spam) e-postaların süzülmesi, bir metnin yazarının ya da dilinin belirlenmesi, belge indeksleme, sözcük anlamının belirlenmesi gibi birçok uygulama metin sınıflandırma uygulamasına örnektir. Diğer yandan başka tür sınıflandırma uygulamaları da metin sınıflandırma çözüm yöntemleri ile gerçekleştirilebilir. Konuşmaların sınıflandırılması uygulamasında, konuşma tanıma işleminden sonra metin sınıflandırma yapılarak konuşmanın uygun sınıfa atanması sağlanabilir. Video gibi çoğul ortamların sınıflandırılması sorunu da, belgede çoğul ortamla ilgili metinlerin sınıflandırılması sorununa indirgenerek çözülmektedir.

Bilgisayarlı metin sınıflandırma uygulamalarının tarihçesi her ne kadar 1960'lara kadar gitse de uygulamaların yoğunluk kazanması ancak 1980'lerin sonu ve 1990'ların başında olmuştur. İlk yıllarda yapılan uygulamaların çoğunluğu, uzman sistemler (expert systems) yaklaşımını temel almaktaydı. Bu yıllarda, sınıflandırmanın nasıl yapılacağını bilen bir uzmanın aldığı kararların benzetiminin yapıldığı kural tabanlı bilgisayar sistemleri tasarlanıyordu. (bkz. Şekil 1).

$$\text{wheat} \;\&\; \text{farm} \longrightarrow \text{wheat}$$
$$\text{wheat} \;\&\; \text{commodity} \longrightarrow \text{wheat}$$
$$\text{bushels} \;\&\; \text{export} \longrightarrow \text{wheat}$$
$$\text{wheat} \;\&\; \text{agriculture} \longrightarrow \text{wheat}$$
$$\text{wheat} \;\&\; \text{tonnes} \longrightarrow \text{wheat}$$
$$\text{wheat} \;\&\; \text{winter} \;\&\; \neg\text{soft} \longrightarrow \text{wheat}$$

	Test Cases	
	wheat	not wheat
wheat	73	8
not wheat	14	3577

Şekil 1: Kural tabanlı sınıflandırma için bir kural örneği ve başarımı [1]

Ancak 1990'lı yıllardan itibaren her bilgi işlem kapasitesinin ve kaynakların artması ile geniş kullanım alanı bulan makine öğrenmesi (machine learning) teknikleri, her alanda olduğu gibi metin sınıflandırma alanında da baskın yöntem olarak öne çıkmıştır. Günümüzde güncel sistemlerin neredeyse tamamı makine öğrenmesi yöntemleri ile sınıflandırma yapmaktadır.

Bu bölümde metin sınıflandırmanın genel bir tanımı ve türleri sunulacak, uygulama alanları tanıtılacak, çözüm için kullanılan yöntemlere örnekler verilecek ve metin sınıflandırma sonuçlarının başarımının nasıl değerlendirileceğine ilişkin yöntemler anlatılacaktır. Ayrıca Türkçe için yapılan metin sınıflandırma çalışmalarına da bölümün sonunda yer verilmiştir.

Bu bölümde metin sınıflandırmaya ait genel özellikler verilmektedir. Uygulamalara özel bazı durumlar, yöntemler ve sonuçlar farklılık gösterebilmektedir. Örneğin bazı uygulamalarda metnin yayımlandığı tarih, yayımlayıcısı, belge türü gibi metin ile ilgili ek bilgiler bulunabilir ve bu bilgiler sınıflandırma sürecinde kullanılabilir.

Ancak, metin sınıflandırma açıklanırken yalnızca metin içeriğinin kullanılması, sorunu en genel hâli ile ifade etmektedir. Bu nedenle bölüm kapsamında sınıflandırmada işinde sadece metnin içeriğinden faydalanıldığı varsayılmaktadır.

2. Metin Sınıflandırma Tanımı

Metin sınıflandırma sorunu, $B=\{b_1,b_2,\ldots,b_n\}$ kümesindeki her bir belgenin (metin), önceden tanımlanmış $S=\{s_1,s_2,\ldots,s_m\}$ kümesindeki sınıflara ait olup olmadığının belirlenmesidir. Yani her $(b_j, s_i) \in B \times S$ çifti için doğru ya da yanlış biçiminde bir mantıksal değer üretilmesi gerekmektedir. Bir *g* fonksiyonun, gerçek sonuçları, yani *j.* belge, *i.* sınıfa ait ise doğru, değil ise yanlış değerlerini ürettiği kabul edilsin. Daha kurallı bir yazım ile $g : D \times C \rightarrow \{doğru, yanlış\}$ biçiminde ifade edilebilir. Bu durumda, benzer biçimde çalışacak bir f fonksiyonu $f : D \times C \rightarrow \{doğru, yanlış\}$ makine öğrenmesi yöntemi ile tasarlanabilir. Tasarlanan *f* fonksiyonun ürettiği sonuçların, mümkün olduğu kadar *g* fonksiyonun sonuçları ile aynı olması hedeflenmektedir. Makine öğrenmesi yöntemleri ile bir model kurularak *g* fonksiyonuna benzer çalışan *f* fonksiyonu (sınıflandırıcı (*classifier*)) gerçekleştirilir ve daha sonra bu *f* fonksiyonun gerçek sonuçlar ile yani *g* ile ne kadar benzeştiği ölçülür.

3. Metin Sınıflandırma Yöntemlerinin Özellikleri

Metin sınıflandırmada işleminde kullanılan yöntemler bu bölümde açıklanmıştır:

3.1 Tek Etiketli ve Çok Etiketli Sınıflandırma

Farklı uygulamalar geliştirilirken isterler farklı olmakta, bu da sınıflandırmaların değişik biçimlerde yapılması zorunluluğunu doğurmaktadır. Sınıflandırma sonuçlarının tek etiketli ya da çok etiketli olarak üretilmesi de bu zorunluluklardan bir tanesi olarak ortaya çıkmaktadır. Tek etiketli sınıflandırma (*single-label*), var olan *m* sınıftan her bir metin için ilgili olduğu belirlenen sadece bir tanesinin seçildiği ve belirlenen sınıfa ait etiketin metne "yapıştırıldığı" ya da "atandığı" uygulamadır. Çok etiketli sınıflandırma (*multi-label*) ise, *m* adet sınıftan bir metin için ilgili olduğu belirlenen bir ya da birden fazla etiket seçilir.

Birçok uygulamada bir metin için tek bir sınıfın belirlenmesi yeterli olurken bazı uygulamalarda bir metnin, birden fazla sınıfa girecek şekilde sınıflandırılması istenebilir. Örneğin bir ülke başbakanının bir konser katılımında yaptığı açıklamaları içeren bir haber metni hem "siyaset" sınıfında hem de "kültür sanat" sınıfında yer alabilir.

Tek etiketli yöntemler, her metin için sadece bir sınıfın atanmasını şart koşmaktadır. Diğer bir deyişle, sabit bir *j* değeri için (d_j, c_i) eşleşmelerinden sadece bir tanesi "*doğru*" değerini alabilecektir. Çok etiketli sınıflandırmalarda ise, belirli bir $d_j \in D$ belgesi için k adet $(0 \leq k \leq m)$ sınıf atanabilir. Tek etiketli yöntemlerin özel bir durumu ise ikili sınıflandırma (*binary classification*) olarak adlandırılır. İkili sınıflandırma, adından da anlaşılacağı gibi m=2 olan, yani sadece iki sınıf arasında tek etiketli atama yapılabilecek sınıflandırma işidir.

Teknik açıdan bakıldığında tek etiketli ya da çok etiketli bütün sınıflandırma işleri, sınıf sayısından bağımsız olarak ikili sınıflandırma ile çözülebilir. Yani her türlü metin sınıflandırma işi ikili sınıflandırma sorununa dönüştürülebilir. Örnek olarak haber metinleri *Siyaset*, *Spor*, *Magazin* başlıklı üç kategoride tek etiketli biçimde sınıflandırılacak olsun. Bu sorun 3 ikili sınıflandırma sorununa indirgenebilir: "*Siyaset*"-"*Siyaset Dışı*", "*Spor*"-"*Spor Dışı*", "*Magazin*"-"*Magazin Dışı*". Teorik açıdan, sınıfları $S=\{s_1,s_2,\ldots,s_m\}$ biçiminde belirlenen tek etiketli ya da çok etiketli her türlü sınıflandırma sorunu, $S_i = \{s_i, \overline{s_i}\}$ sınıflarını içeren *m* tane bağımsız ikili sınıflandırma işinin çözümü şeklinde ifade edilebilir. Dolayısı ile genel yöntemlerin anlatılmasında çoğunlukla ikili sınıflandırma sorununun çözümü üzerinde durulur. Bu alanda geliştirilen bir çözüm, ikiden çok sınıf içeren tek etiketli ya da çok etiketli sınıflandırma sorunları için de kolaylıkla genişletilebilmektedir.

3.2 Sınıf Odaklı ve Belge Odaklı Sınıflandırma

Olağan şartlarda, sınıflandırma işleminin başında sınıflar (*S* kümesi) ve sınıflandırılacak metinler (*B* kümesi) tamamen bilinir. Ancak farklı uygulamalarda zaman içerisinde *S* kümesine yeni sınıflar veya *B* kümesine yeni belgeler eklenebilmektedir. Bu tür durumlarda metinleri sınıflandırmak için sınıf odaklı (category-pivoted)

veya belge odaklı (document-pivoted) sınıflandırma ilkelerinden bir tanesinin benimsenmesi gereklidir. Belge odaklı sınıflandırma yönteminde, belirli bir b_i belgesiyle ilişkili olası bütün s_i sınıf etiketlerinin belirlenmesi hedeftir. Sınıf odaklı sınıflandırmada ise her bir sınıf s_i için, o sınıfa girmesi gereken bütün b_i belgelerinin belirlenmesi amaçlanır.

Belge odaklı sınıflandırma ilkesi, zaman içerisinde sisteme yeni belgelerin girdiği uygulamalar için daha uygundur. Bu tür bir uygulamaya örnek olarak haber metinlerinin sınıflandırılması verilebilir. Sistem tasarlanıp belirli sayıda belge sınıflandırıldıktan sonra, yeni sınıf ekleme ya da var olan sınıfları birleştirme biçiminde sınıf kümesi değiştirilecekse, sınıf odaklı çalışma düzeni daha doğru bir kullanım tercihi olacaktır.

Seçilen ilkeye göre yöntemler farklılık göstermektedir. Bazı yöntemler sadece bir tür çalışma düzenini gerçeklemektedir. Dolayısı ile teorik açıdan ciddi bir farklılık göstermeyen bu iki farklı çalışma ilkesi, uygulama geliştirme anlamında farklılıklar getirmektedir.

3.3 Sınıflandırmanın Kesinliği

Sınıflandırma işleminin sonucunda, belirli bir b_i belgesinin, belirli bir s_i sınıfında değerlendirilmesi durumu doğru ya da yanlış şeklinde kesin biçimde belirtiliyorsa, bu sınıflandırmaya kesin sınıflandırma (hard categorization) adı verilir. Bir diğer yöntem ise belirli bir b_i belgesi için, o belgeye atanması olası sınıfların sıralı bir listesinin hazırlandığı sıralı sınıflandırma (ranking categorization) yöntemidir. Bu listedeki sınıflar, seçilen yönteme göre belge ile en çok ilişkili olduğu hesaplanan sınıftan, en az ilişkili olan sınıfa doğru sıralanır. Listenin başındaki sınıf, belgeyle en çok ilişkili olduğu hesaplanmış sınıf olmasına karşın yöntem kesin bir yargı sonucunu üretmez ve çıktı olarak sadece bu sıralı listeyi üretir. Bu sıralı listede, S kümesinde tanımlı bütün sınıflar bulunabileceği gibi, sadece en yüksek ilişkili değere sahip N tane ($N \leq m$) sınıf da bulunabilir.

Sıralı sınıflandırma yöntemi, genellikle otomatik sınıflandırmanın hatalarının istenmediği, yüksek doğruluk gerektiren kritik işlemleri kolaylaştırmak amacı ile kullanılır. Sistem üzerinde çalışan işletmenlerin (operatör) karşısına en olası N sınıfı içeren bir liste sunularak işletmenin bu olası sınıflar içerisinden doğru olanı işaretlemesine yardım edilmiş olunur. Bu yöntem özellikle eğitim veri kümesinin kısıtlı olduğu, eğitim kümesi dışında kalan farklı tür belgelerin de sınıflandırma olasılığının yüksek olduğu durumlarda tercih edilir.

4. Uygulama Alanları

Metin sınıflandırmanın birçok farklı kullanım alanı bulunmaktadır. Bu alanlardan bazıları bu bölümde tanıtılacaktır.

Metin süzme (text filtering) uygulaması, bir kaynaktan gelen metinlerin işlenerek belirli sınıflara atanması işlemidir. Bu uygulama türlerinin en tipik örneği, yaramaz e-postaların süzülmesi için hazırlanan araçlardır (spam filter). Geliştirilen sınıflandırıcılar, gelen e-postaları metin içeriklerine göre normal ya da yaramaz sınıflarından birine atarlar [2-4]. Yaramaz olarak işaretlenen e-postalar uygulamaya göre doğrudan silinebilir ya da ayrı bir saklama bölgesine kaldırılabilir. Yaramaz e-posta süzme uygulamalarının standart uygulamalardan farklı bir özelliği bulunur: Sınıflandırıcının yaptığı hatanın maliyeti simetrik değildir, yani yaramaz bir e-postayı süzememe hatasının maliyeti, normal bir e-postayı yaramaz diye nitelendirip süzme hatasının maliyetine göre çok daha önemsizdir. Bu uygulamalar için geliştirilen sınıflandırıcılar, bu asimetrik hata maliyeti göz önüne alınarak ve maliyeti yüksek hatayı daha az yapacak biçimde geliştirilir. Metin süzme uygulamalarının bir diğer örneği de haberlerin sınıflandırılarak kullanıcının ilgisini çekmeyecek haberlerin süzülmesidir.

İndeks oluşturulması, metin sınıflandırmanın bir başka uygulama alanıdır. Bilgi bul getirimi (information retrieval) için belgeler genellikle anahtar sözcükler ve/veya sözcük grupları ile eşleştirilir. Kullanıcılar, bu anahtar sözcükler üzerinden Boolean aramalar gerçekleştirerek gereksinim duydukları belgelere erişirler. Anahtar sözcükler, önceden hazırlanmış denetimli söz varlığı kümesinden (controlled vocabulary) seçilmektedir. Söz varlığı kümesi ise konusunda uzman kişiler tarafından hazırlanır. Örneğin tıp alanında yaklaşık 25 bin sözcükten oluşan MeSH (Medical Subject Headings) isimli bir söz varlığı bulunmaktadır. Yeni gelen bir metne ya da belgeye, kullanılan denetimli söz varlığı kümesinden içeriğine uygun olacak şekilde anahtar sözcüklerin seçilmesi ve atanması bir metin sınıflandırma uygulamasıdır. Bu tür uygulamalarda söz varlığındaki anahtar sözcüklerin tamamı birer sınıf olarak nitelendirilir. Diğer bir deyişle söz varlığı kümesi sınıflandırma tanımındaki S kümesi görevini görmektedir. Her bir belge için bu

kümeden belirli sayıda sınıfa atama yapılır (anahtar sözcük seçilir). Bu nedenle bu tür uygulamalar belge odaklı çok etiketli uygulamalara birer örnektir.

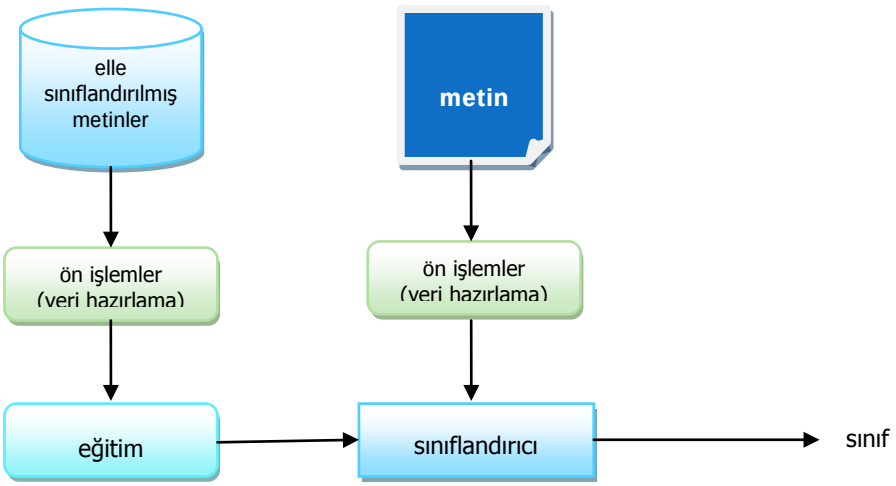
Metinlerin bilinmeyen bilgilerinin tahmin edilmesi de metin sınıflandırma uygulaması olarak gösterilir. Metin dilinin, yazarının ya da türünün belirlenmesi ile ilgili farklı metin sınıflandırma uygulamaları bulunmaktadır. İçerik dili bilinmeyen bir metnin dilinin belirlenmesi sorununda sınıflar, sistemin tanıdığı dillerden seçilir. Benzer biçimde sistemin tanıması istenen yazarlardan sınıflar seçilirse, gerçekleştirilen sınıflandırıcı yeni gelen bir metin için bir sınıf (yazar) belirleyebilir. Kuşkusuz bu sınıflandırıcı, yazarın önceki metinlerinde kullandığı sözcük, sözcük öbekleri, sözdizimsel yapılar üzerinden eğitilmelidir. Metin türünün belirlenmesi de bir metnin makale, haber, roman, vb. metin türlerinden hangisine daha çok benzediğini belirlemek üzere kullanılmaktadır. Doğal olarak bu durumda sınıflar, sistemin tanıması istenilen metin türleri olmaktadır.

5. Metin Sınıflandırmada Makine Öğrenmesi

Makine öğrenmesi, " bir performans ölçütünü eniyilemek etmek adına örnek verileri ve önceki deneyimlerin kullanılarak bilgisayarların programlanmasıdır" biçiminde tanımlanmaktadır [5]. Makine öğrenmesinin temelinde, konunun uzmanı tarafından sınıf etiketleri elle belirlenmiş bir belge kümesi üzerinde olarak eğitilerek belgelerin hangi sınıfa girmesi gerektiğine ilişkin özellikleri otomatik olarak öğrenen bir sınıflandırıcı hazırlanır. Bu sınıflandırıcıya yeni bir belge geldiğinde sınıflandırıcı öğrendiği özelliklere göre belgenin sınıfını belirler. Makine öğrenmesi açısından sınıflandırma sorunu "gözetimli öğrenme" (supervised classification) yöntemidir. Makine öğrenmesi ile metin sınıflandırmanın çalışma biçimi Şekil 2'de gösterilmiştir.

5.1 Eğitim ve Sınama Verisi Kümeleri

Makine öğrenmesi yöntemlerinin gerçekleştirilmesinde kuşkusuz en önemli etken elle sınıflandırılmış eğitim kümesidir. Bunun yanında en yüksek başarımı sağlayacak sınıflandırma algoritmasının seçilmesi ve uygun parametrelerin bulunması da gerekmektedir. Bu amaçla genel bir eğilim olarak elle sınıflandırılmış veri 2 gruba bölünür: eğitim kümesi (training set) ve sınama kümesi (test set). Bu kümelerin dengeli bir biçimde olması gerekmez. Eğitim kümesi, algoritmaların öğrenmesini sağlamak üzere kullanılırken, test kümesi ise sınıflandırıcının başarımının ölçülmesinde kullanılmaktadır. Eğitim kümesinde yer alan metinler, başarımın ölçülmesinde kullanılan sınama kümesinde yer almamalıdır. Gerçekleştirilen sınıflandırıcı, eğitim kümesindeki veriler kullanılarak geliştirilmiştir ve adil bir değerlendirme için sınama kümesinde eğitim kümesinden herhangi bir veri yer almamalıdır. Bu düzende sınıflandırıcı daha önce görmediği test kümesindeki metinleri sınıflandırır ve bu sınıflandırma sonuçları ile



Şekil 2- Makine öğrenmesi ile metin sınıflandırma

metinlerin önceden elle yapılan sınıflandırma sonuçları karşılaştırılarak başarım bulunur.

Eğitim ve sınama kümelerinin oluşturulması genelde rastgele yapılır. Sınıflandırıcının başarımının doğru biçimde ölçülebilmesi için hem eğitim kümesinin hem de sınama kümesinin yeterli temsil yeteneğine sahip olması yani tipik metinleri içermesi gereklidir. Bunu sağlamak üzere rastgele bölmeleme yaparak eğit-ve-sına adımlarını izlemek yerine k katlı çapraz doğrulama (k-fold cross-validation) yöntemi yaygın olarak kullanılmaktadır. Bu yöntemde elle etiketlenmiş toplam veri kümesi, ortak eleman içermeyen *k* tane farkı gruba ayrılır. İlk adımda, bu gruplardan birincisi sınama kümesi olarak seçilir ve geri kalan *k-1* tanesi eğitim kümesi olarak kullanılır. Sınıflandırma algoritması belirlenen bu *k-1* tane grup içeren eğitim kümesi ile eğitildikten sonra sınama kümesi üzerinde bir başarım hesaplanır. İkinci adımda ikinci grup sınama kümesi olarak seçilir. Geri kalan *k-1* grup eğitim kümesi olarak belirlenir, algoritma bu küme kullanılarak eğitilir ve başarımı sınama kümesinde hesaplanır. Bu çalışma mantığında *k* adımda eğitim ve sınama gerçekleştirildikten sonra başarımların ortalaması alınarak son başarım değeri olarak belirlenir. Genelde *k=10* ya da *k=5* değerleri sıklıkla kullanılmaktadır.

Eğitim kümesi üzerinde eğitilen algoritmalarda çoğunlukla belirli parametreler kullanılmaktadır. Bazı uygulamalarda, bu parametrelerin eniyilemesinin (optimization) yapılabilmesi için eğitim kümesi ve sınama kümesi dışında bir geçerleme kümesi (validation set, hold-out set) de hazırlanmaktadır.

5.2 Metin (Belge) Temsili

Makine öğrenmesi algoritmalarının eğitilebilmesi ve kullanılabilmesi amacıyla metinlerin algoritmaların üzerinde işlem yapabileceği biçime dönüştürülmesi gerekmektedir. Eğitim, geçerleme ve sınama kümesi içerisindeki bütün metinler bir ön işlemden geçirilmeli ve içeriğini temsil edecek sistematik bir gösterime dönüştürülmelidir. Metnin içeriği, temel olarak içerdiği sözcüklerle ve sözcüklerin diziliminden ortaya çıkan anlamlarla belirlenir. Ancak çoğu uygulamada basitlik açısından sadece sözcükler ya da sözcük grupları göz önüne alınır ve sözcüklerin sıralamaları ihmal edilir. Metni temsil etmek üzere seçilen sözcük ya da sözcük öbeklerine terim (term, feature) adı verilir. Her bir metin (belge), seçilen terimlerin ağırlıklarından (term weight) oluşan bir $\overrightarrow{b_j} = \{a_{1j}, a_{2j}, \dots, a_{|T|j}\}$ vektörü olarak temsil edilir. $T = \{t_1, t_2, \dots, t_{|T|}\}$ kümesi ise eğitim kümesindeki en az bir metinde geçen terimler içerisinden seçilerek oluşturulur. Metni temsil eden vektördeki elemanlar, terim ağırlıkları, $0 \leq a_{ij} \leq 1$ değer aralığından, ilgili terimin metin içeriğini ne kadar temsil ettiğini gösterecek biçimde değer alır.

Genelde terim olarak sözcükler işlenmektedir. Bu yaklaşıma sözcük sepeti (bag of words) adı verilir; çünkü metni temsil ettiği düşünülen sözcükler metin içindeki kullanımından ve yerinden bağımsız olarak (sanki bir sepete atılıyormuş gibi) bir vektörde toplanmaktadır. Yapılan çalışmalarda daha karmaşık gösterim biçimlerinin (sözdizimsel açıdan ya da istatistiksel açıdan beraber kullanılan sözcük öbekleri gibi) kayda değer başarım artışları sağlamadığı görülmüştür [1].

Terim ağırlığının hesaplanmasında da farklı uygulamalar gözlenmektedir. Örneğin ikili ağırlıklandırmada, herhangi bir t_i terimi, bir b_j metninde geçiyorsa $a_{ij}=1$, geçmiyorsa $a_{ij}=0$ olarak hesaplanır.

Tablo 1 - İngilizce metin, terim örnekleri ve ikili gösterimde belgelere ilişkin vektörler

Metin No	Metin	Terimler
b_1	web web graph	web graph
b_2	graph web net graph net	graph web net
b_3	page web complex	page web complex

V = [web, graph, net, page, complex]

V_1 = [1 1 0 0 0]

V_2 = [1 1 1 0 0]

V_3 = [1 0 0 1 1]

Ağırlıkların belirlenmesi için bir diğer yöntem ise bilgi bul getir alanında kullanılan terim sıklığı – belge sıklığının tersi TS-BST (term frequency – inverse document frequency TF-IDF) yöntemidir.

TS-BST hesabında aşağıdaki iki basit ilke temel alınmaktadır:

- Bir belge içerisinde sıkça geçen bir terim, belgede sadece bir kez geçen bir diğer terime göre daha önemlidir (ya da içerik hakkında daha çok ipucu verir). Buna Terim Sıklığı (term frequency) adı verilir ve aşağıdaki gibi hesaplanır:

$$TS_{ij} = \frac{n_{ij}}{|b_j|}$$

  Bu formülde herhangi bir b_j belgesinde (metninde) t_i teriminin kaç defa geçtiği n_{ij} bilgisidir. $|b_j|$ ise metin içerisindeki toplam (tekrarlı) terim sayısını göstermektedir.

- Bir metin kümesinde az sayıda belgede geçen terimlerin, belgenin içeriği ile ilgili ayırıcılık sağlama olasılığı yüksektir. Örneğin haber metinlerinin sınıflandırılmasında, *"parite"* teriminin ekonomi konulu haberlerde geçme olasılığı yüksek iken, kümedeki diğer belgelerde rastlanma olasılığı düşüktür; bu yüzden ayırıcılığı fazladır. Ancak *"başarı"* terimi ele alındığında, hemen hemen her konudaki (ekonomi, spor, siyaset vb.) metinlerde geçtiği, dolayısı ile sınıflandırmada ayırıcılık değerinin yüksek olmadığı kolaylıkla görülecektir. Bu mantıktan yola çıkılarak Belge Sıklığının Tersi (inverse document frequency) hesaplanır:

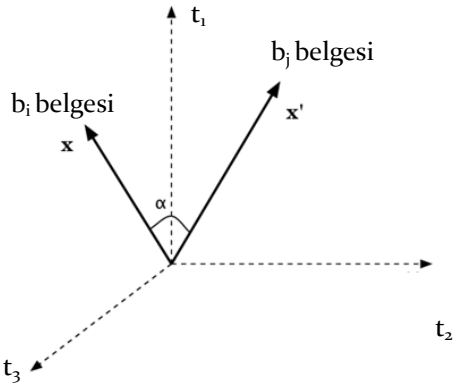
$$BST_i = \log\frac{n}{n_i}$$

  Bu formülde n_i sayısı, t_i teriminin geçtiği belge sayısını; n ise kümedeki toplam belge sayısını ifade etmektedir.

Yukarıda açıklanan iki bileşen birleştirilerek bir t_i teriminin b_j belgesi için ağırlığı a_{ij} aşağıdaki biçimde bulunur:

$$a_{ij} = (TS - BST)_{ij} = TS_{ij} \times BST_i$$

Belgeler vektörel biçime dönüştürüldüğünde artık her bir belge vektör uzayı modelinde (vector space model) bir vektör olarak temsil edilmektedir. Bu $|T|$ boyutlu uzayın her bir boyutu bir terimi göstermektedir. Aşağıdaki şekilde 3 terim içeren bir vektör uzay modelinde b_i ve b_j belgeleri x ve x' vektörleri ile ifade edilebilmektedir.



Şekil 3 – Vektör uzayı modeli

Vektör uzayı gösteriminde terimlerin seçilmesi de ayrı bir ön işlem olarak karşımıza çıkmaktadır. Tüm belgelerde geçen tüm sözcüklerin terim olarak seçilmesi, uzayın boyutunu çok büyütmekte ve verilerin bilgisayarla işlenmesini zorlaştırmaktadır. Bu nedenle belgeler vektörlere dönüştürülürken, T kümesi daha az sayıda eleman içeren T' kümesine $|T'| < |T|$ indirgenir. Bu işleme boyut indirgemesi (dimensionality reduction) adı verilir. Hangi terimlerin indirgenmiş terim kümesine dahil edileceğini belirlemek için bilgi kazancı (information gain), karşılıklı bilgi miktarı (mutual information), ki-kare (chi-square), olasılık oranı (odds ratio), ilişkililik puanlaması (relevancy score) vb. gibi farklı yöntemler kullanılmaktadır [6].

Boyutu indirgemek amacıyla bazı ek işlemler de yapılabilmektedir. Bu işlemler genelde dile özgü doğal dil işleme teknikleri kullanılarak uygulanmaktadır. En yaygın uygulamalardan bir tanesi, dilde çok sık geçen terimlerin (sözcükler) belirlenerek bunların T kümesinden çıkartılmasıdır. Böylelikle ayırıcılığı sağlayamayacak kadar sık geçen terimler devre dışı bırakılmış olur. Bu işleme sık geçen sözcüklerin ayıklaması (stop word list removal) denilmektedir. Aşağıda Türkçede ve İngilizcede en sık geçen sözcüklerin kırpılmış bir listesi verilmiştir.

Tablo 2 - Türkçede ve İngilizcede en sık geçen sözcüklerin kırpılmış listesi

İngilizce		Türkçe	
I	it	ve	kadar
a	of	bu	tam
about	on	da	ne
an	or	eğer	ise
are	that	de	iki
as	the	icin	var
at	this	ile	çünkü
be	to	olarak	büyük
by	was	kabul	yeni
com	what	daha	her
de	when	çok	o
en	where	en	bütün
for	who	gibi	ilk
from	will	artık	son
how	with	sonra	ancak
in	the	olan	değil
is	www	ama	fakat

Ek almış sözcükler de terim sayısının artmasının bir başka nedenidir. Türetilmiş sözcüklerin köklerinin bulunması ve sadece köklerin terim olarak seçilmesi hem temsil uzayının boyunu azaltmakta, hem de başarıma olumlu katkı sağlamaktadır [7]. Metinde geçen sözcüklerin köklerinin bulunması işlemine gövdeleme (stemming) adı verilmektedir. İngilizce için 100 satırı aşmayan program koduyla gerçekleştirilen bu işlem (Porter stemmer [8]), Türkçe gibi bitişken diller için ciddi bir sorun oluşturmaktadır. Türkçenin eklemeli ve üretken biçim bilimsel yapısı sayesinde, sadece bir sözcük kökünden bir milyondan fazla çekimli sözcük üretilebilmektedir [9]. Dolayısı ile Türkçe metinler üzerinde gövdeleme yapılmadan vektör uzayı modeli tabanlı bir metin sınıflandırma işlemi pratikte çok ciddi sorunlara yol açmaktadır.

5.3 Başarım Ölçütleri

Metin sınıflandırmanın başarım ölçütleri de bilgi bul getir (information retrieval) alanın başarım ölçütlerine benzemektedir: kesinlik (precision) ve bulma (recall). Bu ölçütlerin hesaplanmasında sınıflandırma sonuçlarına ilişkin sayıları içeren Tablo 1 kullanılır. Bu tablo, belirli bir s_i sınıfı için geçerlidir. Tablodaki değerler, gerçek sınıflar ile sistemin kararlarının ne kadar uyuştuğunu ya da uyuşmadığını belirtir.

DP_i (doğru pozitif), sistemin kararının pozitif olduğu (yani sistemin metnin s_i sınıfına ait olduğu sonucunu ürettiği durumları) ve bu metnin gerçekte de s_i sınıfına ait olduğu (yani elle etiketleyen uzmanın da bu metni s_i sınıfına atamış olduğu) durumların sayısıdır. Diğer bir deyişle gerçekte s_i sınıfında olan DP_i tane metin, sistem tarafından doğru bir biçimde s_i sınıfına atanmıştır. Benzer şekilde, gerçekte s_i sınıfında olmayan DN_i (doğru negatif) tane metin için de sistem tarafından doğru bir biçimde s_i sınıfında değildir diye karar üretilmiştir. YP_i (Yanlış Pozitif) değeri gerçekte s_i sınıfında olmayan metinlerin sistem tarafından s_i sınıfına atandığı durumların sayısını; YN_i (Yanlış Negatif) değeri de gerçekte s_i sınıfına ait olan ancak sistem tarafından s_i sınıfına atanmayan durumların sayısını göstermektedir.

Kesinlik terimi, herhangi bir b_j metninin, sınıflandırıcı tarafından s_i sınıfına atanması durumunda bu atamanın doğru olma olasılığını belirtmektedir. Bulma oranı ise gerçekte s_i sınıfına ait olan belgelerin kaç tanesinin sınıflandırma sonucunda s_i sınıfına atanır.

$$Kesinlik_i = \pi_i = \frac{DP_i}{DP_i + YP_i}$$

$$Bulma_i = \rho_i = \frac{DP_i}{DP_i + YN_i}$$

Tablo 1 : s_i sınıfı için sınıflandırma doğruluk tablosu

s_i sınıfı		ELLE ETİKETLEYEN UZMANIN KARARI	
		EVET	HAYIR
SINIFLANDIRICININ KARARI	EVET	DP_i	YP_i
	HAYIR	YN_i	DN_i

Hesaplanan bu değerler, s_i sınıfı için geçerlidir. Genel sınıflandırma başarımının hesaplanmasında ise mikro ortalama (micro-averaging) ve makro ortalama (macro-averaging) yöntemlerinden bir tanesi seçilir.

Tablo 2 : Mikro ve makro ortalama formülleri

Mikro Ortalama	Makro Ortalama
$\pi^{\mu} = \frac{\sum_{i=1}^{k} DP_i}{\sum_{i=1}^{k}(DP_i + YP_i)_i}$	$\pi^{M} = \frac{1}{K}\sum_{i=1}^{K} \pi_i$
$\rho^{\mu} = \frac{\sum_{i=1}^{k} DP_i}{\sum_{i=1}^{k}(DP_i + YN_i)_i}$	$\rho^{M} = \frac{1}{K}\sum_{i=1}^{K} \rho_i$

Başarım ölçümü için kesinlik ve bulma dışında başka ölçütler de bulunmaktadır. Bunlardan en yaygın kullanılanları doğruluk (accuracy - A) ve hata oranı (error rate - E) olarak adlandırılır.

$$Doğruluk_i = A_i = \frac{DP_i + DN_i}{DP_i + DN_i + YP_i + YN_i}$$

$$Hata\ Oranı_i = E_i = 1 - A_i$$

Başarımın artırılması anlamında hem kesinlik hem de bulma oranının 1'e yaklaşması amaçlanmaktadır. Bu iki ölçütün birden kullanılması zaman zaman zorluklara yol açtığından, bu iki değerden türetilen farklı başarım ölçütleri tanımlanmıştır: F Puanı (F-Score) [10-11], 11 puan ortalama kesinlik (eleven-point average precision) [12-13] ve başa baş noktası (breakeven point) [1, 10]. Literatürde yer alan çalışmaların sonuçları sunulurken ne yazık ki tek bir başarım ölçütünün kullanılması yönünde fikir birliğine varılamamıştır.

5.4 Yöntemler

Metin sınıflandırma için makine öğrenmesi uygulamalarında çok farklı algoritmalar kullanılmaktadır. Naive Bayes, karar ağaçları (decision tree), yapay sinir ağları (artificial neural network), örnek tabanlı sınıflandırıcılar (example based classifier), destek vektör makineleri (support vector machine) ve istatistiksel dil modeli (statistical language model) tabanlı sınıflandırıcılar yaygın olarak tercih edilmektedir. Anılan yöntemlerin ayrıntıları bu bölümde verilmeyecektir ancak konunun anlaşılmasına yardımcı olması açısından Naive Bayes yöntemi ile karar ağaçları kısaca tanıtılacaktır.

5.4.1 Naive Bayes

Olasılık tabanlı sınıflandırma yöntemlerinden bir tanesi olan Naive Bayes modeli, basit varsayımlar temelinde kurulmuştur. Ancak bu modelin kullanıldığı çalışmalarda, varsayımların basitliğiyle ters orantılı biçimde başarılı sonuçlar elde edildiği görülmüştür. Naive Bayes modeli de diğer tüm olasılık tabanlı modeller gibi aşağıdaki formüldeki olasılığın en büyük değerini bulacak s_i sınıfını bulmaya çalışarak sınıflandırma yapmaktadır.

$$P(s_i|\vec{b}_j) = \frac{P(s_i) \times P(\vec{b}_j|s_i)}{P(\vec{b}_j)}$$

Bu formülde, $P(s_i|\vec{b}_j)$ olasılığı $\vec{b}_j$ belgesinin s_i sınıfında olması olasılığını ifade etmektedir. Formülde yer alan $P(\vec{b}_j)$, vektör uzayında rasgele seçilen bir belgenin $\vec{b}_j$ vektörüne sahip olması olasığı, $P(s_i)$ ise rasgele seçilen bir belgenin s_i sınıfında olması olasılığıdır. Sınıflandırma sorunu, belirli bir $\vec{b}_j$ belgesi için bu belgenin gireceği sınıfı belirlemek üzere $P(s_i|\vec{b}_j)$ olasığını en büyük yapan s_i sınıfını bulmaktır. Sabit bir $\vec{b}_j$ belgesi için $P(\vec{b}_j)$ olasılığı da sabit olduğundan sorun aşağıdaki şekilde indirgenir:

$$\underset{s_i}{\operatorname{argmax}} P(s_i|\vec{b}_j) = \underset{s_i}{\operatorname{argmax}} \frac{P(s_i) \times P(\vec{b}_j|s_i)}{P(\vec{b}_j)} = \underset{s_i}{\operatorname{argmax}} P(s_i) \times P(\vec{b}_j|s_i)$$

Bu formülde $P(s_i)$ olasılığının hesaplanması göreceli olarak daha basittir. En büyük olabilirlik kestirimi (maximum likelihood estimation) yöntemiyle kolayca hesaplanabilir:

$$P(s_i) = \frac{n_{s_i}}{n}$$

Burada n_{s_i}, eğitim kümesinde s_i sınıfındaki belgeleri sayısı, n ise eğitim kümesindeki toplam belge sayısıdır.

Oysa $P(\vec{b}_j|s_i)$ olasılığının hesaplanması ise bu kadar kolay değildir. Olasılık tabanlı yöntemler, bu olasılığın farklı biçimlerde hesaplanması ilkesine dayanmaktadır. Bu yöntemlerden bir tanesi olan Naive Bayes yaklaşımında ise işleri basitleştirmek adına belgeyi oluşturan vektörün boyutlarının istatistiksel olarak birbirlerinden bağımsız olduğu varsayılmaktadır. İstatistiksel açıdan aslında

gerçekle pek de örtüşmeyen bu varsayım ile $P(\vec{b}_j|s_i)$ olasılığının hesaplanması çok daha basitleşmektedir:

$$P(\vec{b}_j|s_i) = \prod_{k=1}^{|T|} P(a_{kj}|s_i)$$

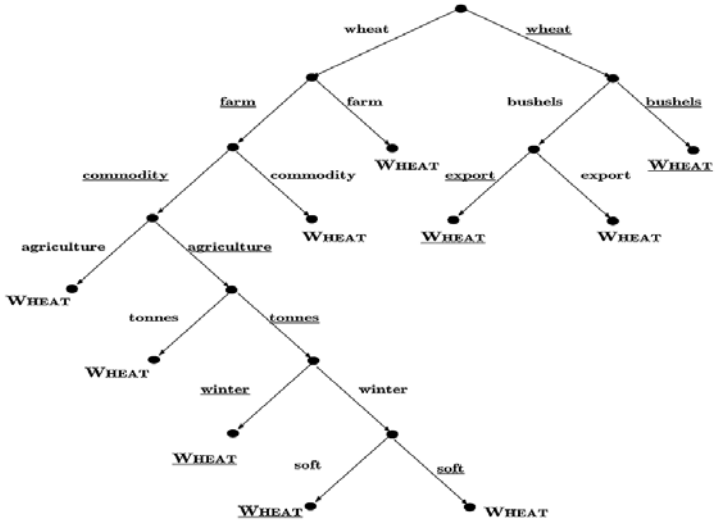
Bu formülden de anlaşılacağı gibi, Naive Bayes yönteminde her bir terimin, o sınıftaki kullanım olasılığı bulunur ve bu olasılıklar çarpılarak belgenin o sınıfa ait olma olasılığı bulunur.

5.4.2 Karar Ağaçları

Olasılık tabanlı yöntemlerle başarılı sonuçlar elde edilebilmekte ancak yöntemler sayısal verilerle işlem yaptığından çalışma biçimleri insanlar tarafından kolayca yorumlanamamaktadır. Oysa kural çıkarımlı sınıflandırıcılar ve karar ağaçlarının çalışma yöntemleri daha kolay anlaşılabilmektedir.

Metin sınıflandırmada kullanılan karar ağaçlarının, iç düğümleri terimleri, yaprak düğümleri (en alttaki düğümler) ise sınıfları göstermektedir. Her iç düğümden, yani terimden, çıkan iki dallanma bulunmaktadır. Bu dallanmalardan bir tanesi o terimin ilgili belgede bulunduğu durumu, diğeri de bulunmadığı durumu göstermektedir. Şekil 4'teki sınıflandırma kurallarına ilişkin karar ağacı aşağıdaki gibidir:

Bir belge sınıflandırılırken ilgili terimlerin o belgede yer alıp almamasına göre bu ağaç üzerinde gezilir ve sonuçta ulaşılan yaprak düğümün etiketine göre sınıflandırma yapılır. Örnekteki ağaç bir ikili sınıflandırıcıya ait ağaçtır, bu nedenle yaprak düğümleri WHEAT ya da WHEAT (wheat değil) etiketleri bulunmaktadır.



Şekil 4 - Karar ağacı örneği[1] [6]

Karar ağaçlarının oluşturulması ise makine öğrenmesi konusuna girmektedir. Bu amaçla ID3 , C4.5 [10] , C5 [14] gibi bazı standart yöntemler bulunmaktadır.

6. Türkçe İçin Metin Sınıflandırma Çalışmaları

Türkçenin bitişken ve üretken biçimbilimsel yapısı nedeni ile terim adaylarının sayısı diğer dillere oranla çok fazladır. Bu nedenle Türkçe için yapılan metin sınıflandırma çalışmalarının çoğunda gövdeleme kullanılmıştır. Türkçe metinler üzerinde farklı amaçlarla yapılmış farklı yöntemler kullanan çalışmalar bulunmaktadır. Metin yazarını bulma amacıyla yapılan bir çalışmada Naive Bayes, destek vektör makineleri ve karar ağaçları kullanılmıştır [15]. Türkçe e-postalar arasından yaramaz e-

postaları süzen bir sınıflandırma çalışmasında ise yapay sinir ağları yöntemi kullanılmıştır [16]. Yine Türkçe yaramaz e-postaların ayıklanması konusunda yapılan bir diğer araştırmada ise Naive Bayes, destek vektör makineleri ve yapay sinir ağı yöntemlerinin başarımları karşılaştırılmıştır [2].

Türkçe haber metinlerinin sınıflandırılması konusunda da çalışmalar gerçekleştirilmiştir. Bunlardan 20.000 haber metni üzerinde sınıflandırma yapan bir çalışmada *k* en yakın komşu (*k*NN) yöntemi ile aynı yöntemin çalışma süresini kısaltan bir başka sürümü (feature projection text categorization) karşılaştırılmıştır [17]. Bir diğer haber metni sınıflandırma makalesinde ise Naive Bayes ve yapay sinir ağları yöntemleri çok küçük bir veri kümesi üzerinde sınanmıştır [18].

Türkçe gövdelemenin metin sınıflandırma başarımına etkisinin ölçüldüğü bir çalışmada ise destek vektör makinaları ve centroid yöntemleri kullanılmıştır [19]. Çalışmanın sonucunda terim seçiminde gövdeleme yapmak yerine çekimli sözcükten sesli harflerin atılmasının en yüksek başarımı verdiği gözlenmiştir. Ancak bu uygulama biçiminin sadece terim sayısının az olduğu durumlarda geçerli olduğu belirtilmiştir.

Kaynakça

[1] Apté, C., F. Damerau, and S. Weiss, *Automated learning of decision rules for text categorization.* ACM Transactions on Information Systems (TOIS), 1994. 12(3): p. 233-251.

[2] Tantug, A.C. and G. Eryiğit, *Performance Analysis of Naïve Bayes Classification, Support Vector Machines and Neural Networks for Spam Categorization* in *Applied Soft Computing Technologies: The Challenge of Complexity*. 2006, Springer Berlin. p. 495-504.

[3] Androutsopoulos, I., et al. *An Experimental Comparison of Naive Bayesian and keyword based anti-Spam Filtering with personal email messages*. in *ACM International Conference on Research and Development in Information Retrieval*. 2000.

[4] Drucker, H., D. Wu, and V. Vapnik, *Support vector machines for spam categorization.* IEEE Transactions on Neural networks, 1999. 10(5): p. 1048-1054.

[5] Alpaydin, E., *Introduction to machine learning*. 2004: The MIT Press.

[6] Sebastiani, F., *Machine learning in automated text categorization*. 2002, ACM New York, NY, USA. p. 1-47.

[7] Peng, F., D. Schuurmans, and S. Wang, *Augmenting naive bayes classifiers with statistical language models.* Information Retrieval, 2004. 7(3): p. 317-345.

[8] Porter, M., *An algorithm for suffix stripping.* 1997.

[9] Hankamer, J. *Finite State Morphology and Left to Right Phonology*. in *West Coast Conference on Formal Linguistics Forum*. 1986: Stanford University.

[10] Cohen, W. and Y. Singer, *Context-sensitive learning methods for text categorization.* ACM Transactions on Information Systems (TOIS), 1999. 17(2): p. 141-173.

[11] Lewis, D. and W. Gale. *A sequential algorithm for training text classifiers*. 1994: Springer-Verlag New York, Inc.

[12] Schütze, H., D. Hull, and J. Pedersen. *A comparison of classifiers and document representations for the routing problem*. 1995: ACM New York, NY, USA.

[13] Yang, Y. *Expert network: Effective and efficient learning from human decisions in text categorization and retrieval*. 1994: Springer-Verlag New York, Inc. New York, NY, USA.

[14] Li, Y. and A. Jain, *Classification of text documents.* The Computer Journal, 1998. 41(8): p. 537-546.

[15] Amasyali, M. and B. Diri, *Automatic Turkish Text Categorization in Terms of Author, Genre and Gender*. Lecture Notes in Computer Science, 2006. 3999: p. 221.

[16] Özgür, L., T. Güngör, and F. Gürgen, *Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish.* Pattern Recognition Letters, 2004. 25(16): p. 1819-1831.

[17] İlhan, U., *Application of K-NN and FPTC based text categorization algorithms to turkish news reports*, in *Dept. of Comp. Eng.* 2001, Bilkent University: Ankara.

[18] Amasyali, M., T. Yildirim, and Y. Bolumu. *Automatic text categorization of news articles*. 2004.

[19] Cataltepe, Z., Y. Turan, and F. Kesgin, *Turkish Document Classification Using Shorter Roots*, in *IEEE 15th Signal Processing and Communications Applications, 2007. SIU 2007.* 2007. p. 1-4.