YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

METİN MADENCİLİĞİ İLE METİN SINIFLANDIRMA

İsmail Ferhat Pilavcılar

FBE Matemetik Mühendisliği Anabilim Dalı Matematik Mühendisliği Programında Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Nilgün GÜLER BAYAZIT

İSTANBUL, 2007

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ iii

ŞEKİL LİSTESİ iv

ÇİZELGE LİSTESİ v

ÖNSÖZ vi

ÖZET vii

1. GİRİŞ 1

2. VEKTÖR UZAY MODELİ 4

3. ÖN İŞLEME (PRE PROCESSING) AŞAMASI 6

3.1 Ön İşleme Genel Adımları 6
3.1.1 Joker (Wild Card) Yöntemi 7
3.1.2 Veri Filtreleme ve Vektörün Ağırlıklandırılması 10
3.1.3 Kelime Değerleri 11

4. ALGORİTMALAR 14

4.1 K-NN Algoritması 14
4.1.1 K-NN Algoritması Bit Ağırlıklandırma Yöntemi 16
4.1.2 K-NN Algoritması Frekans Ağırlıklandırma Yöntemi 19
4.1.3 K-NN Algoritması Tf-IDF Ağırlıklandırma Yöntemi 21
4.2 Naive Bayes Olasılık Metodu 23
4.2.1 Naive Bayes ile Binary Ağırlıklandırma (Multi-Variate Model) 24
4.2.2 Naive Bayes Frekans Ağırlıklandırma (Multi-Nominal Model) 28

5. YÖNETİM PROGRAMI 31

5.1 Ana Bölüm 31
5.2 Sözlük 32
5.2.1 Kelimeler Tablosu 33
5.2.2 Jokerler Tablosu 35
5.3 Eğitim Dokümanları 37
5.3.1 Eğitim Dokümanları Tablosu 38
5.3.2 Config Tablosu 39
5.3.3 Tekrar Sayıları Tablosu 40
5.4 Dokümanlar 41
5.4.1 Dokümanlar Tablosu 42
5.5 Kategoriler 42
5.5.1 Kategoriler Tablosu 43
5.6 Akış Diyagramı 44

6 UYGULAMALAR 45
6.1 Veritabanı 45

6.2 Tablolar 46
6.2.1 Çapraz Doğrulama (Cross Validation) 47
6.2.2 K Defa Çapraz Doğrulama 47
6.2.3 Hatalı Sonuçlardan Örnekler 56
7. SONUÇLAR 58
KAYNAKLAR 61
INTERNET KAYNAKLARI 63
ÖZGEÇMİŞ 64

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Vektör uzay modeli 4
Şekil 4.1 Vektör uzay modelinde dokümanların gösterimi 16
Şekil 4.2 Karakaçan projesi ekran görüntüsü 27
Şekil 4.3 Karakaçan projesi ekran görüntüsü 2 28
Şekil 5.1 Yönetim programı ana bölümü 31
Şekil 5.2 Yönetim programı sözlük bölümü 32
Şekil 5.3 Kelimeler tablosu 33
Şekil 5.4 Terim uzayı 35
Şekil 5.5 Jokerler tablosu 35
Şekil 5.6 Yönetim programı eğitim dokümanları bölümü 37
Şekil 5.7 Eğitim dokümanları tablosu 38
Şekil 5.8 Config (ayarlar) tablosu 39
Şekil 5.9 Tekrar sayıları tablosu 40
Şekil 5.10 Yönetim programı dokümanlar bölümü 41
Şekil 5.11 Dokümanlar tablosu 42
Şekil 5.12 Yönetim programı kategori işlemleri bölümü 42
Şekil 5.13 Kategoriler tablosu 43
Şekil 5.14 Yönetim programı akış diyagramı 44
Şekil 6.1 Algoritmalar ve çalışma süreleri 53
Şekil 6.2 Algoritma başarıları 54

ÇİZELGE LİSTESİ

Sayfa

Çizelge 6.1 Kullanılan veritabanı ve makale sayısı 45
Çizelge 6.2 K değerlerine göre K-NN algoritması başarıları 46
Çizelge 6.3 K-NN frekans algoritmasının kategori bazında başarısı (K=3 için) 48
Çizelge 6.4 K-NN frekans algoritması hata matrisi 48
Çizelge 6.5 K-NN bit algoritmasının kategori bazında başarısı (K=3 için) 49
Çizelge 6.6 K-NN bit algoritması hata matrisi 49
Çizelge 6.7 K-NN tf-idf algoritmasının kategori bazında başarısı (K=3 için) 50
Çizelge 6.8 K-NN tf-idf algoritması hata matrisi 50
Çizelge 6.9 Naive bayes algoritmasının kategori bazında başarısı (multinominal) 51
Çizelge 6.10 Naive Bayes algoritması hata matrisi 52
Çizelge 6.11 Algoritmaların dizin bazında başarıları (k=3) ve standart sapma (SS) değerleri 52
Çizelge 6.12 Algoritmaların eğitim dökümanları sayısına göre başarıları 55

ÖNSÖZ

Bilgisayarlar her geçen gün hayatımıza daha çok girmekte ve elimizdeki verileri saklayabileceğimiz, istediğimizde bu verilere ulaşabileceğimiz, arama yapabileceğimiz birer araç olmuşlardır. Bununla birlikte, son yirmi beş yıldır; internet, intranet veya veritabanlarında saklanan verinin boyutunda büyük bir artış olmuş ve buna bağlı olarak da, verinin çeşitli yöntemlerle organize edilme ihtiyacı doğmuştur. Tüm durumlarda amaç hep aynıdır. Böylesine büyük boyutta, biçimsiz ve karmaşık veriler arasından istenen veriyi bulup çıkarmak.

Metin halindeki verilerde ise bunu kolaylaştıran yöntem: Metin Kategorizasyonu'dur.

Bu tezde, metin halindeki veriye erişimi kolaylaştıran, zaman ve hız kazandıran Metin Kategorizasyon Teknikleri (Naive Bayes, k-NN) ve çeşitli ağırlıklandırma metodları incelenmiş olup, daha sonra bu teknikleri kullanarak VisualBasic.NET programlama dili ile metin kategorizasyonu programı yazılmış ve aynı zamanda ilgili tekniklerin doğru sınıflandırma olasılıkları açısından kıyaslamaları yapılmıştır.

Bu tezin hazırlanmasında yardımlarını ve desteğini esirgemeyen değerli hocam Sayın Yrd. Doç. Dr. Nilgün Güler BAYAZIT'a teşekkürü bir borç bilirim.

Mayıs 2007

ÖZET

Bilgisayarların çıkışı ve gelişmesiyle her geçen gün biraz daha değişen ve gelişen bir dünyada yaşamaktayız. Bilgisayarlar yaşantımıza birçok kolaylık katmakta, yapılan işlerin yükünü hafifletmekte, daha iyi sonuçlara, daha kısa yollardan ulaşmamızı sağlamaktadır. Bilgisayarlar aynı işi otomatik olarak ve daha verimli yapacağından insan kaynaklı hatalar en aza indirgenir.

Bilgisayarların gelişimine paralel olarak, insanlar daha fazla bilgiye erişim olanakları bulmuş ve günden güne, çok sayıda veriyi depolayan sistemler, yani veritabanları oluşturulmuş ve bu veritabanlarının boyutları da günden güne büyümüştür.

Çeşitli tipte veritabanları mevcuttur. Metin halindeki verilerin bulunduğu veritabanlarından bilgiyi kolayca elde etmek için metin kategorizasyon yöntemleri uygulanır. İlk zamanlarda insan aracılığıyla yapılan sınıflandırma, günümüzde doküman sayısının çok hızlı bir şekilde artması dolayısıyla otomatik olarak yapılır hale gelmiştir. Bunun için, daha önceden kategorileri tanımlanmış olan eğitim dokümanları yardımıyla metin halindeki veriler sınıflandırılabilmektedir.

Tezde, amaç doğrultusunda, metin halindeki verilerin sınıflandırılmasında kullanılan metin kategorizasyon teknikleri (Naive Bayes, k-NN) ve çeşitli ağırlıklandırma yöntemleri incelenmiş olup, daha sonra bu teknikleri kullanarak VisualBasic.NET programlama dili ile metin kategorizasyon programı yazılmış ve aynı zamanda ilgili tekniklerin doğru sınıflandırma olasılıkları açısından kıyaslamaları yapılmıştır. Bu tezde, metin sınıflandırması üzerinde çalışmak için Anadolu Ajansı adlı Türkçe bir veri kümesinin derlemesi sunulmuştur.

Anahtar Kelimeler: Metin kategorizasyonu, naive bayes ve k-nn algoritmaları, metin madenciliği, sınıflandırma, joker (wild card) yöntemi.

ABSTRACT

After the invention and development of computers, we have been living in a more different and developing world. Computers make our life easier and improve the quality of our life by providing beter results with easier ways. Since computers make the same work automatically and effectively, human sourced errors become less.

In the same way with the development of computers, human had the facility of access to too much data, and up to now, new systems, that is "databases", have been formed and these database systems have been getting bigger and bigger as the time passes.

There are different kinds of databases. Text categorization methods are used in order to get the information from the databases which includes text type data in. With the increase of the number of documents, classification has been being made automatically, not by humans. For this purpose, with the help of the keywords of which categories are determined firstly, text type data can be classified.

In that way, during the researching of this thesis, text categorization techniques which are used in text type data classification (Naive Bayes, K-NN) and various weightening methods are examined, then a text categorization programme has been done by using these techniques and VisualBasic.NET programming language, and at the same time exact classification probabilities of these techniques have been compared with each other. This thesis presents compilation of a Turkish dataset, called Anadolu Agency Newsgroup in order to study in Text Categorization.

Key Words: Text categorization, naive bayes and k-nn algorithms, text mining, classification, wild card method.

1. GİRİŞ

İnternete erişim kolaylığı, optik okuyucular, yüksek hızlı ağlar ve pahalı olmayan, yüksek miktardaki bilgi depolama imkanları gibi yeni teknolojik gelişmeler sonucu, online metin, makalelere, e-maillere, teknik raporlara vb. erişilebilirlik konusunda büyük bir artış yaşandı.

Metin madenciliği, özel amaçlar için metinden bazı bilgiler çıkarmak adına, metinin analiz edilmesi işlemidir (Visa, 2001). Metin sınıflandırması ise önceden belirlenmiş kategorilere göre, doğal dil metinlerinin sınıflandırılmasıdır (Soucy vd., 2001). Sonsuz sayıda doğal dil girdilerini, küçük bir kategoriler kümesine indirgemek, metin tabanlı bilgileri işleyen hesaplama sistemlerinin temel stratejisidir.

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

Makine öğrenmesi (machine learning = ML) açısından düşünüldüğünde ise, yapılan son araştırmalar, veri madenciliği gibi yöntemler üzerine olmuştur. Makine öğrenmesine ihtiyaç duyulmasının nedeni, elle kategorizasyonun pahalı ve zaman tüketen bir iş oluşudur ki, ayrıca elle sınıflandırmada, sınıflandırmayı yapan uzmanların vermiş oldukları kararlara bağlı olarak sonuçlar da değişmektedir (İlhan, 2001).

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

- Metin bulup getirme ve kütüphane organizasyonu gibi alanlarda destek sağlayan, "dokümanlara kategori ataması yapmak",
- Bu kategorileri insanların atadığı uygulamalarda "kategori atamasında yardımcı rol oynamak",
- Mesajları, haberleri ve diğer "metin akımı (stream) halindeki bilgileri alıcılara ulaştırmak",

- Doğal dil işleme sistemlerinin bir parçası olarak; “ilgisiz metinleri ve metin parçalarını filtrelemek”, “metinleri, kategori bazlı işleme mekanizmalarına yönlendirmek” veya “sınırlı şekillerde bilgi edinimini sağlamak”, “sözcük analizi işlerinde yardımcı olmak (sözcük belirsizliğini giderme gibi)” vb.

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

Araştırmacılar, eğitim dokümanları kümesinin yeterince bilgi içermesi gerektiğini, ancak bunun yanında zaman performansını sağlamak adına büyük miktarda veri içermemesi gerektiğini ortaya koymuşlardır. Zaman performansını sağlayıcı nitelikte eğitim dokümanı bulundurmak, büyük veriler için kategorizasyon problemlerini çözmede önemli olmaktadır.

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

Türkçe gibi bitişken (bükümlü) dillerde ise kelimeler, en küçük anlamlı parçasının sınırlarına dair bir belirti göstermez, üstelik bu parçalar, morfolojik ve fonolojik şartlara bağlı olarak şekil alırlar. Türkçe’de bir kelimenin son ekine bir tane daha ekleyerek, nispeten uzun kelimeler elde edilebilir, üstelik, sadece bir tek Türkçe kelimeden çok miktarda değişik anlamlı kelimeler oluşturulabilir. Bu karmaşık morfolojik yapı yüzünden, Türkçe; İngilizce’den ve benzeri dillerden daha farklı metin işleme teknikleri gerektirir. Bu nedenle, bütün kelimelerin küçük harfe çevrilmesi ve noktalama işaretlerinin kaldırılması dışında; joker kelimeler ile anahtar kelimelerin oluşturulması gibi bazı ön hazırlıklar yapılması gerekmektedir (İlhan, 2001).

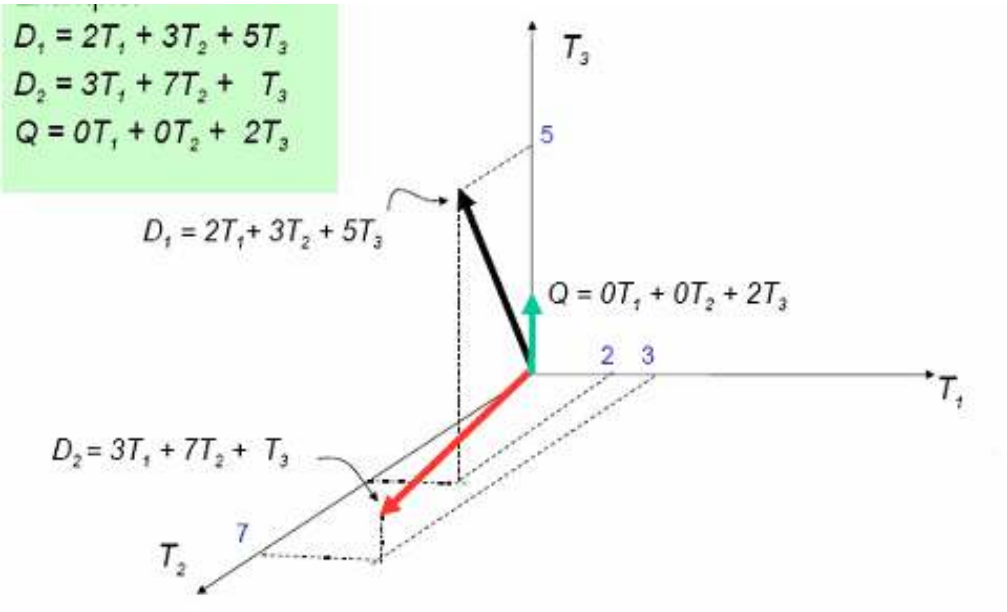
Bu tezde, örnek veri kümesi olarak, Anadolu Ajansı haberlerine ait veri kümesi kullanılmıştır. Bu veri kümesi yaklaşık olarak 1370 adet Türkçe haber metni içermektedir. Bu haber metinleri kategorizasyon işlemi için çeşitli işlemlere tabi tutulmuştur.

Bölüm 3’te ön işleme işleminin yapılma tekniği anlatılmış, bölüm 4’te tezde kullanılan algoritmalardan bahsedilmiş, üzerinde çalışmakta olduğum Karakaçan Projesi’ne de ana

hatlarıyla bu bölümde değinilmiştir. Kullanmış olduğumuz veri tabanında, Naive Bayes ve K-NN algoritmalarının performansı Bölüm 6'da incelenmiş ve son bölümde de bu tezden elde etmiş olduğumuz sonuçlara yer verilmiştir.

2. VEKTÖR UZAY MODELİ

Vektör uzay modeli bilgi çıkarımı, bilgi filtreleme, indeksleme gibi alanlarda kullanılan cebirsel bir modeldir. Doğal dil belgelerinin çok boyutlu uzayda özel bir anlamını simgelemektedir.



Şekil 2.1 Vektör uzay modeli

Dokümanlar şekilde görüldüğü gibi kelimelerin vektörleri olarak ifade edilirler. T'ler aslında kelimeleri ifade etmektedirler (Jun ve Houkuan, 2002).

Anahtar kelime araması yapılan dokümanların ilişki düzeyleri, doküman benzerlik teorisindeki varsayımlar kullanılarak, yani her bir doküman vektörü ile (eğitim dokümanı) sorgu vektörü arasındaki açıların sapmalarını karşılaştırarak, hesaplanabilir.

Vektörler arasındaki gerçek açıların hesaplanması yerine, vektörler arasındaki açının kosinüsü hesaplanır ve karşılaştırılır.

Metin sınıflandırma işlemi için ilk yapılması gereken sınıflandırmak istediğimiz dokümanı ve

eğitim dokümanlarımızı vektörel olarak uzayda ifade edebilmektir. Bir çok metin sınıflandırma algoritması bu prensibe dayanır. Bunun için uzay eksenlerini belirlemeliyiz. Uzayımızın eksenlerini aslında bizim kategori belirttiğini düşündüğümüz kelimeler oluşturacaktır. Bu kelimeler de sözlükte, yani kelimeler tablosunda tutulmuştur.

Dokümanlarımızı vektör olarak temsil edebilmek için metnin içerisinde geçen kelimelerin bir takım ön işlemlere sokulması gereklidir. Bu işlemin sonucunda vektörlerimiz oluşacaktır.

Sistemimizin verilen metinden kategorileri otomatik olarak bulabilmesi için eğitilmesi gereklidir. Bunun için kategori belirten makaleler sistemimizi eğitmek için kullanılmıştır. Bu makaleler ile arka planda sözlüğümüzün boyutu ile sınırlandırılmış vektörler oluşturulup, kategorisi bulunması istenen dokümanın vektörü ile çeşitli algoritmalarla (K-NN, Naive Bayes) karşılaştırılarak sınıflandırılmak istenen doküman ilgili kategoriye atanmıştır.

3. ÖN İŞLEME (PRE-PROCESSING) AŞAMASI

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

1-Veri üzerinde bulunabilen problemleri çözmek,

2-Verinin doğal yapısını öğrenerek daha anlamlı ve kaliteli analiz yapabilmek,

3-Veriden daha anlamlı bilgi üretebilmek (İlhan, 2001).

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

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

3.1 Ön İşleme Genel Adımları

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

aşağıdaki gibidir:

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

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

Örnek:

Gözlükler

Gözlükte

Gözlüğü

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

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

3.1.1 Joker (Wild Card) Yöntemi

Sistemde metinlerdeki kelimelerin kendileri yerine gövdelerinin kullanıldığı daha önceden belirtilmişti. Bunun sebebi Türkçe gibi eklemeli dillerde bir gövdenin sonuna birçok farklı ek alarak farklı biçimlerde karşımıza çıkabilmesidir. Örneğin “araba” kelimesi ile “arabadan”, “arabayı”, “arabada”, ve “arabanın” kelimeleri eğer ayrıştırıcı olmasa ayrı ayrı kelimeler

olarak görüleceklerdi. Bunun sonucu olarak hem oluşturulan sözlük boyutu çok artacak hemde sınıflandırma başarısı düşecekti (Amasyalı ve Yıldırım, 2004).

Joker kelime, aynı söz dizimi ile başlayan ve çeşitli ekler almış ancak yakın anlamda olan sözcükleri tek bir gösterimle grup altında toplayan kelimelerdir. Joker kelime gövdeleme yöntemine benzemektedir. Gövdelemede çekim ve yapım eklerinden ayrıştırılan kelimeler, ortak bir köke indirgenir. Ancak burada köke indirgeme şartı yoktur. Kökün yanında ek de kalabilir (İlhan, 2001). Joker kelimeler kategoriyi belirlememize yardımcı anahtar kelimelerden veya sık kullanılan kelimelerden seçilir.

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

Örnek: simitçi ve simitçiler

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

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

(Ör:kitap* ve kitab* yerine kita*)

Not: Sert sessizler: Ç,F,T,H,S,K,P,Ş

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

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

Evlen*, evcil*

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

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

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

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

3.1.2 Veri Filtreleme ve Vektörün Ağırlıklandırılması

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

Taraftarlar İstanbul'da bırakıldı.

Sonuç: taraftarlar istanbul bırakıldı

Örnek: trafik kazası: 1 ölü...

Sonuç: trafik kazası 1 ölü

2- Noktalama işaretleri çıkarılmış dokümanda geçen tüm kelimeler bir diziye atılır.

Dizi: (Taraftarlar,İstanbul,bırakıldı)

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

taraftar*

İstanbul*

Gol*

Futbolcu*

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

3.1.3 Kelime Değerleri

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

-Terim Frekansı (TF),

-Ters Doküman Frekansı (IDF),

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

-Terim Ayrıştırma Değeri,

-Olasılıksal Terim Ağırlıklandırma,

-Tek Terim Doğruluğu,

-Genetik Algoritmalar.

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

Kelimeler(Sözlük) **Jokerleri**

enflasyon*

grip* ----------→ grib*,

hakem*

İlaç* ----------→ ilac*,

taraftar*

tarım*

Örnek eğitim dokümanları aşağıdaki gibi olursa;

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

2-İlacını aksatanlar hastalığa davetiye çıkarırlar. (Sağlık)

3-Yıllık enflasyon oranı bu senede yükselişte. (Ekonomi)

4-Tarımla uğraşanlar bu yıl tarımdan zarar edecekler. (Ekonomi)

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

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

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

Kategorisini bulmak istediğimiz metin aşağıdaki gibi olursa:

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

Sözlük={ enflasyon*, grip* , hakem*,İlaç*, taraftar*,tarım*}

Eğer vektörlerimizi kelime frekansına göre ifade edecek olursak;

D1=(0,2,0,1,0,0)

D2=(0,0,0,1,0,0)

D3=(1,0,0,0,0,0)

D4=(0,0,0,0,0,2)

D5=(0,0,2,0,0,0)

D6=(0,0,0,1,2,0)

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

Eğer vektörlerimizi bitsel olarak ifade etmek istersek;

D1=(0,1,0,1,0,0)

D2=(0,0,0,1,0,0)

D3=(1,0,0,0,0,0)

D4=(0,0,0,0,0,1)

D5=(0,0,1,0,0,0)

D6=(0,0,0,1,1,0)

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

4. ALGORİTMALAR

4.1 K-NN Algoritması

K en yakın komşuluk algoritması sorgu vektörünün en yakın k komşuluktaki vektör ile sınıflandırılmasının bir sonucu olan denetlemeli öğrenme algoritmasıdır. Bu algoritma ile yeni bir vektörü sınıflandırabilmek için doküman vektörü ve eğitim dokümanları vektörleri kullanılır. Bir sorgu örneği verilir, bu sorgu noktasına en yakın k tane eğitim noktası bulunur. Sınıflandırma ise bu k tane nesnenin en fazla olanı ile yapılır (Kutlu, 2001). K en yakın komşuluk uygulaması yeni sorgu örneğinin sınıflandırmak için kullanılan bir komşuluk sınıflandırma algoritmasıdır.

K en yakın komşuluk algoritması çok kolaydır. K en yakın komşulukları bulmak için sorgu örneği ile eğitim dokümanları arasındaki en küçük uzaklıklar dikkate alınır. En yakın komşuları bulduktan sonra bu komşulardan kategorisi en çok olanın kategorisi dokümanın kategorisini tahmin etmekte kullanılır.

Avantajları

Uygulanabilirliği basit bir algoritmadır.

Gürültülü eğitim dokümanlarına karşı dirençlidir.

Eğitim dokümanları sayısı fazla ise etkilidir.

Dezavantajları

K parametreye ihtiyaç duyar.

Uzaklık bazlı öğrenme algoritması, en iyi sonuçları elde etmek için, hangi uzaklık tipinin ve hangi niteliğin kullanılacağı konusunda açık değildir.

Hesaplama maliyeti gerçekten çok yüksektir çünkü her bir sorgu örneğinin tüm eğitim örneklerine olan uzaklığını hesaplamak gerekmektedir. Bazı indeksleme metodları ile (örneğin K-D ağacı), bu maliyet azaltılabilir.

En yakın komşuluk prensibine dayanır. Tüm dokümanlar vektörel olarak temsil edilir. Sorgu dokümanı ile diğer dokümanlar arasındaki kosinüs benzerliği hesaplanır. Similarty oranı 1'e en yakın olan n tane vektörün kategorisinden çok olanı dokümana atanır.

di =(wdi1,wdi2,……wdij)

wij terimin doküman içerisindeki ağırlığı, di eğitim dokümanı vektörüdür. q ise kategorisi bulunması istenen vektördür.

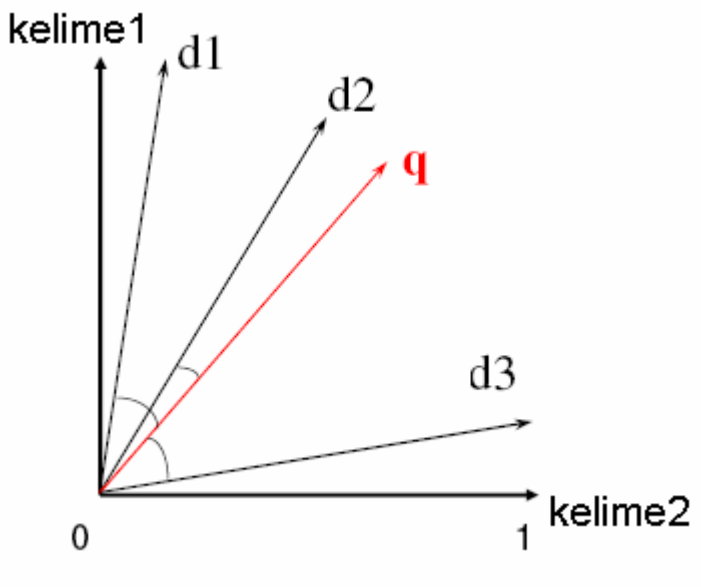
$$sim(d_i,q) = \cos\theta \tag{4.1}$$

$$sim(d_i,q) = \frac{d_i \cdot q}{|d_i||q|} = \frac{\sum_j W_{i,j} * W_{q,j}}{\sqrt{\sum_j W^2{}_{i,j}}\sqrt{\sum_j W^2{}_{q,j}}} \tag{4.2}$$

sim(d,q)=1 => d=q

sim(d,q)=0 ise terim paylaşımı yoktur.

Tüm dokümanlar ve kategorisi bulunması istenen doküman önceki bölümlerde açıklanan kurallar doğrultusunda vektörel olarak ifade edilirler. Her bir boyut aslında kelimelere karşılık gelmektedir.



Şekil 4.1 Vektör uzay modelinde dokümanların gösterimi

Burada d1, d2 ve d3 eğitim dokümanlarımızdan oluşan vektörler, q ise sınıfını bulmak istediğimiz vektördür.

K-NN algoritmasında terim ağırlıklandırma için 3 çeşit yöntem kullanılabilir.

1-Bit

2-Frekans

3-Tf-IDF

4.1.1 K-NN Algoritması Bit Ağırlıklandırma Yöntemi

Bu yöntemde vektörün ağırlıkları dokümanda bulunma veya bulunmamasına göre belirlenir. Yani bir kelimenin bir dokümanda 2 kere bulunması ağırlığını değiştirmeyecektir. Kelimenin

dokümanda olması 1, olmaması 0 ile ifade edilmelidir. Sözlükteki her bir kelime (Kelimeler tablosu) için dokümanda geçip geçmediği kontrol edilir, ağırlıklar bulunur.

Vektörü Bitsel İfadenin Sözde Kodu

For i=0 to TumKelimeler.Count-1

'Sözlükteki her bir kelimeye bakılıyor ve sırasıyla dokümanımızdan oluşturduğumuz kelime dizisi ile karşılaştırılıyor

for j=0 to DokumandakiKelimeler.count-1

if TumKelimeler(i)=DokumandakiKelimeler

'İlgili i boyutunu 1'e eşitle, 2.döngüden çık 1'e devam et.

Else

'Sözlükteki kelimemize eşit değilse bir de o kelimemizin jokerlerine bakalım.

For k=0 to Jokerler(i).Count-1

İf Jokeler(i)=Dokumandakikelimeler

'İlgili i boyutunu 1'e eşitle

exit for

endif

Next

Next

Next

Ön işleme aşamasında eğitim vektörlerimizi ve sorgu vektörünü ağırlıklandırmıştık. 3. bölümde kullandığımız eğitim dokümanlarını kullanarak bit ağırlıklandırma yöntemi ile kelimelerin dokümanda geçip geçmediğine göre vektörler aşağıdaki gibi oluşturulmalıdır.

D1=(0,1,0,1,0,0)

D2=(0,0,0,1,0,0)

D3=(1,0,0,0,0,0)

D4=(0,0,0,0,0,1)

D5=(0,0,1,0,0,0)

D6=(0,0,0,1,1,0)

DS=(0,0,1,0,1,0)

Bitsel olarak uzayda ifade ettiğimiz her vektörün sorgu vektörümüze (DS) olan uzaklığını hesaplamalıyız.

(4.2) eşitliği yardımı ile aşağıdaki değerler hesaplanmalıdır.

sim(d1,q)=0

sim(d2,q)=0

sim(d3,q)=0

sim(d4,q)=0

sim(d5,q)= $\frac{1}{1*\sqrt{2}}$ =0.707

sim(d6,q)= $\frac{1}{\sqrt{2}*\sqrt{2}}$ =0.5

Similarty'si 1'e en yakın olanı alacağız. Beş ve altıncı dokümanımız similarty değerleri 1'e en yakın olan dokümanlarımız oldukları için ve bu dokümanlarımızın kategorileri spor olduğu

için doküman spor kategorisindedir diyebiliriz.

4.1.2 K-NN Algoritması Frekans Ağırlıklandırma Yöntemi

Bu yöntemle vektörlerin ağırlıkları dokümanda kelimenin bulunma sayısına göre hesaplanır. Örneğin bir kelime bir dokümanda 2 kere bulunuyorsa ağırlığı 2 alınmalıdır.

Vektörü Frekans Olarak İfadenin Sözde Kodu

For i=0 to TumKelimeler.Count-1

'Sözlükteki her bir kelimeye bakılıyor ve sırasıyla dokümanımızdan oluşturduğumuz kelime dizisi ile karşılaştırılıyor

for j=0 to DokumandakiKelimeler.count-1

if TumKelimeler(i)=DokumandakiKelimeler

'İlgili i boyutunu 1 arttır, 2.döngüden çık 1'e devam et.

else

'Sözlükteki kelimemize eşit değilse bir de o kelimemizin jokerlerine bakalım.

For k=0 to Jokerler(i).Count-1

İf Jokeler(i)=Dokumandakikelimeler

'İlgili i boyutunu 1 arttır

exit for

endif

Next

'Jokerlerde varsa boyutu 1 arttırıp 1.döngüden devam edelim.

end if

Next

Next

Ön işleme aşamasında eğitim vektörlerimizi ve sorgu vektörünü ağırlıklandırmıştık. 3. bölümde kullandığımız eğitim dokümanlarını kullanarak frekans ağırlıklandırma yöntemi ile kelimelerin dokümanda geçme sayılarına göre vektörler aşağıdaki gibi oluşturulmalıdır.

D1=(0,2,0,1,0,0)

D2=(0,0,0,1,0,0)

D3=(1,0,0,0,0,0)

D4=(0,0,0,0,0,2)

D5=(0,0,2,0,0,0)

D6=(0,0,0,1,2,0)

DS=(0,0,2,0,1,0)

Tekrarlama frekansını baz alarak uzayda ifade ettiğimiz her vektörün sorgu vektörümüze olan uzaklığını hesaplamalıyız.

(4.2) eşitliği yardımı ile aşağıdaki değerler hesaplanmalıdır.

sim(d1,q)=0

sim(d2,q)=0

sim(d3,q)=0

sim(d4,q)=0

$$\text{sim(d5,q)}=\frac{4}{\sqrt{4}*\sqrt{5}}=0.8944$$

$$\text{sim(d6,q)}=\frac{2}{\sqrt{5}*\sqrt{5}}=0.4$$

Similarty'si 1'e en yakın olan dokümanlar üzerinden yorum yapmamız doğru olacağından beş ve altıncı dokümanlar 1'e en yakın olan dokümanlar oldukları için ve bu dokümanlarımızın kategorileri spor olduğu için doküman spor kategorisindedir diyebiliriz.

4.1.3 K-NN Algoritması Tf-Idf Ağırlıklandırma Yöntemi

Tf-idf, dokümanları vektör uzay modelinde tanımlayabilmemiz için kullanılan en önemli ağırlıklandırma metodlarından biridir. Metin kategorizasyonunda tf-idf ağırlıklandırma metodu 2 önemli öğrenme metodu ile ilişkilidir. Bunlar K-NN ve SVM'dir. Tf-idf ağırlıklandırmasında her bir dokümandaki kelimelerin frekansı rol oynamaktadır. Böylece dokümanda daha fazla görülen kelimeler varsa (TF, terim frekansı yüksek) o doküman için daha değerli olduğu anlaşılır. Ayrıca IDF tüm dokümanlarda seyrek görülen kelimeler ile ilgili bir ölçü verir. Bu değer tüm eğitim dokümanlarından hesaplanılır. Bu yüzden eğer bir kelime dokümanlarda sık geçiyorsa doküman için belirleyici olmadığı düşünülebilir (stop words). Eğer kelime dokümanlarda çok sık geçmiyorsa o kelimenin o doküman için belirleyici özelliği vardır diyebiliriz. Tf-idf genel olarak sorgu vektörü ile eğitim dokümanı vektörü arasındaki benzerlik oranını bulmak için kullanılır. Tf-idf fonksiyonunun çeşitli versiyonları mevcuttur (Soucy ve Mineau, 2005).

$$W_{d,t} = tf_{d,t} \bullet idf_t \tag{4.3}$$

$$idf_t = \log\left(\frac{D}{dft}\right) \tag{4.4}$$

tf (Dokümanda bulunan kelimenin dokümandaki görülme sıklığı)

D (Toplam doküman sayısı)

dft (Dokuman Frekansı(Kaç eğitim dokümanında ilgili kelime geçmiş))

Sistemimizde eğitim dokümanlarımızın tf-idf ağırlıkları eğitim dokümanları sisteme eklendikten sonra oluşturulur ve config tablosunda tutulur. Programın çalışması esnasında tf-idf ile yapılabilecek bir kategorilendirme işlemindeki sorgu vektörünün ağırlığı ise programın çalışma zamanında hesaplanır.

Tüm bu ağırlıklandırma formüllerinden biri kullanılarak ağırlıklar belirlenir. Similarty formülünden hesaplamalar yapılır. Similartysi 1'e en yakın olan n tane vektörün kategorisinden çok olanın kategorisi dokümana atanır.

3. bölümde kullandığımız eğitim dokümanlarını kullanarak tf-idf ağırlıklandırma yöntemine örnek vermek gerekirse öncelikle dokümanların frekans vektörleri bulunur.

D1=(0,2,0,1,0,0)

D2=(0,0,0,1,0,0)

D3=(1,0,0,0,0,0)

D4=(0,0,0,0,0,2)

D5=(0,0,2,0,0,0)

D6=(0,0,0,1,2,0)

DS=(0,0,2,0,1,0)

Şimdi D1 dokümanının tf-idf ağırlığını hesaplayalım (Örnek olarak 2. kelimenin (grip) ağırlığını bulalım).

Tf(grip)=2

D=6

dft=1

IDF=log(D/dft)=log(6/1)=0.778

(4.3) eşitliği yardımı ile tf-idf değeri aşağıdaki incelikte hesaplanabilir.

Tf*IDF=2*0,778=1,556

W1=(0,1.556,...)

Böylece grip boyutunun D1 dokümanındaki ağırlığını bulmuş olduk. Benzer şekilde diğer ağırlıkları da hesaplayıp vektörün tüm eksenleri için ağırlıklar bulunur. Tüm dokümanlar için bu işlem yapılır. Her doküman sorgu ile karşılaştırılır. Similarity'si 1'e en yakın olan dokümanlardan n tanesinden en çok olanının kategorisi sorgu dokümanının kategorisine atanır.

4.2 Naive Bayes Olasılık Metodu

Naive Bayes sınıflandırıcı algoritması metin dokümanlarını sınıflandırmak için kullanılan en başarılı algoritmalardan biridir. Uygulanabilirliği kolay bir algoritma olup performans bakımından da başarılıdır. Naive (saf) Bayes sınıflandırıcısı için binary bağımsızlık modeli ve multinominal model olmak üzere 2 model versiyonu mevcuttur. Naive Bayes'e göre kelimeler sınıftan bağımsızdır (Eyheralendy, 2003).

4.2.1 Naive Bayes Binary Ağırlıklandırma (Multi-Variate Model)

Klasik saf Bayes modeli, koşullu bağımsızlık ilkesine dayanır (Eyheralendy vd., 2003). Bu algoritma ile dokümanlarda kelimelerin olup olmamasına bakılarak ağırlıklandırma yapılır. Sözlükteki (kelimeler tablosu) her bir kelime doküman ile karşılaştırılarak dokümanın ağırlığı bulunur (kelimenin tekrar sayısı dikkate alınmaz, sadece dokümanda varsa 1 yoksa 0 olarak alınır). Doküman 1'ler ve 0'lardan oluşacak şekilde ifade edilir. 2 olasılıklı durumlarda binary classification (Multi variate model) kullanılabilir (Örneğin Spam mail ve spam olmayan mail bulunması).

Sonuçta sözlük boyutumuz kadar 1'ler ve 0'lardan oluşan vektörlerimizi elde ederiz.

V=(1,0,0,1,0,0,1,1,0)

Bu vektörün herhangi bir sınıfta olma olasılığını hesaplamak için aşağıdaki eşitlikler kullanılır.

d vektörünün cj kategorisinde olma olasılığı:

$$p(d|c_j) = \prod_{t=1}^{|V|} p(w_t|c_j)^{x_t}(1 - p(w_t|c_j))^{(1-x_t)} \tag{4.5}$$

$$p(w_t|c_j) = \frac{1 + B_{jt}}{2 + |c_j|} \tag{4.6}$$

|V| (Sözlükteki kelime sayısı)

Bjt (”cj” kategorisinde bulunan ve “wt” kelimesini içeren eğitim dokümanı sayısı)

|Cj| (“cj” sınıfında bulunan eğitim dokümanı sayısı)

xt (Kelimenin ağırlığı (1 veya 0))

$$M(C) = P(word1|C)^{n1} P(word2|C)^{n2} ... P(wordv|C)^{nv} P(|C) \quad (4.7)$$

M(C) değeri en büyük olan kategoriye doküman atanır.

Bu algoritmaya örnek olarak 2 olasılıklı durumları gösterebiliriz. Örnek olarak yapacağımız projede sistem yazılan mesajları arka plandaki eğitim dokümanları aracılığı ile yorumlayarak kategoriye karar verecektir. Olumlu ve olumsuz olarak 2 kategori kullanılacaktır. Eğitim dokümanı olarak kolay anlaşılması açısından 2’şer tane cümle ekleyerek örnek verilecektir. (Normalde sistemin sağlıklı eğitilebilmesi için uzun ve çok sayıda eğitim dokümanları kullanılmalıdır.)

Kelimeler (Sözlük)

Akıllı*

Aptal*

Evcil*

Güzel*

Pis*

Zeki*

Test Dokümanı

Sorgu-Sen çok akıllısın. (?)

Aşağıdaki gibi kelimenin dokümanda bulunması veya bulunmaması prensibi ile eğitim dokümanlarımızın aşağıdaki gibi ağırlıklandırıldığını düşünürsek:

D1(1,0,1,0,0,1) Olumlu

D2(0,0,0,1,0,0) Olumlu

D3(0,1,0,0,1,0) Olumsuz

D4(0,1,0,0,0,0) Olumsuz

olacaktır. Test dokümanımız ise:

DS (1,0,0,0,0,0)

(4.5), (4.6) ve (4.7) eşitlikleri yardımı ile aşağıdaki değerler hesaplanabilir.

P(Olumlu)=1/2

P(Olumsuz)=1/2

$$M(\text{Olumlu})=\frac{1}{2}*\frac{1}{2}*\frac{3}{4}*\frac{1}{2}*\frac{1}{2}*\frac{3}{4}*\frac{1}{2}=0{,}075$$

$$M(\text{Olumsuz}) = \frac{1}{2} * \frac{1}{4} * \frac{1}{4} * \frac{3}{4} * \frac{3}{4} * \frac{1}{2} * \frac{3}{4} = 0{,}0065$$

P(d|Olumlu)> P(d|Olumsuz) olduğu için olumlu bir kategori olarak yorumlanacaktır.

Şekil 4.2 Karakaçan projesi ekran görüntüsü

Olumlu bir cümle yazdığımızda olumlu olduğunu maskotumuzun gülümsemesi ile anlıyoruz.

Şekil 4.3 Karakaçan projesi ekran görüntüsü 2

Projeye www.ismailpilavcilar.info adresinden ulaşılabilir.

4.2.2 Naive Bayes Frekans Ağırlıklandırma (Multinominal Model)

Doküman içindeki kelimelerin tekrar sayılarını hesaplarımızda kullanmanın multi-variate naive bayes algoritmasına göre daha iyi çalıştığı bulunmuştur. Burada dikkat edilmesi gereken nokta her bir kelimenin tekrar etme sayısı diğer kelimelerin tekrar etme sayılarından bağımsızdır (Schneider, 2004). Sonuçta sözlük boyutumuz kadar, kelimelerin tekrar sayılarından oluşan vektörlerimizi elde ederiz.

(2,3,0,.....,10,100)

$$p(d|c_j) = p(|d|)|d|!\prod_{t=1}^{|V|}\frac{p(w_t|c_j)^{xt}}{x_t!} \tag{4.8}$$

$$p(w_t|c_j) = \frac{1+N_{jt}}{|V|+N_J} \tag{4.9}$$

d (Kategori Sayısı)

N_{jt} (j sınıfındaki dokümanlar içinde t kelimesinin görülme sıklığı)

N_j (j sınıfındaki toplam kelime sayısı)

P(|d|) (Kategori olasılığı)

x_t (kelimenin frekansı)

|V| (Kelime sayısı)

$$M(C) = P(word1|C)^{n1}P(word2|C)^{n2}...P(wordv|C)^{nv}P(|C) \tag{4.10}$$

M(C) değeri en büyük olan kategoriye doküman atanır.

Ağırlıklandırılmış vektörlere aşağıdaki eğitim doküman vektörlerini ve kategorilerini örnek gösterebiliriz. Bunlara göre aşağıda bir dokümanın kategorisinin Multinominal Model ile nasıl bulunabileceği anlatılmaktadır.

D1=(0,2,0,1,0,0) Sağlık

D2=(0,0,0,1,0,0) Sağlık

D3=(1,0,0,0,0,0) Ekonomi

D4=(0,0,0,0,0,2) Ekonomi

D5=(0,0,2,0,0,0) Spor

D6=(0,0,0,1,2,0) Spor

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

(4.8), (4.9) ve (4.10) eşitlikleri yardımı ile aşağıdaki değerler hesaplanabilir.

$$M(\text{Sağlık})= \frac{1}{3} * 3!* 1*1*\left(\frac{1+0}{6+4}\right)^2 \frac{1}{2!} *1*\left(\frac{1+0}{6+4}\right)\frac{1}{1!} *1=0.0010$$

$$M(\text{Ekonomi})= \frac{1}{3} * 3!* 1*1*\left(\frac{1+0}{6+3}\right)^2 \frac{1}{2!} *1*\left(\frac{1+0}{6+3}\right)\frac{1}{1!} *1=0.0013$$

$$M(\text{Spor})= \frac{1}{3} * 3!* 1*1*\left(\frac{1+2}{6+5}\right)^2 \frac{1}{2!} *1*\left(\frac{1+2}{6+5}\right)\frac{1}{1!} *1=0.02$$

M(Spor) en büyük olduğu için doküman 0.02 olasılıkla Spor kategorisindedir.

5. YÖNETİM PROGRAMI

Metin kategorizasyonunu yapabilmemizi sağlayan ve arka planda önceki konularda anlatılmış olan algoritmalarla çalışan yönetim programı 5 bölümden oluşmaktadır. Yönetim programı Visual Studio.NET 2005 ortamında Visual Basic dili kullanılarak geliştirilmiştir. Veritabanı olarak Sql Server 2000 kullanılmıştır.

1.Ana Bölüm

2.Sözlük

3.Eğitim Dokümanları

4.Dokümanlar

5.Kategori İşlemleri

5.1 Ana Bölüm

Şekil 5.1 Yönetim programı ana bölümü

Bu bölüm programın ilk arayüzünün görüntülendiği bölümdür. Buradan Sözlük, Eğitim Dokümanları, Dokümanlar (Sınıflandırılmış Dokümanlar), Kategoriler gibi alt bölümlere buradan ulaşılabilir.

5.2 Sözlük

Şekil 5.2 Yönetim programı sözlük bölümü

Bu bölümde veritabanımızda yer alan tüm kelimeler listelenir. Kelime ekleme, silme, güncelleme, joker ekleme gibi işlemler bu bölümden yapılabilir (4 kategori için 412 adet kelime kullanılmıştır).

Not: Kelime ekleme, güncelleme, joker ekleme gibi işlemler mevcut eğitim dokümanlarında değişikliğe sebep olacağından bu gibi işlemler sonrasında eğitim vektörleri (Config tablosunda bulunan) otomatik olarak güncellenir.

5.2.1 Kelimeler Tablosu

Bu tablodaki kelimelerimizi aslında vektör uzayımızdaki eksenler olarak düşünebiliriz. Bunun için aşağıda görülen alanlar tabloda tanımlanır. Spor, sağlık, ekonomi ve politika kategorilerinde geçebilecek olan kelimeler bu tabloya eklenmelidir. Id değerimiz tekil bir numarayı (primary key) ifade etmektedir. Bu alan kelimeler tablosunu jokerler tablosu ile birleştirmek için kullanılır.

	Column Name	Data Type	Length	Allow Nulls
🔑	Id	int	4	
	Kelime	nvarchar	50	

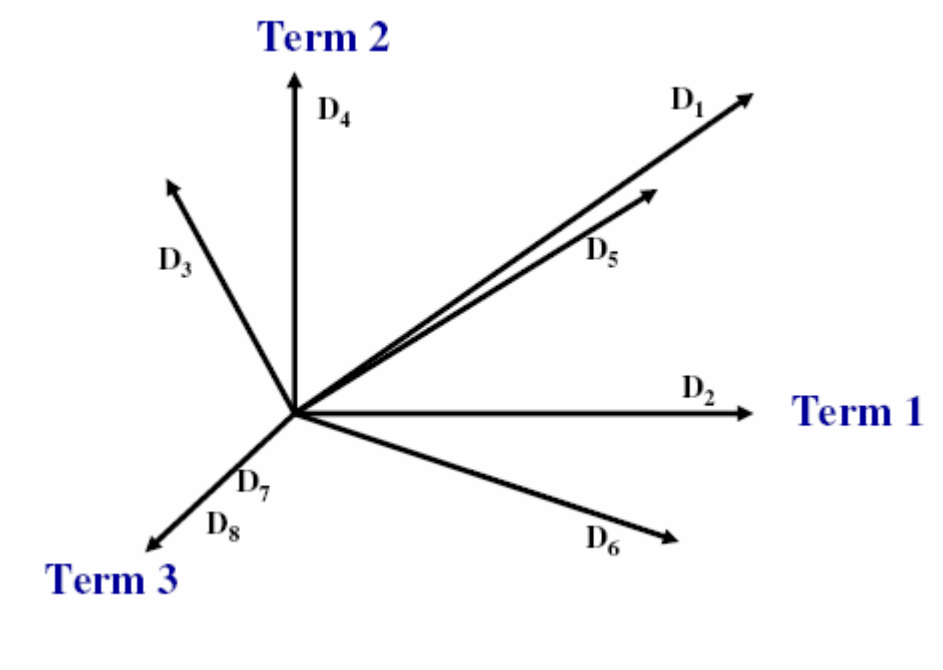
Şekil 5.3 Kelimeler tablosu

Örnek Kayıtlar:

813	alerji*
814	alıcı*
815	alkol*
816	altın*
817	alzheimer*
818	ambulans*
819	ameliyat*
820	anne*

821	antibiyoti*
822	antreman*
823	antrenör*
824	aslan*
825	asprin*
826	aşı
827	aşk*
828	ateş*
829	atletizm*
830	attı*
831	avantaj*
832	averaj*

Yaklaşık 450 tane kelime tabloya eklenmiştir.



Şekil 5.4 Terim uzayı

Term ile belirtilenler aslında bizim kelimelerimiz, D'ler ise eğitim dokümanlarımızdır.

5.2.2 Jokerler Tablosu

	Column Name	Data Type	Length	Allow Nulls
▶	KelimeId	int	4	
	Joker	varchar	50	

Şekil 5.5 Jokerler tablosu

Örnek Kayıtlar:

KelimeId	Joker
80	ilac*

(Buradaki 80 nolu kelimeId, kelimeler tablosu ile jokerler tablosunun birleştirilmesi için kullanılan bir anahtardır).

Eğer kelimeler tablosunda tutulan kelimelerin başka gövdeleri de varsa bu tabloda belirtilmelidir. Örneğin kelimeler tablosunda bulunan "ilaç*" kelimesi Türkçe kurallarına göre "ilac*" olarakta ifade edilebilir. Örneğin;

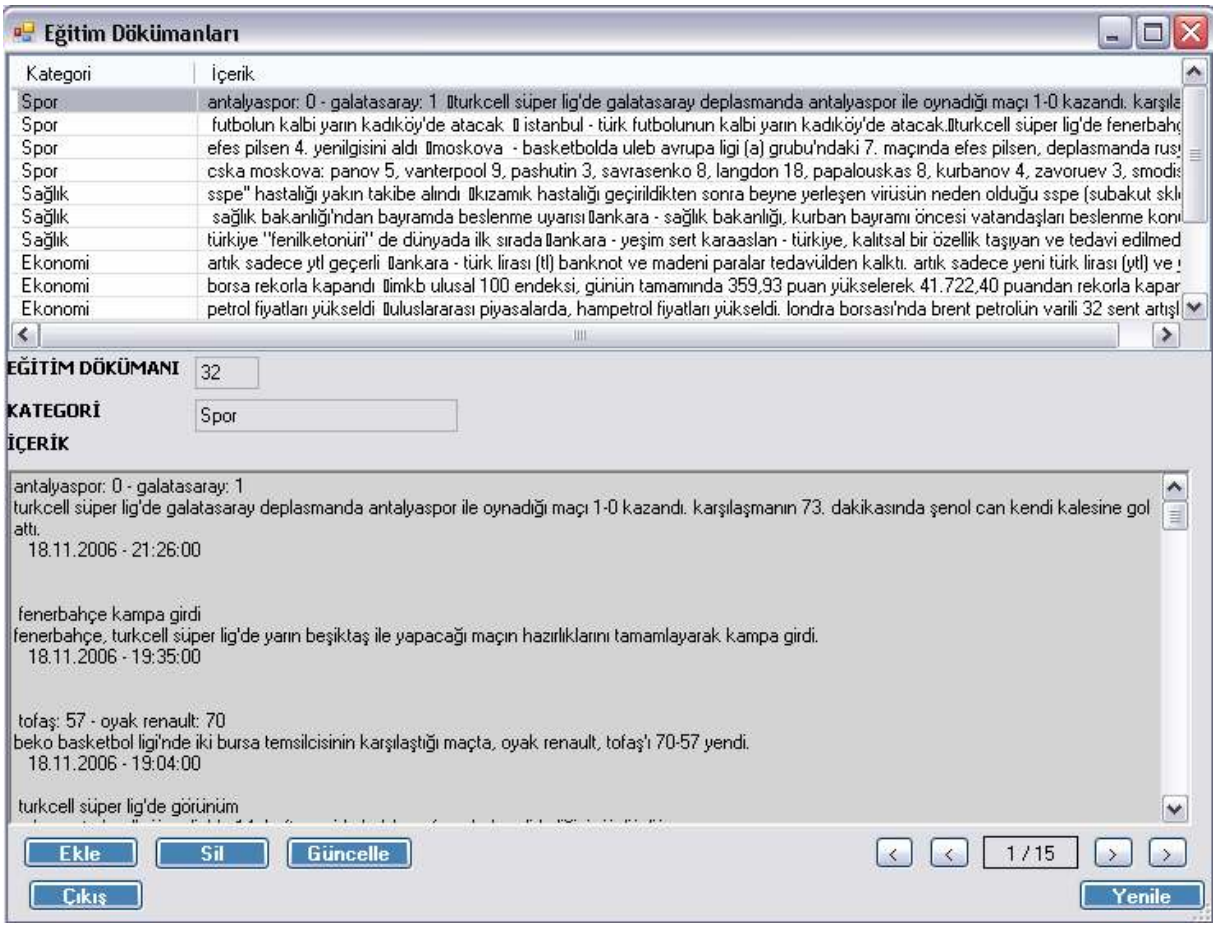
İlacını almalısın.

İlaçtaki kimyasal maddeler hemen kana karışır.

Bu iki cümlede asıl vurgulanmak istenen ilaç kelimesidir. Ancak görüldüğü gibi cümle içinde "ç" , "c" ye dönüşebilir. Bu tür durumlarda her iki kelimeyide ilaç olarak kabul edebilmek için jokerler tablosunda tüm olası durumlar yazılmalıdır.

Kelimeler tablosunda bulunan "ilaç*" kelimesi ile başında ilaç olan bir kelime görüldüğü anda vektördeki "ilaç" boyutunun ağırlığını 1 arttırmalıdır. Benzer şekilde jokerler tablosundan da diğer olasılıklar (ilac*) araştırılır. Uygun olanlar için ağırlıklar arttırılır.

5.3 Eğitim Dokümanları



Şekil 5.6 Yönetim programı eğitim dokümanları bölümü

Bu bölümde sistemimizi eğitmekte kullanacağımız eğitim dokümanları bulunmaktadır. Eğitim dokümanı ekleme, silme, güncelleme gibi işlemler bu bölüm aracılığı ile yapılabilir.

Not: Yeni bir eğitim dokümanı eklendiğinde, ya da mevcut olan bir eğitim dokümanı silindiğinde veya güncellendiğinde eğitim vektörleri ve ilişkili tabloları otomatik olarak güncellenir (Config ve Tekrar Sayıları tabloları).

5.3.1 Eğitim Dokümanları Tablosu

Bu tabloda eğitim dokümanları tutulmaktadır.

Column Name	Data Type	Length	Allow Nulls
DokumanIcerikId	int	4	
DokumanIcerikText	text	16	
KategoriId	int	4	

Şekil 5.7 Eğitim dokümanları tablosu

Örnek:

ARTIK SADECE YTL GEÇERLİ

ANKARA - Türk Lirası (TL) banknot ve madeni paralar tedavülden kalktı. Artık sadece Yeni Türk Lirası (YTL) ve Yeni Kuruşlar (YKr) kullanılacak. Yeni yıl ile birlikte artık bütün ödemeler sadece YTL ile yapılacak. Türk Lirası banknotlar 10 yıl, madeni paralar ise sadece 1 yıl boyunca Merkez Bankası ve Ziraat Bankası şubelerinde değiştirilebilecek. Ayrıca yine 1 Ocak 2006'dan itibaren ödeme emirleri, kıymetli evraklar, sözleşmeler ve diğer tüm belgeler sadece YTL üzerinden düzenlenecek.

İşyerleri 1 Ocak 2006'dan itibaren tarife ve etiketlerinde çift fiyat göstermede zorunlu olmayacak, fiyatların sadece YTL üzerinden gösterilmesi yeterli olacak.

Bu arada, sadece YTL'nin kullanılmasına başlanması ile birlikte, Bakanlar Kurulu tarafından belirlenecek bir tarihte, YTL'deki "yeni" ibaresi kaldırılacak. YTL'deki "Yeni" ibaresinin kaldırılması sonucunda Türkiye tekrar "TL" ye geçmiş olacak.

DOĞALGAZA OCAK AYINDA ZAM YOK

Doğalgaza ocak ayında zam yapılmayacağı, 2005 yılı aralık ayı fiyatlarıyla devam edileceği bildirildi. BOTAŞ Yönetim Kurulu'nun aldığı karar uyarınca, BOTAŞ'ın doğalgaz metreküp fiyatı dağıtım kuruluşları için 0.335699 YTL olarak devam edecek.

5.3.2 Config Tablosu

Bu tabloda eğitim dokümanlarımızın vektörel koordinatları tutulmaktadır. Bu koordinatlar ön işleme aşamasında belirtilen prensiplere uygun olarak belirlenir.

	Column Name	Data Type	Length	Allow Nulls
▶	DokumanId	char	10	
	vektor	int	4	✓
	sira	int	4	
	tfidf	float	8	✓

Şekil 5.8 Config (ayarlar) tablosu

DokumanId dokümanımızın anahtar değeridir (Dokumanlar tablosu ile bu anahtar değer üzerinden bağlıdır).

Örnek olarak;

DokId	vektör	sıra	tfidf
32	0	1	0

32 2 2 2,34

32 0 3 0

32 1 4 0

Sıra değeri sözlükteki sıraya göre kelimeleri ifade eder. Vektör değeri ise sıra alanındaki kelimenin doküman içinde kaç defa geçtiği bilgisini tutar. Tf-idf değeri ise kelimelerin tf-idf kuralına göre ağırlıklarını gösterir.

5.3.3 Tekrar Sayıları Tablosu

	Column Name	Data Type	Length	Allow Nulls
▶	kategoriadi	varchar	50	✓
	sira	int	4	✓
	sayi	int	4	✓

Şekil 5.9 Tekrar sayıları tablosu

Bu tabloda tüm kelimelerin tüm kategorilerde kaç defa geçtiği bilgisi tutulur.

Örnek;

Spor 1 0 1.kelimeden Spor eğitim dokümanlarında 0 tane varmış.

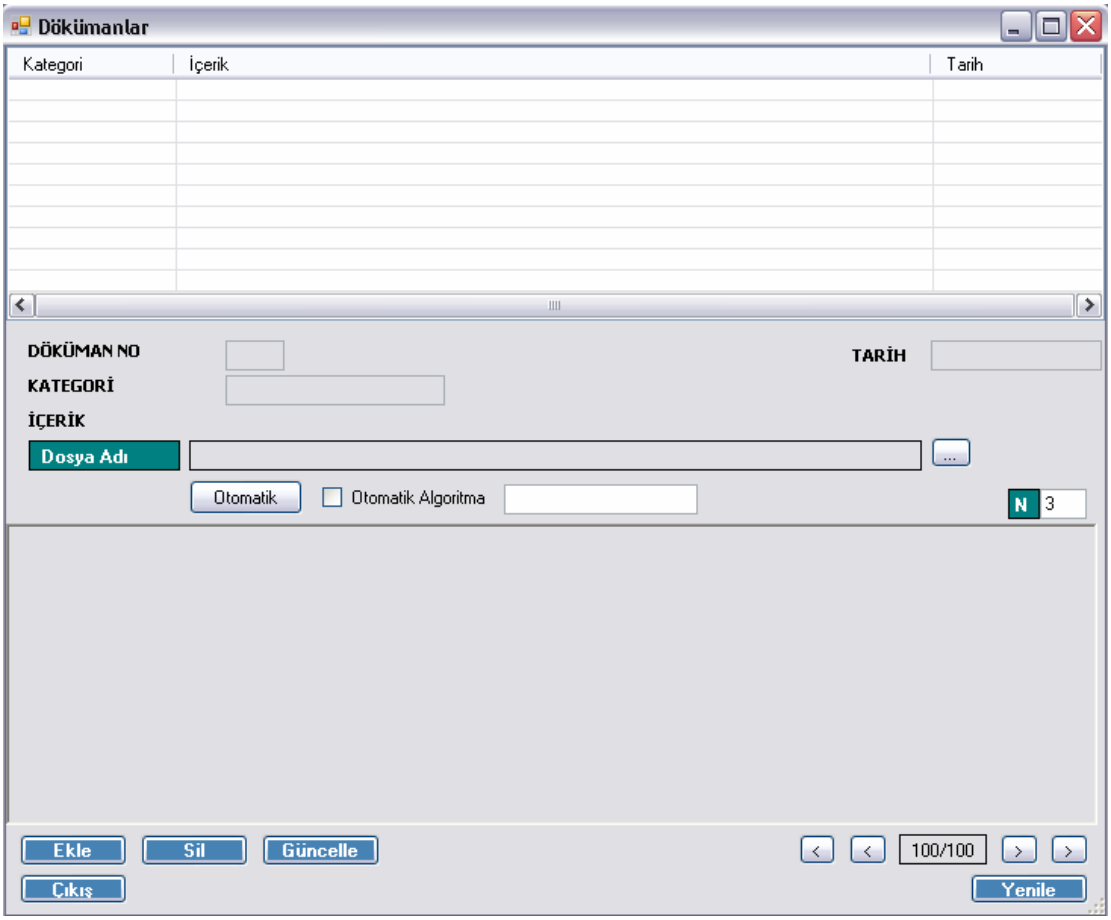
Spor 2 3 2.kelimeden Spor eğitim dokümanlarında 3 tane varmış.

Spor 3 0 3.kelimeden Spor eğitim dokümanlarında 0 tane varmış.

Spor 4 1 4.kelimeden Spor eğitim dokümanlarında 1 tane varmış.

Spor 5 0 5.kelimeden Spor eğitim dokümanlarında 0 tane varmış.

5.4 Dokümanlar



Şekil 5.10 Yönetim programı dokümanlar bölümü

Bu bölümde sınıflandırılan dokümanlara ailt bilgiler tutulmaktadır. Aynı zamanda sınıflandırılmak istenen doküman bu bölümden istenen algoritmaya göre sınıflandırılır.

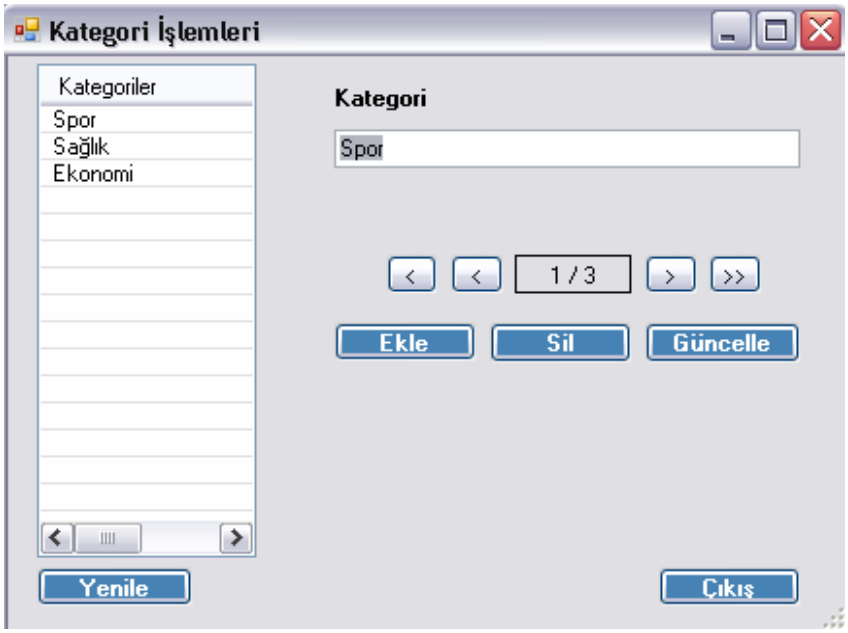
5.4.1 Dokümanlar Tablosu

	Column Name	Data Type	Length	Allow Nulls
🔑	DokumanId	int	4	
	DokumanIcerik	text	16	
	KategoriId	int	4	
	Tarih	datetime	8	

Şekil 5.11 Dokümanlar tablosu

Bu tabloda ise sınıflandırdığımız bir dokümanın içeriği, kategori bilgisi ve sınıflandırma yaptığımız tarihi tutulur.

5.5 Kategoriler



Şekil 5.12 Yönetim programı kategori işlemleri bölümü

Kategori silme, ekleme gibi işlemler bu bölümden yapılır.

5.5.1 Kategoriler Tablosu

Bu tablomuzda kategori isimleri tutulmuştur. Sınıflandırma işlemimiz bu kategoriler ile yapılmıştır (Spor, Ekonomi, Sağlık).

	Column Name	Data Type	Length	Allow Nulls
🔑	KategoriId	int	4	
	KategoriAdi	varchar	50	

Şekil 5.13 Kategoriler tablosu

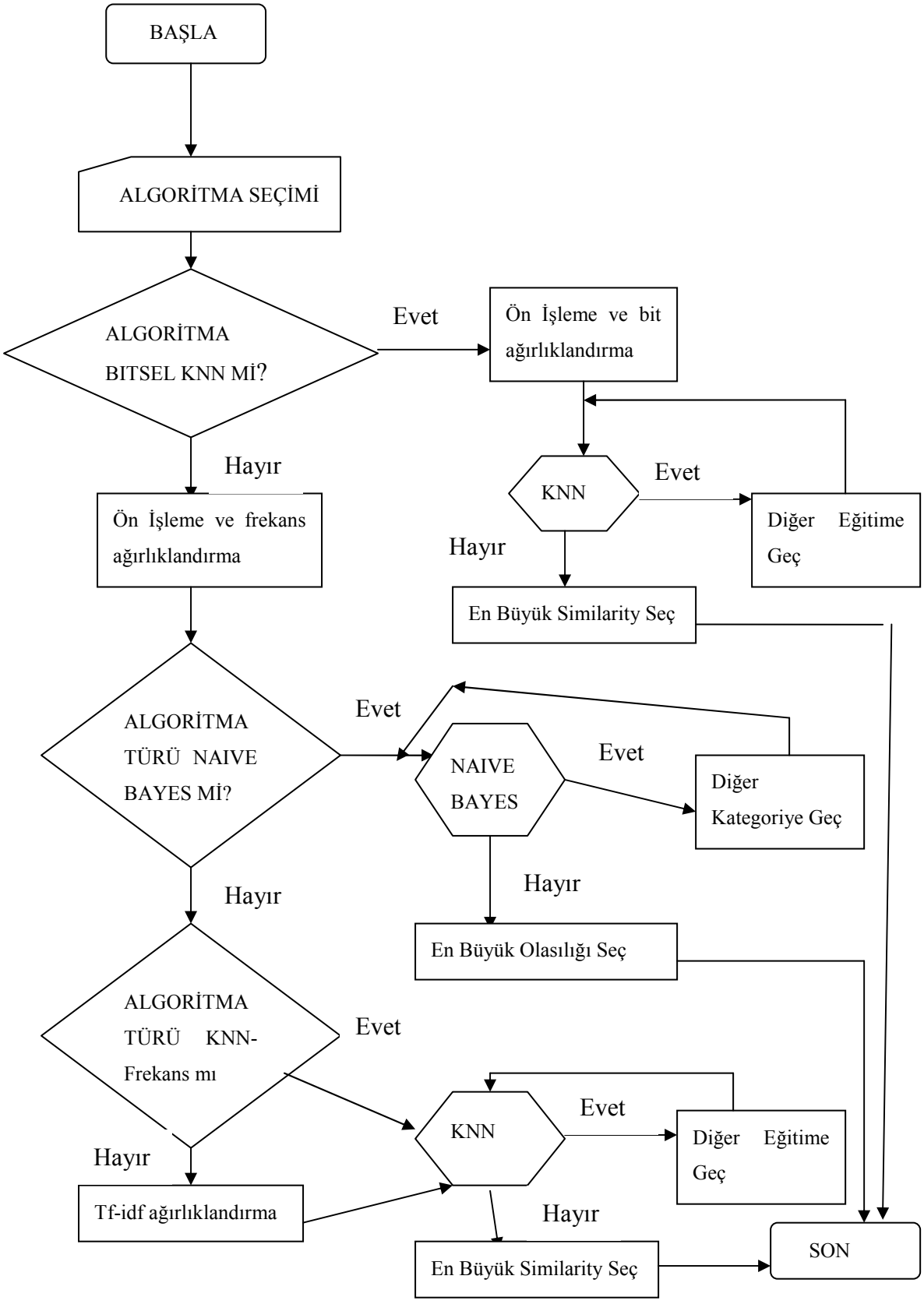
Örnek Kayıtlar:

1 Spor

7 Ekonomi

3 Sağlık

5.6 Akış Diyagramı



Şekil 5.14 Yönetim programı akış diyagramı

6. UYGULAMALAR

Bu bölümde metin kategorizasyonunda kullanılan veritabanı hakkında bilgiler verilmiş olup tablolar bölümünde yapılmış olan testler ve bunların sonuçları hakkında yorumlara yer verilmiştir.

6.1 Veritabanı

Metin kategorizasyonunda kullanılan veritabanı anadolu ajansı (www.anadoluajansi.com) haber sitesinden 2006 ve 2007 tarihli 2600 makale kullanılarak oluşturulmuştur. Bu makalelerden 440 tanesi ekonomi, 470 tanesi politika, 320 tanesi sağlık, 1370 tanesi spor kategorilerinden seçilmiştir. Bu makaleler sistemin eğitilmesinde ve test aşamasında kullanılmıştır (Çizelge 6.1).

Çizelge 6.1 Kullanılan veritabanı ve makale sayısı

Kategoriler	**Toplam Makale Sayısı**
Spor	**1370**
Politika	**470**
Ekonomi	**440**
Sağlık	**320**
Toplam	**2600**

6.2 Tablolar

Tezde ilk olarak örnek tabanlı bir öğrenme yöntemi olan K-NN (k en yakın komşuluk) algoritması çeşitli k değerleri için veri tabanına uygulanmıştır. En başarılı k sayısı testler sonucu bulunmuş ve bu k sayısını kullanarak yapılmış olan test sonuçlarına tez içerisinde yer verilmiştir.

Bu yöntemde test örneği ile eğitim örneği arasındaki benzerlik hesaplanır ve girdi verilerinin kategorilerini anlaşılabilmesi için en yüksek derecede yakınlığa sahip k adet örnek dikkate alınarak en uygun kategori bulunur. K-NN algoritmasına üç çeşit terim ağırlıklandırma yöntemi kullanılmıştır. Bunlar, bit ağırlıklandırma, frekans ağırlıklandırma ve tf-idf ağırlıklandırmadır. Bit ağırlıklandırma yönteminde doküman vektörünün ağırlıkları kelimenin dokümanda bulunma veya bulunmaması durumuna göre belirlenir. Frekans ağırlıklandırma yönteminde ise doküman vektörünün ağırlıkları kelimenin dokümanda bulunma sayısına göre belirlenir. Kullanılan son ağırlıklandırma yöntemi olan tf-idf'de ise terim ağırlıkları normalize edilir. Böylece uzun dokümanlardaki gereksiz ağırlıklandırma önlenmiş olur. Bu yöntemde dokümanın uzunluğuna bağlı olarak terimlere verilen önem belirlenir. K değerlerine göre K-NN algoritmalarının başarıları Çizelge 6.2'de gösterilmiştir. K=3 için en başarılı sonuçlar elde edilmiştir.

Çizelge 6.2 K değerlerine göre K-NN algoritması başarıları

K-NN	Ortalama K=1	Ortalama K=2	Ortalama K=3	Ortalama K=4
K-NN Bit	1833/2080 %88.12	1837/2080 %88.3	1860/2080 %89.42	1859/2080 %89.37
K-NN Frekans	1802/2080 %86.6	1810/2080 %87.01	1814/2080 %87.21	1814/2080 %87.21
K-NN TF.IDF	1856/2080 %89.2	1866/2080 %89.7	1880/2080 %90.38	1877/2080 %90.2

6.2.1 Çapraz Doğrulama (Cross Validation)

Çapraz Doğrulama (Cross-Validation), diğer bir adıyla dönüşümlü tahmin, bir veri kümesinin alt kümelere bölünerek ilk analizin tek bir alt kümede yapıldığı ve bu esnada diğer alt kümelerin sonradan yapılacak olan ilk analizi doğrulamak amacıyla tutulduğu istatistiksel bir uygulamadır.

Başlangıç olarak kullanılan veri alt kümesini eğitim kümesi (training set), ve diğer alt kümelerde doğrulama (validation) veya test kümesi (testing set) olarak adlandırılır.

Seymour Geisser tarafından ortaya atılan Çapraz Doğrulama teorisi daha önce kullanılmakta olan test hipotezlerine, özellikle maliyeti yüksek olanlara karşı önemli bir yer edinmiştir.

6.2.2 K-defa Çapraz Doğrulama

K-defa (fold) çapraz doğrulama'da orjinal örnek K adet alt örneklere parçalanmıştır. Bu K adet alt örnekten bir alt örnek, modeli test etmek için doğrulama verisi olarak ayrılır ve geri kalan K-1 adet alt örnek de eğitim verisi olarak kullanılır. Çapraz doğrulama işlemi daha sonra, doğrulama verisi olarak bir kere kullanılmış olan K adet alt örnekle, K defa daha tekrarlanır. Bu tekrarlamalardan çıkan K adet sonucun, tek bir tahmin yürütmek amacıyla ortalaması alınabilir (veya birleştirilebilir).

Bu tezde, haber makaleleri her bir kategori bazında 5 eşit parçaya bölünmüştür. Her bir parça ayrı ayrı sistemi eğitmekte kullanılmış, diğer kalan parçalar ise test aşaması için kullanılmıştır. Örneğin kategorilerin ilk parçalarıyla sistem eğitilirken diğer parçalarla test yapılmıştır. İkinci parça eğitimde kullanıldığında ise birinci parça, diğer parçalarla birlikte tekrar test aşamasına dahil edilmiştir. Bu işlem son parçaya kadar bu şekilde devam etmektedir.

K-NN ve Naive Bayes algoritmaları ile yapılan çapraz doğrulama test sonuçları Çizelge 6.3, Çizelge 6.5, Çizelge 6.7 ve Çizelge 6.9'da gösterilmiştir.

Sınıflandırma sonucu kategorilerin dağılımını görebilmek için hata matrisi (confusion matrix) kullanılmıştır. Hata matrisinde, satırlar ve sütunlarla gösterilen sınıflandırma sonuçlarında;

satırlar sınıf verilerini, sütunlar da örnek noktaya dayalı yer gerçeklerini ifade eder. Tüm algoritmalara ait hata matrisleri Çizelge 6.4, Çizelge 6.6, Çizelge 6.8 ve Çizelge 6.10'da gösterilmiştir.

Çizelge 6.3 K-NN frekans algoritmasının kategori bazında başarısı (K=3 için)

K-NN Frekans	F1	F2	F3	F4	F5	Ortalama
Politika	344/376 %91.4	315/376 %83.7	335/376 %89.09	327/376 %86.9	319/376 %84.84	1640/1880 %87.2
Ekonomi	260/352 %74	282/352 %80.1	256/352 %73	275/352 %78.1	295/352 %83.8	1368/1760 %77.7
Sağlık	212/256 %82.8	223/256 %87.1	230/256 %89.8	226/256 %88.2	215/256 %82.8	1106/1280 %86
Spor	998/1096 %91	1028/1096 %93.8	1009/1096 %92.06	1037/1096 %94.61	1007/1096 %91.8	5079/5480 %92.6

Çizelge 6.4 K-NN frekans algoritmasının hata matrisi

	Politika	Ekonomi	Sağlık	Spor
Politika	1640	135	3	0
Ekonomi	178	1368	1	0
Sağlık	3	1	1106	52
Spor	10	2	35	5079

Çizelge 6.5 K-NN bit algoritmasının kategori bazında başarısı (K=3 için)

K-NN Bit	F1	F2	F3	F4	F5	Ortalama
Politika	342/376 %90.9	342/376 %90.9	337/376 %89.6	327/376 %86.9	327/376 %86.9	1675/1880 %89
Ekonomi	278/352 %78.9	269/352 %76.4	265/352 %75.2	281/352 %80	286/352 %81.25	1379/1760 %78.3
Sağlık	223/256 %87.1	229/256 %89.4	227/256 %88.6	231/256 %90.2	221/256 %86.3	1131/1280 %88.3
Spor	1017/1096 %92.7	1044/1096 %95.2	1046/1096 %95.4	1032/1096 %94.1	1058/1096 %96.5	5197/5480 94.8%

Çizelge 6.6 K-NN bit algoritması hata matrisi

	Politika	Ekonomi	Sağlık	Spor
Politika	1675	130	2	0
Ekonomi	201	1379	4	0
Sağlık	3	1	1131	62
Spor	7	2	33	5197

Çizelge 6.7 K-NN tf-idf algoritmasının kategori bazında başarısı (K=3 için)

K-NN Tf-Idf	F1	F2	F3	F4	F5	Ortalama
Politika	341/376 %90.6	331/376 %88	335/376 %89.09	320/376 %85.1	324/376 %86.1	1651/1880 %87.8
Ekonomi	286/352 %81.2	289/352 %82.1	246/352 %70	291/352 %82.6	293/352 %83.2	1405/1760 %80
Sağlık	207/256 %80.8	223/256 %87.1	228/256 %89.06	228/256 %89.06	216/256 %84.3	1102/1280 %86
Spor	1046/1096 %95.4	1058/1096 %96.5	1052/1096 %95.9	1052/1096 %95.9	1055/1096 %96.2	5260/5480 %96.04

Çizelge 6.8 K-NN tf-idf algoritması hata matrisi

	Politika	Ekonomi	Sağlık	Spor
Politika	1651	151	4	0
Ekonomi	198	1405	3	0
Sağlık	5	2	1102	65
Spor	8	1	38	5260

Daha sonra tezde multivariate ve multinominal Naive Bayes algoritmaları kullanılmıştır. Multivariate Naive Bayes algoritmasında, kelimelerin dokümanda bulunması ya da bulunmaması önemlidir. Kelimenin dokümanda bulunma sayısı kullanılmaz. Böylece doküman, 0 ve 1'lerin olduğu bir vektör ile gösterilir duruma gelir. Bu yöntem belli bazı durumlarda (genellikle 2 kategorinin bulunduğu durumlar) iyi sonuç vermesine karşın çoğu durumda kelimenin dokümanda bulunma sıklığını dikkate alarak yapılan hesaplamaların daha iyi sonuç verdiği gözlenmiştir. Bu durumda kullanılan yöntem ise multinominal Naive Bayes algoritmasıdır. Sonuçlar Çizelge 6.9'da gösterilmiştir.

Çizelge 6.9 Naive bayes algoritmasının kategori bazında başarısı (multinominal)

Naive Bayes	F1	F2	F3	F4	F5	Ortalama
Politika	351/376 %93.35	346/376 %92.02	342/376 %90.9	334/376 %88.82	338/376 %89.89	1711/1880 %91.01
Ekonomi	290/352 %82.3	285/352 %81	290/352 %82.3	293/352 %83.2	299/352 %85	1457/1760 %82.7
Sağlık	225/256 %87.89	235/256 %91.79	236/256 %92.18	239/256 %93.35	227/256 %88.67	1162/1280 %90.7
Spor	1051/1096 %95.8	1067/1096 %97.3	1058/1096 %96.5	1062/1096 %96.89	1066/1096 %97.2	5304/5480 %96.7

Çizelge 6.10 Naive Bayes algoritması hata matrisi

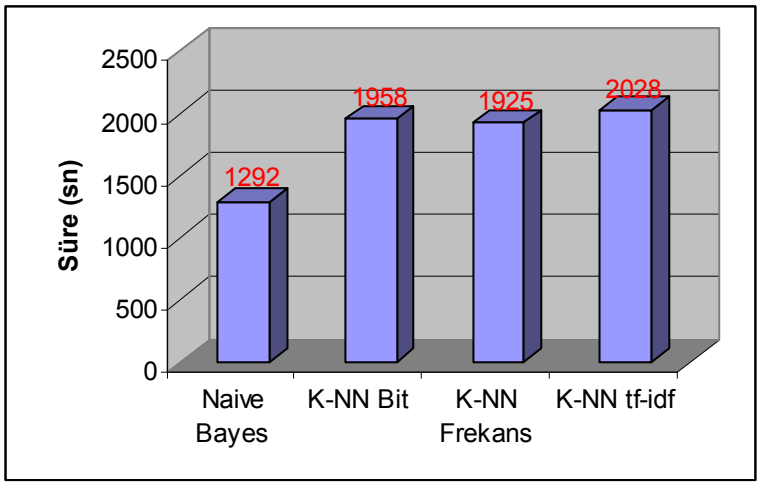
	Politika	Ekonomi	Sağlık	Spor
Politika	1711	140	2	0
Ekonomi	156	1457	4	0
Sağlık	4	6	1162	48
Spor	7	2	44	5304

Hata matrislerinden de görüldüğü gibi sınıflandırıcı bazı durumlarda haberleri benzerlik göstermesinden dolayı birbirine atamıştır. Örneğin Çizelge 6.10'a bakıldığında ekonomi haberlerinin 156 tanesi politika, politika haberlerinin ise 140 tanesi ekonomi olarak yorumlanmıştır.

Çizelge 6.11 Algoritmaların dizin bazında başarıları (k=3) ve standart sapma (SS) değerleri

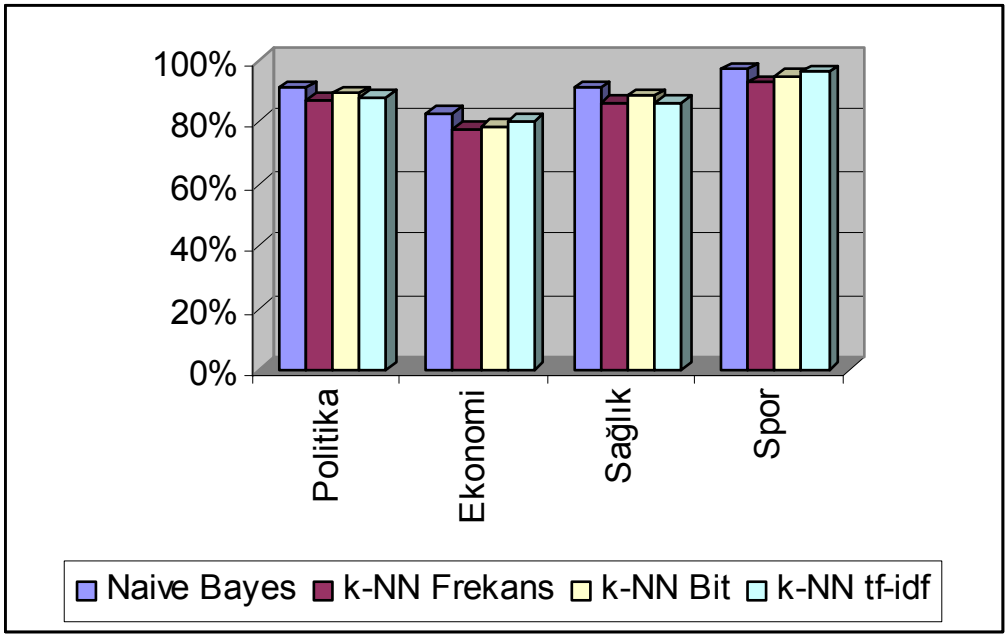
Algoritma	F1	F2	F3	F4	F5	Ortalama	Eğitim(sn)	Test(sn)	SS
Naive Bayes	%92.16	%92.9	%92.59	%92.69	%92.78	%92.62	230	1062	0.2535
K-NN Bit	%89.42	%90.57	%90.14	%89.95	%90.56	%90.13	230	1728	0.4278
K-NN Frekans	%87.21	%88.84	%87.98	%89.66	%88.26	%88.39	230	1695	0.8237
K-NN Tf-idf	%90.38	%91.39	%89.47	%90.9	%90.76	%90.58	230	1798	0.6423

Çizelge 6.11' den görüldüğü gibi en başarılı algoritma %92.6 doğrulukla Naive Bayes olarak bulunmuştur. Naive Bayes doğruluk bakımından metin dokümanlarını sınıflandırmak için kullanılan en başarılı algoritmalardan biridir, ayrıca uygulanabilirliği de kolay ve performansı yüksek olan bir algoritmadır. Yapılan testlerde K-NN algoritmasının tf-idf ağırlıklandırması %90.58 yüzdeyle en başarılı 2.algoritma olmuştur.



Şekil 6.1 Algoritmalar ve çalışma süreleri

Algoritmanın çalışma süresi bakımından karşılaştırma yapılırsa yine Naive Nayes algoritması daha performanslı bulunmuştur. Naive Bayes algoritması olasılık hesabı yaparken K-NN algoritması gibi tüm eğitim dokümanlarına olan cosinus benzerliğini hesaplamaz, bu yüzden daha hızlı çalışmıştır. (Şekil 6.1)



Şekil 6.2 Algoritma başarıları

Hata matrisleri incelendiğinde Spor kategorisi en başarılı sonuç veren kategori olarak dikkat çekmektedir (Şekil 6.2). Bunun sebebi spor kategorisinde kullanılan kelimelerin fazla çeşitlilik göstermeyen kelimeler olması olarak yorumlanabilir. Ayrıca spor kategorisinde kullanılan eğitim dokümanlarının kelime sayısı olarak daha fazla olması, başarıyı arttıran nedenlerden biri olarak gösterilebilir. Ekonomi ve politika kategorilerinde ise başarı Spor kategorisine oranla daha az olmuştur. Çünkü bu kategorilerde kullanılan makalelerdeki kelimeler her iki kategori için de benzerlik göstermektedir. Örneğin aşağıdaki makalenin asıl kategorisi ekonomi olmasına karşın “parti”, “iktidar”, “bakan” ,”basbakan” gibi kelimelerden dolayı politika kategorisine atanmıştır.

SENER:"AK PARTI IKTIDARI ILE TÜRK PARASI TEKRAR (PARA) OLMUSTUR"

Devlet Bakani ve Basbakan Yardimcisi Abdüllatif Sener, "AK Parti iktidari ile Türk parasi tekrar (para) olmustur" dedi. Sener, ülkenin bugününü ve yarinini güçlü kilmak için ne yapmak lazimsa onu yaptiklarini söyledi.

08.04.2006 - 15:44

Eğitim dokümanları sayısı iyi bir sınıflandırma için önemli bir kriterdir. Eğitim dokümanlarını geniş tutmak verimli bir sınıflandırma için olumlu bir yaklaşım olsa da çok büyük boyuttaki eğitim dokümanları performansı olumsuz yönde etkileyerek başarıyı düşürmektedir. Çizelge 6.12 eğitim dokümanlarının sayılarına göre başarı yüzdelerini göstermektedir. Görüldüğü gibi eğitim dökümanlarının sayısının artmasıyla başarıda artmaktadır.

Çizelge 6.12 Algoritmaların eğitim dökümanları sayısına göre başarıları

Algoritma	100 Makalelik Eğitim Dök.	200 Makalelik Eğitim Dök.	400 Makalelik Eğitim Dök.	600 Makalelik Eğitim Dök.
Naive Bayes	1817/2080 %87.35	1840/2080 %88.46	1880/2080 %90.38	1917/2080 %92.16
K-NN Bit	1779/2080 %85.52	1803/2080 %86.68	1835/2080 %88.22	1860/2080 %89.42
K-NN Frekans	1728/2080 %83.07	1758/2080 %84.51	1796/2080 %86.34	1814/2080 %87.21
K-NN tf-idf	1750/2080 %84.13	1769/2080 %85.04	1821/2080 %87.54	1880/2080 %90.38

Özetlemek gerekirse Naive Bayes algoritmasının Türkçe haber metinlerinde yaklaşık %2 farkla K-NN (tf-idf) algoritmasına göre daha başarılı olduğunu görmekteyiz. Testler sonrası algoritmalar arası performans süreleri karşılaştırıldığında ise yine Naive Bayes algoritmasının K-NN algoritmasından daha başarılı olduğu ortaya çıkmıştır. K-NN algoritması eğitim verilerinin artması durumunda başarılı sonuç veren bir algoritmadır. Eğitim verilerinin sayısının fazla olmaması durumunda bu algoritma başarılı sonuç vermeyebilir. Ancak eğitim verilerinin fazla olması bu algoritmanın çalışma hızını olumsuz olarak etkileyecektir. Çünkü

K-NN algoritmasında kategorisi bulunması istenen doküman vektörünün tüm eğitim doküman vektörlerine olan kosinüs benzeşim (cosinüs similarty) değeri bulunup bu değerler üzerinden yorumlar yapılmaktadır.

6.2.3 Hatalı Sonuçlardan Örnekler

Yapılan testler sonucu hatalı olarak sınıflandırılan bazı haber metinlerine aşağıda örnekler verilmiştir.

Brezilyali futbol efsanesi Pele'nin gayri mesru kizi Sandra Arantes do Nascimento öldü. Brezilya'nin liman kenti Santos'daki Beneficencia Portuguese hastanesi, 42 yasindaki Nascimanto'nun meme kanserinden öldügünü bildirdi.

17.10.2006 - 23:56

Kelimeler : futbol , hastane, meme , kanserinden

Dokümanın asıl kategorisi Spor olmasına rağmen algoritma sonucu Sağlık kategorisine atanmıştır.

VESTEL MANISASPOR'UN SOFÖRÜ VEFAT ETTI

Vestel Manisaspor'un emektar soförü Muhsin Perköz (48), tedavi gördügü hastanede vefat etti.

20.10.2006 - 23:52

Kelimeler : tedavi ,hastane

Dokümanın asıl kategorisi Spor olmasına rağmen algoritma sonucu Sağlık kategorisine

atanmıştır.

CHRISTOPH DAUM BOGAZINDAN AMELIYAT OLDU

Fenerbahçe'nin eski teknik direktörü Christoph Daum'un bogazinda olusan yaralar nedeniyle ameliyat oldugu bildirildi. Kölner Express gazetesi Alman teknik adamin son günlerde artan bogaz agrilari ve olusan yaralar nedeniyle ameliyat masasina yattigini belirterek, Daum'un ancak hafta sonu hastaneden taburcu olabilecegini yazdi.

Daum'un son günlerde konusamaz hale geldigi belirtilen haberde, 53 yasindaki tecrübeli çalistiricinin, "Bogazimdaki iki yara alindi. Simdi antibiyotik kullaniyorum. Çok agrilarim vardi. Yine de simdilik fazla konusamiyorum. Çünkü zorlaniyorum" dedigi ifade edildi.

Gazete ayrica Daum'un "FC Köln ile görüsmedim. Herhangi bir pazarligimiz olmadi. Önce sagligima tekrar kavusmaliyim" dedigini yazdi.

07.11.2006 - 23:18

Kelimeler : bogaz, ameliyat, teknik, bogaz, ameliyat, teknik, bogaz, agrıları, ameliyat, hafta, hastane, antibiyotik, agri, saglık

Dokümanın asıl kategorisi Spor olmasına rağmen algoritma sonucu Sağlık kategorisine atanmıştır.

7. SONUÇLAR

Bu bölümde K-NN ve Naive Bayes algoritmaları ile bu algoritmalarda kullanılmış ağırlıklandırma metodlarının gerçek bir veritabanı üzerindeki deneysel değerlendirmelerine yer verilmiştir. Veritabanı olarak Anadolu Ajansı web sitesinin 2006-2007 yıllarına ait 2610 adet makalesi kullanılmıştır. Bu algoritmaların etkinliği, Anadolu Ajansı haberlerine ait veritabanı kullanılarak karşılaştırılmıştır. Ayrıca gelecekteki çalışmalar için öneriler getirilmeye çalışılmıştır.

Tez içerisinde yapılan kategorizasyon programı uygulaması, VisualStudio.Net 2005 ortamı kullanılarak Visual Basic dilinde yazılmış ve Anadolu Ajansı veritabanı Sql Server 2000 ortamında oluşturulmuştur. Uygulamada dört adet kategoriye yer verilmiş olmasına rağmen, arayüz programında yapılacak basit işlemlerle (kategori, kelime ve eğitim dokümanları ilavesi) ilgili uygulamanın daha çok kategori için çalışması sağlanabilir.

Tez içerisinde gerçekleştirilen kategorizasyon programı uygulaması göstermiştir ki otomatik sınıflandırma, iyi seçilmiş ve ön işleme tabi tutularak tüm gürültülü verilerden arındırılmış eğitim dokümanları ve yine iyi seçilmiş anahtar kelimeler yardımı ile gerçekten başarılı sonuçlar üretmektedir.

Uygulama esnasında yapılan test sonuçlarından da görüldüğü gibi en başarılı algoritma yüksek doğruluk oranıyla Naive Bayes olarak bulunmuştur. Naive Bayes doğruluk bakımdan metin dokümanlarını sınıflandırmak için kullanılan en başarılı algoritmalardan biridir. Naive Bayes algoritması uygulanabilirliğinin kolay ve performansının yüksek olması ile dikkat çekmektedir. Doğruluk yüzdesi bakımından, Naive Bayes algoritmasını K-NN algoritmasının tf-idf ağırlıklandırma yöntemi izlemektedir. Çalışma süreleri performansı açısından karşılaştırıldığında yine Naive Bayes Algoritmasının K-NN algoritmasına göre daha etkin olduğu görülmektedir. Bunun nedeni Naive Bayes algoritmasının olasılık hesabı yaparken kategorisi bulunacak olan vektörün tüm eğitim vektörleri ile aralarındaki kosinüs benzerliğini hesaplamasına gerek duymamasıdır.

K-NN algoritmasına ait tf-idf ağırlıklandırma yönteminin doğruluk oranı, K-NN algoritmasına ait diğer ağırlıklandırma yöntemlerine göre daha yüksektir. Bunun nedeni ise, tf-idf ağırlıklandırma yönteminin, kategorisi bulunacak olan dokümanda çok bulunan kelimeler için yüksek, ancak tüm dokümanlarda da sık geçen kelimeler için (stop words)

düşük sonuç vermesi ve buna bağlı olarak daha hassas ve doğru sonuç üretmesi olarak yorumlanabilir.

Tezde dört farklı kategori kullanılmıştır. Bunlar içerisinden spor kategorisinde diğer kategorilerde bulunmayacak türde, yani spor kategorisini fazlasıyla belirleyici kelimeler yer almaktadır. Diğer kategorilerden farklı kelimelerin çokça bulunması, spor kategorisini doğru sonuç verme konusunda daha performanslı hale getirmiştir.

Ekonomi ve politika kategorilerinde ise doğruluk performansı spor kategorisine göre daha az çıkmıştır. Bunun nedeni bu iki kategorinin içerik olarak benzer kelimeler bulundurması olasılığının yüksek olmasıdır.

Kısacası, Naive Bayes algoritması genel olarak K-NN algoritmasına göre üstün olmakla birlikte söylenebilir ki, K-NN algoritması "en yakın komşuluk" prensibini kullandığı için eğitim verisi sayısının artmasıyla başarı oranı da artmaktadır. Ancak bilimektedir ki eğitim verisi sayısı arttıkça da kullanılan algoritmalar da hız performansı bakımından sorunlar yaşanabilmektedir.

Algoritmaların başarısını arttırmak için eğitim dokümanlarının sayısı arttırılabilir, kategori belirtmeyen kelimeler (stop words) kullanılabilir, indeksleme algoritmaları ile performans bakımından iyileştirmeler yapılabilir. Bunun dışında literatürdeki farklı kelime sayısının büyüklüğü, kelime seçme algoritmaları (filtreleme) üzerinde odaklanmaya doğru bir eğilim oluşturmuştur. Filtreleme ile seçilen kelimeler kategorizasyonda yüksek başarıyı sağlayabilecek türdeki kelimelerdir. Filtreleme yapılırken, sıkça görülen kelimeler, kategoriye özel olup dokümanlarda görülen kelimeler, kategoriye özel olmayan ve dokümanlarda görülen kelimeler, dokümanlarda görülmeyen kelimelere dikkat edilir.

Gelecekte otomatik sınıflandırmanın arama motoru desteğine sahip siteler içinde yararlı olacağı kesindir. Bir sorgu sonucunda, arama motorları bilgi alışı ve kategorizasyon yeteneklerini birleştirerek anlamlı url'ler içeren en uygun kategorileri raporlayabilir. Bu kullanıcıların günümüzde yaşamış oldukları problemleri de azaltacaktır.

Dijital ortamda metin dokümanlarının sayıca artmasına bağlı olarak bir sınıflandırıcı için yeni veri geldikçe kendini güncelleyebilme özelliği önem kazanmıştır. Buna "online öğrenme" de diyebiliriz. Online öğrenme'de her bir eğitim örneği yalnızca bir kez kullanılır, ve yeni bir eğitim örneği geldiğinde sınıflandırıcı güncellenerek yeni örneği kullanır. Online öğrenme, verinin devamlı olarak yenilendiği ve başlangıç öğrenme veri kümesinin büyük olduğu yani

bütün veriyi saklamanın maliyetli olduğu durumlarda kullanılır. Çok sayıda kelimenin birleşiminden ve sınıflandırma performansı ile hesaplama miktarını azaltan alternatif algoritmalardan faydalanılarak, online öğrenme alanında gelecek çalışmalar için çeşitli olasılıklar mevcuttur.

Sonuç olarak, kategorizasyon programı uygulamasının başarısı da bize göstermiştir ki yakın gelecekte elle sınıflandırma yerini tamamen otomatik sınıflandırmaya bırakacak, ancak elle sınıflandırma daha çok özel girişimlerle sınırlı kalacaktır. İnsan gücü tasarrufu, daha sık güncelleme yapabilme, büyük miktardaki verinin yönetilmesi, minimum insan gücü ile yeni metin verilerinin kategorizasyonu, metinsel veritabanlarının içerikleri değiştiğinde bilinen bu veritabanlarının yeniden kategorizasyonu gibi avantajları yanında, kazandırdığı hız da bu metodolojinin gücünün kanıtlarıdır.

KAYNAKLAR

Amasyalı, F., Yıldırım, T., (2004), "Otomatik Haber Metinleri Sınıflandırma", Yıldız Teknik Üniversitesi.

Akbaba, O., (2003), "Web Sayfalarının Otomatik Olarak Sınıflandırılması Üzerine Yaklaşımlar ve Örnek Similasyon Uygulaması", "Gebze Yüksek Teknoloji Enstitüsü".

Dumais, S., Platt, J., David Heckerman, D., (1998), "Inductive Learning Algorithms and Representations for Text Categorization", Proceedings of ACM-CIKM98, 1:148-155.

Eyheralendy, S., Lewis, D., Madigan D., (2003) , "On the Naive Bayes Model for Text Categorization".

Frank, E., Pfahringer, B., Holmes, G.,Kibriya, A., Pfahringer, b., (2004) , "Multinomial Naive Bayes for Text Categorization Revisited ", 3339:488-489.

Hong, E.H., Karypis, G., (2001), "Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification", 5th Pacific-Asia Conference, 2035:53-65.

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

Jun, H., Hokuan, H., (2002), "An algorithm for text categorization with SVM" , TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, 1:47-50.

Kutlu, F., (2001), "Categorization In A Hierarchically Structured Text Database", Bilkent.

Manning , C., (2002), "Text Classification Lecture Notes".

McCallumzy, A., Nigam, K., (1998) "Comparison of Event Models for Naive Bayes Text Classification" , Pittsburg.

Metsis, V., Paliouras, G., (2006) "Spam Filtering with Naive Bayes –Which Naive Bayes".

Schneider, K., (2004), "On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification", Department of General Linguistics University of Passau, Sydney, 474-486.

Soucy, P., Mineau, W., (2001) , "A Simple KNN Algorithm for Text Categorization", IEEE International Conference, 647-648.

Soucy, P., Mineau, W., (2005), "Beyond TFIDF Weighting for Text Categorization in the Vector Space Model", IEEE International Conference, 1130-1135.

Tan, P., Kumar, V., (2006), "Lecture Notes for Chapter 5 Introduction to Data Mining".

Tan, P., Kumar, V., (2006), "Lecture Notes for Chapter 4 Introduction to Data Mining".

Visa, A., (2001), "Technology of Text Mining", Tampare University of Technology, 1-11.

Wang, Y., Hodges, J., (2003), "Classification of Web Documents Using a Naive Bayes Method", 15th IEEE International Conference, 560 – 564.

Williams, K., Calvo, R., (2006) ," A Framework for Text Categorization", NSW, Sydney University.

Zak, I., Ciura, M., (2002), "Automatic Text Categorization".

INTERNET KAYNAKLARI

[1] http://people.revoledu.com/kardi

[2] http://www.scils.rutgers.edu/~msharp/text_mining.htm

[3] http://www.text-mining.org/

[4] www.verivizyon.com

[5] http://www.clearforest.com/

[6] http://www.trl.ibm.com/projects/textmining/takmi/takmi_e.htm

[7] http://www.megaputer.com/

[8] http://www.textanalysis.info/

[9] www.cs.man.ac.uk/~jls/CS2411/DocumentClassify.pdf.gz

[10] http://www.cs.rpi.edu/courses/fall03/ai/misc/naive-example.pdf

ÖZGEÇMİŞ

Doğum tarihi	10.10.1980	
Doğum yeri	İstanbul	
Lise	1994-1997	Ataköy Cumhuriyet Lisesi
Lisans	1998-2002	Yıldız Üniversitesi Elektrik-Elektronik Fak. Elektrik Mühendisliği Bölümü
Yüksek Lisans	2004-	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Kimya-Metalurji Fakültesi Matematik Mühendisliği Bölümü

Çalıştığı kurum(lar)

2005-2005	Luckyeye Interactive Group Yazılım Mühendisi
2005-2006	SYS Sesli Yanıt Sstemleri Test Mühendisi
2006- Devam ediyor	Hobim Bilgi İşlem Merkezi Yazılım Uzmanı