ColumnREADY: İnternet Gazeteleri Köşe Yazılarını Hazırlama Uygulama Yazılımı

Mehmet Fatih KARACA[1], Salih GÖRGÜNOĞLU[2]

[1] Gaziosmanpaşa Üniversitesi, Erbaa Meslek Yüksekokulu, Tokat

[2] Karabük Üniversitesi, Bilgisayar Mühendisliği Bölümü, Karabük

mehmetfatih.karaca@gop.edu.tr, sgorgunoglu@karabuk.edu.tr

Özet: İnternetin günümüzde hayatımızın her alanına girmiş olmasıyla birlikte verilerin yaşantımızdaki değeri ve boyutu artmıştır. İnternette bulunan bütün veriler bir çok kişi için gerekli yada faydalı değildir. Verilerden katkı sağlayacak bilgilerin elde edilmesi için çeşitli madencilik işlemleri gerçekleştirilmelidir. Gerçekleştirilecek işlemlerden önce verilerin hazırlanması, ön işlemden geçirilmesi gerekmektedir. Bu çalışmada internet gazetelerinde yazı yazan köşe yazarlarının yazıları ele alınmıştır. Köşe yazılarının ön işlemden geçirilmesi, köşe yazısını temsil edecek terimlerin belirlenmesi ve köşe yazısının madencilik çalışmalarının yapılmasına hazır hale getirilmesi başarı ile gerçekleştirilmiştir.

Anahtar Sözcükler: Veri madenciliği, metin madenciliği, metin madenciliği ön işlemler

ColumnREADY: An Application Software for Preparing Website Article

Abstract: Values and amounts of data have been gradually increasing with the use of internet upon every area of daily routine within our lives. All data exist on internet are not necessary or beneficial for many people. Various mining processes should be realised in order to obtain contributory information out of data. Data should be prepared and pre-processed before the application of the processes. The articles of columnists who have been writing for newspapers on internet were utilised within the study. The following stages were successfully carried out: the pre-processes of articles, the determination of keywords which represent the article and the preparation of articles for mining studies.

Keywords: Data mining, text mining, text mining preprocesses

1. Giriş

Bilgisayarın birçok işi kolaylaştırdığı ve işlemleri daha hızlı yaptığı bir gerçektir. Bilgisayar kullanımının artması eldeki veri miktarının da o derece artmasını sağlamıştır. Bilgisayardaki herhangi bir dosyayı bulmak bile oldukça zaman alıcı ve zor bir iş halini alabilirken, arama işleminin dosya içeriklerine göre yapılması bu işlemi daha da zorlaştırmaktadır.

Elektronik ortamdaki bilgiler bilgisayarımızdaki dosyalardan ibaret değildir. Web sayfaları, webde yayınlanan köşe yazıları, e-mailler bunlara örnek olabilir. Bu tür veriler genel itibariyle yapısal olmayan verileri içermektedir. Merrill Lynch kullanılabilen iş bilgilerinin %80'inden fazlasının yapısal olmayan verilerden çıkarıldığını ifade etmiştir [1]. Günümüzde birçok veri yapısal olmayan formda tutulmaktadır. Bu durum bizleri sadece yapısal verilerle değil yapısal olmayan verilerle de ilgilenmeye, yapısal olmayan

verilerin yapısal veri haline dönüştürülmesine zorlamış, bununla ilgili çalışmalar yapılması gereksinimini arttırmıştır.

Yapısal veri, üzerinde terim olarak işlem yapmaya olanak sağlayan verilere denir. Yapısal olmayan veriler işlenmeden önce hazırlanmalıdır. Metin madenciliğinin amacı yapısal olmayan verinin veri madenciliğinde kullanılacak yapısal veri haline dönüştürülmesidir [2,3].

Yapısal olmayan "Ben M.Fatih Karaca, branşım bilgisayar, 2009'dan beri Gaziosmanpaşa Üniversitesi'nde Öğretim Görevlisi olarak görev yapmaktayım ve Tokat'lıyım." verisinin Tablo 1'de yapısal şekle dönüştürülmüş hali vardır.

Ad	M.Fatih
Soyad	Karaca
Çalıştığı_Üniversite	Gaziosmanpaşa Üniversitesi
Göreve_Başlangıç_Yılı	2009
Branş	Bilgisayar
Görev	Öğretim Görevlisi
Doğum_Yeri	Tokat

Tablo 1. Yapısal olmayan verinin yapısal forma dönüştürülmüş hali

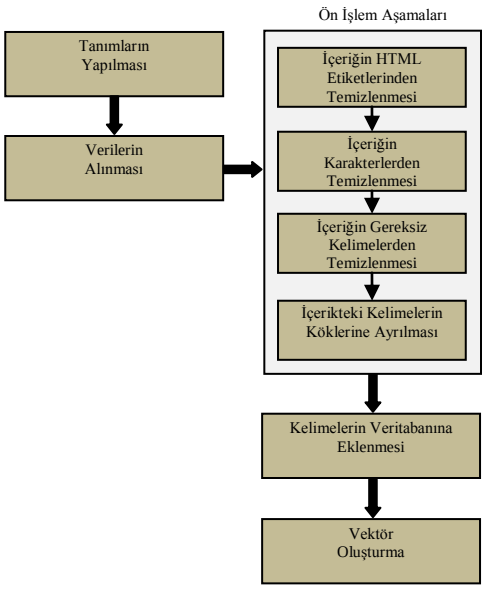
2. Sistemin Yapısı

ColumnREADY, Visual Basic ile Visual Studio 2008 ortamında Access veritabanı kullanılarak hazırlanmıştır. Ayrıca kelime köklerinin bulunmasında Java kullanılmıştır.

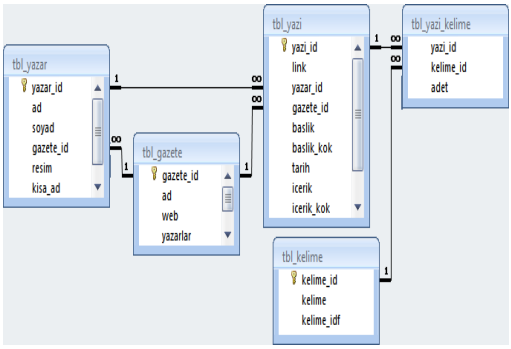
Veri seti olarak Akşam, Posta, Habertürk, Zaman ve Star gazeteleri kullanılmış olup her bir gazeteden 10'ar adet köşe yazarı alınmıştır. Köşe yazarları seçilirken siyaset, ekonomi, spor, sağlık, eğitim, magazin gibi çeşitli konularda yazı yazan yazarlar seçilmiştir.

Sistemin çalışması için öncelikle tanımlamaların (gazete ve yazar bilgileri) yapılması gerekmektedir. Bu işlemden sonra yazılar alınmalı, ön işlemden geçirilmeli, dokümanda geçen kelimeler veritabanına kaydedilmeli, vektör oluşturulmalıdır.

Şekil 1 sistemin çalışma yapısını, Şekil 2 ise kullanılan veritabanı modelini göstermektedir.



Şekil 1. Sistem yapısı



Şekil 2. Veritabanı modeli; *tbl_gazete, tbl_yazar, tbl_yazi, tbl_kelime, tbl_yazi_kelime* tabloları bulunmaktadır

3. ColumnREADY Uygulaması

3.1 Tanımların Yapılması

Tanımlama bölümünde sistemde bulunması istenilen gazete ve yazarlar belirlenir ve bu bilgiler veritabanına kaydedilir. Gazetelerin köşe yazarları ve köşe yazıları sayfalarına erişim Tablo 2'de gösterildiği gibi 2 farklı link formatı şeklinde olmaktadır. Yazar-

lar_Sayfası köşe yazarlarının tümünün görüntülendiği, köşe yazıları listelerinin bulunduğu sayfa linkini temsil ederken, Yazılar_Sayfası ise köşe yazıları içeriklerine erişimde kullanılan link olarak görülmektedir. Gazetelerin diğer sayfalar ile yazar ve yazılar sayfalarında farklı link formatı kullanmaları yazıların linklerini ve yazıları alma işlemini kolaylaştırmaktadır.

Yazarlar_Sayfası	http://www.haberturk.com/htyazar/
Yazılar_Sayfası	http://www.haberturk.com/yazar_ad/sayfa

Tablo 2. Habertürk gazetesi iki farklı link formatı örneği

Gazetelerin ve yazarların tanımlanması : Şekil 3'deki tbl_gazete tablosunun hazırlanmasındaki temel neden gazetelerin bünyesinde bulunan yazarlar ve yazılar için ortak bir yapı kullanmalarıdır. Veritabanına kaydedilen gazete ve köşe yazarları sınırlı sayıda tutulsa da yapılacak düzenlemelerle daha fazla gazete ve köşe yazarının sisteme dahil edilmesi mümkündür. Şekil 4'te veritabanında bulunan tbl_yazar tablosundaki veriler görülmektedir.

gazete_id	ad	web	yazarlar
1	Akşam	http://www.aksam.com.tr/	http://www.aksam.com.tr/
2	Posta	http://www.posta.com.tr/	http://www.posta.com.tr/yazarlar/tumyazilari?AuthorID=
3	Habertürk	http://www.haberturk.com/	http://www.haberturk.com/htyazar/
4	Zaman	http://www.zaman.com.tr/	http://www.zaman.com.tr/ara.do?method=search&yazarno=
5	Star	http://www.stargazete.com/	http://www.stargazete.com/yazar/
6	Taraf	http://www.taraf.com.tr/	http://www.taraf.com.tr/
7	Sabah	http://www.sabah.com.tr/	http://www.sabah.com.tr/yazarlar/

Şekil 3. Gazete tablosunda (tbl_gazete) *gazete_id, ad, web, yazarlar* alanları bulunmaktadır.

yazar_id	ad	soyad	gazete_id	resim	kisa_ad
10	İsmail	Küçükkaya	1	resim/ismail_kucukkaya	ismail-kucukkaya-1z
11	Deniz	Gökçe	1	resim/deniz_gokce	deniz-gokce-14z
12	Hüsnü	Mahalli	1	resim/husnu_mahalli	husnu-mahalli-38z
13	Nagehan	Alçı	1	resim/nagehan_alci	nagehan-alci-12z
14	Serdar	Akinan	1	resim/serdar_akinan	serdar-akinan-13z
15	Oray	Eğin	1	resim/oray_egin	oray-egin-3z
16	Çiğdem	Toker	1	resim/cigdem_toker	cigdem-toker-10z
17	Şansal	Büyüka	1	resim/sansal_buyuka	sansal-buyuka-43z
18	Simge	Çıtak	1	resim/simge_citak	simge-citak-36z
19	Elif	Aktuğ	1	resim/elif_aktug	elif-aktug-29z
20	Mehmet Ali	Birand	2	resim/mehmetali_birand	304
21	Rauf	Tamer	2	resim/rauf_tamer	286
22	Hakan	Çelik	2	resim/hakan_celik	289

Şekil 4. Yazarlar tablosunda (tbl_yazar) *yazar_id, ad, soyad, gazete_id, resim, kisa_ad* alanları bulunmaktadır.

3.2 Verilerin Alınması

Gazetelerin web sayfalarını profesyonel birim veya kişiler hazırlamaktadırlar. Bu durum web sayfalarını oluşturan HTML etiketleri (div, p, img, a, li vb), css ve diğer web nesnelerinin kullanımında düzeni sağlamıştır. Düzenli sayfalarda web nesnelerinin varlığı araştırılarak bir verinin başlangıç ve bitiş yerini bulmak daha kolaydır. Düzensiz sayfalarda ise web nesneleri gelişigüzel tasarlandığından aynı nesnenin, aynı içeriğe sahip farklı iki verisinde bile farklılık görülebileceği, farklı şekillerde formatlanabileceği muhtemeldir. Düzenli ile düzensiz oluşturulan web sayfalarından verileri almak aynı zorlukta değildir, düzenli sayfalardan verileri almak daha kolaydır. Bu tür sayfalardan verilerin alınması işlemi iki aşamadır;

- Yazı bilgilerinin elde edilmesi : link, başlık, içerik ve tarih bilgisi,
- Yazının elde edilmesi : köşe yazısının içeriği.

Yazı bilgileri ve yazılar alınırken sayfanın kaynak kodu üzerinden işlemler gerçekleştirilir. Kodlar incelendiğinde yazı bilgileri ve yazı için aşağıdaki başlangıç ve bitiş stringleri kullanıldığı görülür. Şekil 5'teki stringler `InStr` komutuyla arattırılır, eğer bulunursa veri çekilir.

```
<A class=header......(link başlangıç)
<DIV id=HaberTarih.......(link bitiş)

<P id=Haber...(yazı içerik başlangıç)
<DIV id=NewsCom...(yazı içerik bitiş)
```

Şekil 5. Habertürk gazetesi yazı bilgileri ve yazı alınması için gerekli olan başlangıç ve bitiş bilgisi

Gazetelerin web sayfalarında yazarların yazdıkları yazılar tarih sırasına göre sayfa sayfa tutulmakta ve her sayfada ortalama 12 yazı bulunmaktadır. Yazarın sayfasına erişildiğinde ilk sayfa görülür. Daha eski yazılara erişmek istienirse, sayfaların tek tek gezilmesi gerekmektredir. ColumnREADY kullanıcının bu taleblerini alır ve işlemi yapar.

Sisteme tanımlanan gazete ve yazar bilgilerine göre yazar sayfası ziyaret edilerek yazılar bulunmalı ve yazı bilgileri (link, başlık, içerik ve tarih) alınarak veritabanına kaydedilmelidir. Eğer eski yazılara da erişilmek istenirse ColumnREADY ekranındaki başlangıç ve

bitiş sayfalarına değerler girilmeli ve yazı bilgileri elde edilmelidir.

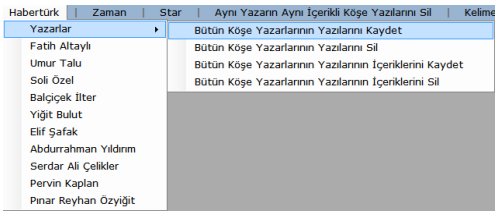
ColumnREADY her gazetede bulunan Şekil 6'daki menü alt seçenekleriyle aşağıdaki işlemleri gerçekleştirmekte ve Şekil 7'deki tabl_yazi tablosuna kaydeilmektedir. Bu işlemler sadece seçimi yapılan menünün bulunduğu gazeteyi etkilemektedir;

- Bütün Köşe Yazarlarının Yazılarını Kaydet : Gazetedeki bütün köşe yazarlarının başlangıç ve bitiş sayfaları arasındaki yazı bilgileri elde edilerek veritabanına kaydedilmektedir,

- Bütün Köşe Yazarlarının Yazılarını Sil : Gazetedeki bütün köşe yazarlarının yazı bilgileri veritabanından silinmektedir,

- Bütün Köşe Yazarlarının Yazılarının İçeriklerini Kaydet : Veritabanında yazı bilgileri mevcut olan bütün köşe yazarlarının yazıları siteden çekilerek veritabanına kaydedilmektedir,

- Bütün Köşe Yazarlarının Yazılarının İçeriklerini Sil : Veritabanında yazı bilgileri mevcut olan bütün köşe yazarlarının yazıları(yazı içerikleri) veritabanından silinmektedir.



Şekil 6. Uygulama programında ilgili gazetedeki tüm yazarlar düzeyinde işlem yapılan seçenekler ekranı

Yazı bilgilerini ve yazıları alma işlemi sadece yukarıda anlatılan menülerden değil ColumnREADY içerisindeki düğmelerden ve pop-up menülerden de gerçekleştirilebilmektedir.

Şekil 7. Yazı tablosunda (tbl_yazi) *yazi_id, link, yazar_id, gazete_id, baslik, baslik_kok, tarih, icerik, icerik_kok, durum* alanları bulunmaktadır.

3.3 Ön İşlem

Yazıların alınması işlemi bittikten sonra ön işleme aşamasına geçilebilir. Sınıflandırma, benzerlik bulma gibi metin analizlerinde işlemler metinle değil metni oluşturan kelimelerle yapılmaktadır. Metni HTML etiketlerinden, özel karakterlerden, gereksiz kelimelerden temizlemek ve kelimeleri köklerine ayırma ön işlem aşamasını oluşturmaktadır. ColumnREADY'de yazı içerikleri alınırken ön işlem yapmaya başlanmıştır.

İçeriğin HTML etiketlerinden temizlenmesi : Yapılan çalışmalarda yazıyı HTML etiketlerinden temizlemek şu şekilde gerçekleştirilir; yazı içeriklerinde HTML etiketleri aratılır, bulunur ve temizlenir. Bu HTML etiketlerinin sayısı ve farklı kullanım şekilleri göz önüne alındığında uzunca bir işlem olduğu görülmektedir. Bunun yerine kaynak koddan Şekil 5'te anlatıldığı gibi başlangıç-bitiş noktaları arasındaki metin alınır ve yazı geçici işlemler için kullanılan web browser nesnesine aktarılır. Browserdan Şekil 8'deki `Browser_Adi.Body.InnerText` koduyla HTML etiketlerinden arındırılmış salt metin elde edilir. Bazı HTML karakterleri browserda karakter çıktısı olarak görülmesine rağmen kaynak kodda ð, Ø gibi kod numaralarıyla görünürler. ColumnREADY bu tür karakterlerin kod şeklinde değil karakter çıktısı şeklinde gösterilmesini sağlar.

```
w_TMP.DocumentText = yazi
yazi =w_TMP.Document.Body.InnerText
```

Şekil 8. İçeriğin HTML etiketlerinden temizlenmesi için gerekli olan kod

İçeriğin karakterlerden temizlenmesi : Çift tırnak("), virgül(,), tire (-), yıldız (★, *) gibi

karakterler temizlenir. SQL komutlarında tırnak (') metin ayıracı olarak kullanıldığında yapılacak sorgularda problem yaşanmaması için (') karakteri (’) ile değiştirilir.

İçeriğin gereksiz kelimelerden temizlenmesi : Türkçede tek başına anlamı bulunmayan, anlama herhangi bir etkisi olmayan edat (gibi, için vb), bağlaç (ile, ama vb) gibi gereksiz kelimelerin (stop words) içerikten temizlenmesi gerekir. Çünkü bu kelimelerin anlama herhangi bir katkısı olmadığı gibi temizlenmediğinde yanlış sonuçlar elde edilmesine de neden olur. Bu kelimeler sisteme tanıtılmıştır ve bu kelimeler bulunursa içeriğe dahil edilmez.

İçerikteki kelimelerin köklerine ayrılması : Türkçe yapı bakımından sondan eklemeli diller arasında yer alır. Kelimeler ek almış ve almamış olarak bulunabilir. Tablo 3'deki gibi aynı köke sahip kelimeyi ifade eden ek almış farklı kelimeler bulunabilir.

Kelime	Kök
aramak	ara
aradım	ara
arayacaktım	ara

Tablo 3. Kelimeler ve kökleri

Çalışmalarımızda kökler yerine kelimeleri kullanmak bütün kelimeleri farklıymış gibi değerlendirmek anlamına gelir. Buda yapılacak işlemlerde yanlış elde edilmesini sağlar.

Kelimelerin köklerine ayrılması için Java'da program yazılmıştır. Bu programda kelimelerin köklerini bulmak için açık kaynak kodlu Türkçe Doğal Dil İşleme kütüphanesi Zemberek kullanılmıştır [4].

Yazılarda (veritabanında bulunan tbl_yazi tablosundaki baslik, icerik bilgisi) geçen kelimelerin köklerini bulma ve yeni verileri oluşturma işlemi şu şekilde gerçekleştirilmektedir;

- İçeriği boş olmayan yazı (veritabanındaki tbl_yazi tablosundaki baslik, icerik bilgisi için ayrı ayrı olmak üzere) alınır,
- Kelimeler diziye aktarılarak elde edilir,
- Elde edilen kelime dizisinin her bir elemanı Şekil 9'da görüldüğü gibi kökleri bulunmak üzere *kelimeCozumle("kökü_bulunacak_kelime")* adındaki Zemberek fonksiyonuna gönderilir ve Tablo 4'teki gibi fonksiyondan kelimenin kökleri, hangi ekleri aldığı ve kökün hangi tipte (sıfat, fiil, özel vb) olduğu döndürülür. Kök elde edilmiş olur.
- Kökler bir karakter boşluk bırakılarak birleştirilir,
- Yazı (veritabanında bulunan tbl_yazi tablosundaki baslik,_kok icerik_kok alanlarına) kaydedilir.

Kelimelerin yazıda olduğu gibi işleme alınması daha doğru sonuçlar alınmasını sağlar. Örneğin kelimelerin küçük harfe dönüşümü yapılırsa özel isimler üzerinde yapılan kök bulma işlemi yanlış sonuçlar verir.

```
Kelime[]
cozumler=z.kelimeCozumle("kelime");
for (Kelime kelime : cozumler)
   {
   System.out.println
   (kelime.kok().icerik());
   }
```

Şekil 9. Kelimeyi fonksiyona gönderen ve fonksiyondan aldığı kökü ekrana yazdıran kod

Kelime	Dönen Kök
ankara	Kök yok
ankara'da	Kök yok
Ankara	Ankara
Ankara'da	Ankara

Tablo 4. Kelimeler ve dönen kökler

3.4 Kelimelerin Veritabanına Eklenmesi

Kelime köklerinin veritabanına kaydedilmesi;

- Kelime köklerin kaydedilmesi,
- Yazılarda geçen kelime köklerinin veritabanına kaydedilmesi şeklinde çalışır.

Kelime Köklerin kaydedilmesi : Kelime köklerinin Şekil 10'daki tbl_kelime tablosuna kaydedilmesi şu şekilde gerçekleştirilmektedir;

- icerik_kok alanı boş olmayan yazı alınır,
- Yazılardan alınan kökler diziye aktarılarak elde edilir,
- tbl_kelime tablosunda bulunmayan kökler kaydedilir. kelime_idf'ye ağırlıklandırma bölümünde değinilmiştir.

kelime_id	kelime	kelime_idf
1	avrupa	0,64836
2	tur	1,718941
3	atla	1,472269
4	umut	0,967119
5	metris	2,25042

Şekil 10. Kelime tablosunda (tbl_kelime) *kelime_id, kelime, kelime_idf* alanları bulunmaktadır.

Yazılarda geçen kelime köklerinin veritabanına kaydedilmesi : Yazılarda geçen kelime köklerinin Şekil 11'deki tbl_yazi_kelime tablosuna kaydedilmesi şu şekilde gerçekleştirilmektedir;

- icerik_kok alanı boş olmayan yazı alınır,
- Yazılardan alınan kökler diziye aktarılarak elde edilir,
- Her bir dizi elemanı için hangi yazıda kaç kez geçtiğine bakılır ve tbl_yazi_kelime tablosundaki adet alanına kaydedilir.

yazi_id	kelime_id	adet
9	1	3
13	1	7
1	1	6
3	1	1
6	1	8

Şekil 11. Yazı_Kelime tablosunda (tbl_yazi_kelime) *yazar_id, kelime_id, adet* alanları bulunmaktadır.

3.5 Vektör Oluşturulması

Yazılar kelimelerin vektörel olarak ifade edilmiş halleridir. Yazılar üzerinde işlemleri gerçekleştirebilmek için ilk yapılması gereken yazının vektörel formata dönüştürülmesidir [5]. Vektör oluştururken bit, frekans ve Tf-Idf olmak 3 farklı yöntem kullanılır.

Kökler elde edildikten sonra köşe yazısını ifade edecek kelimelerin belirlenmesi gerekir. Her bir kelime için yazı içinde kullanım durumlarına göre ağırlıklandırma yapılır.

Her yöntem için 2 yazının, oluşturulan sözlükle olan ilişkisini anlatan örnek vardır;
Sözlük =\{öğrenci, ders, çalış, sınav, başarı, gözlük\}
Yazı1 =\{ders, öğrenci, futbol, oyna, ders, çalış\}
Yazı2 =\{arkadaş, sev, gözlük, ders, oyun, gözlük\}

Bit ağırlıklandırma yöntemi : Kelimenin yazı içerisinde bulunup bulunmadığıyla ilgilenir. Eğer kelime yazı içerisinde geçiyorsa 1, geçmiyorsa 0 değeri verilir.

Yazı1 ve Yazı2'nin bitsel ağırlıklı ifadesi;
D1 =(1,1,1,0,0,0)
D2 =(0,1,0,0,0,1)

Frekans ağırlıklandırma yöntemi : Kelime yazı içerisinde kaç kez geçtiği şeklinde ifade edilir. Eğer kelime yazı içerisinde bir kez geçiyorsa 1, 2 kez geçiyorsa 2 veya geçmiyorsa 0 değeri verilir.

Yazı1 ve Yazı2'nin frekans ağırlıklı ifadesi;
D1 =(1,2,1,0,0,0)
D2 =(0,1,0,0,0,2)

Tf-Idf ağırlıklandırma yöntemine göre tanımlama : Tf-Idf ağırlıklandırma yöntemi bunlardan biridir. Bu yöntemde sistemde bulunan toplam yazı sayısı (D) ve kelimenin geçtiği yazı sayısı (df_i) kullanılır. Bu iki değer çarpılarak kelimenin yazılar içerisinde ne kadar ayırt edici olduğu bulunur. Eğer sonuç yüksek ise kelimenin diğer yazılarda sıkça

geçmediği, değerli ve belirleyici olduğu anlamı çıkar. Eğer sonuç 0 ise kelimenin bütün yazılarda geçtiği, dolayısıyla yazıları ayırt etme de herhangi bir etkisinin olmayacağı anlaşılmaktadır. Formül eşitlik 1'de ifade edilmiştir.

$$Idf_i=\log(D/df_i) \quad (1)$$

Idf_i değerini kelimenin yazı içerisinde kaç kez geçtiğini gösteren değer olan tf_i ile çarparak kelimenin ağırlığını (w_i) bulunur. Formül eşitlik 2'de ifade edilmiştir.

$$w_i=tf_i * Idf_i \quad (2)$$

Tablo 5'teki öğrenci (i=1) kelimesine eşitlik 1 uygulanarak farklılık değeri (Idf_i) şu şekilde bulunur;

Idf_1=log(toplam_yazı_say/geçtiği_yazı_say)

Idf_1=log(2/1)=0.30102

Öğrenci kelimesinin birinci yazı için ağırlığı (w_i) ise eşitlik 2 uygulanarak yani yazıda kaç kez geçtiği ile çarpılarak elde edilir;

w_1=yazı_geçiş_sayısı*farklılık_değeri

w_1=1*0.30102=0.30102

Kelime	Farklılık Değeri
öğrenci (i=1)	0.30102
ders (i=2)	0 (bütün yazılarda var)
çalış (i=3)	0.30102
sınav (i=4)	- (hiçbir yazıda yok)
başarı (i=5)	- (hiçbir yazıda yok)
gözlük (i=6)	0.30102

Tablo 5. Kelime ağırlıkları

Her bir yazı içindeki sözcüklerin yazılardaki frekansları hesaplanıp vektör oluşturulur [6]. Yazı1 ve Yazı2'nin vektörel ifadesi şu şekildedir;

D1 =(1*0.30102, 2*0, 1*0.30102, 0,0,0)
D2 =(0,1*0,0,0,0,2*0.30102)

ColumnREADY'de Tf-Idf kullanılmıştır. Veritabanında bulunan yazıların toplam kelime sayısı, toplam yazı sayısına bölünmüş ve ortalama_terim_sayısı olarak tutulmuştur. Ağırlıklandırması yapılan kelimelerden ilk ortalama_terim_sayısı kadar kelimesi metni temsil edecek şekilde oluşturulmuştur. Bu şekilde her bir yazı için vektör oluşturulmuş olur.

4. Sonuç ve Öneriler

Geliştirilen ColumnREADY uygulama yazılımı, köşe yazılarını gazetelerin sitesinden alarak, yazılarda geçen kelimelerin köklerini bulmakta, ağırlıklandırmasını yaparak metni temsil edecek vektör oluşturmaktadır. Bundan sonraki yapılacak olan veri madenciliği işlemleridir. ColumnREADY köşe yazısı üzerine çalışma yapmak isteyen araştırmacıların veri alma, temizleme ve vektör oluşturma ihtiyacını karşılamaktadır. Sisteme gazete ve yazar ekleyerek tarama alanı genişletilebilir. Ayrıca, Zemberek kütüphanesinin bütün özel kelimeleri tanımıyor olması problemi, sisteme özel isimlerin tanıtılmasıyla ve bir yazarın aynı içerikli yazısının farklı linklerde tekrar yayınlanması problemi ColumnREADY'deki "Aynı İçerikli Köşe Yazılarını Sil" düğmesiyle çözülmektedir.

5. Kaynaklar

[1] Grobernik, M., Mladenic, D., "Text-mining Tutorial", **J. Stefan Institute**, Slovenia

[2] Dolgun, M.Ö., Özdemir, T.G., Oğuz, D., "Veri Madenciliği'nde yapısal olmayan verinin analizi: Metin ve web madenciliği", **İstatistikçiler Dergisi 2**, 48-58 (2009)

[3] Karadağ, A., Takçı, H., "Metin Madenciliği ile Benzer Haber Tespiti", **Akademik Bilişim 2010**, Muğla (2010)

[4] http://code.google.com/p/zemberek/downloads/detail?name=zemberek-2.1.1.zip (19.12.2011 14:00)

[5] Pilavcılar, İ.F., "Metin Madenciliği ile Metin Sınıflandırma", **Yıldız Teknik Üniversitesi FBE**, Yüksek Lisans Tezi (2007)

[6] Aşlıyan, R., Günal, K., "Metin İçerikli Türkçe Doküman Sınıflandırılması", **Akademik Bilişim 2010**, Muğla (2010)