Gazi Üniv. Müh. Mim. Fak. Der.
Cilt 24, No 2, 303-310, 2009

J. Fac. Eng. Arch. Gazi Univ.
Vol 24, No 2, 303-310, 2009

KENDİNDEN DÜZENLENEN HARİTALAR İLE DERS İÇERİKLERİNİN SINIFLANDIRILMASI

Yılmaz ALPDOĞAN ve Hasan Şakir BİLGE
Bilgisayar Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi, Gazi Üniversitesi, 06570, Maltepe-Ankara
alpdogan@gmail.com, bilge@gazi.edu.tr

(Geliş/Received: 31.08.2007 ; Kabul/Accepted: 27.03.2009)

ÖZET

Elektronik dokümanların sayısının büyük bir hızla arttığı günümüzde, otomatik doküman sınıflandırma sistemlerinin, bilgi yönetiminin geleceği açısından çok kritik olduğu değerlendirilmektedir. Bu çalışmanın amacı, teknik dokümanları içeriklerine göre otomatik olarak sınıflandırmaktır. Teknik doküman olarak, birçok terimin sıralanmasıyla oluşan bilgisayar mühendisliği lisans programlarında açılan derslerin içerikleri kullanılmaktadır. Bu çalışmada, danışmansız öğrenme özelliğine sahip Kendinden Düzenlenen Haritalar (SOM) kullanılarak ders içeriklerini otomatik olarak sınıflandıran bir sistem önerilmektedir. Sınıflandırma işleminden önce ders içerikleri üzerinde çeşitli önişlemlerin uygulanması gerekmektedir. Dokümanlardaki durak kelimeleri (bağlaç, zamir v.s.) temizlendikten sonra kelimelerin kökleri bulunmaktadır. Sadece bir dokümanda geçen kelimeler ayırt edici olmadığından dolayı atılmaktadır. Çok tekrar eden kelimeler ise, diğer uygulamalardan farklı olarak burada oldukça anlamlı ve önemli terimler olarak görüldüğü için atılmamaktadır. Daha sonra terim frekansı ve ters doküman frekansı verileri kullanılarak ağırlık vektörleri hesaplanıp normalize edilmiştir. Her ders için hesaplanan bu vektörler kullanılarak kendinden düzenlenen haritalar yöntemi ile sınıflandırma yapılmıştır. Sonuçlar, karşılaştırma amacıyla k-ortalama algoritmasının çıktıları ile birlikte gösterilmiştir. Ders içeriklerini kullanarak yapılan bu sınıflandırma çalışması ile, bir bölümün derslerinin arasındaki içeriğe dayalı ilişkiler açık bir şekilde görülmektedir. Ayrıca farklı üniversitelerin farklı kodlara ve adlara sahip fakat içerik olarak benzer olan dersleri, SOM haritası üzerinde başarılı bir şekilde birbirine yakın çıkmaktadır.

Anahtar kelimeler: Doküman sınıflandırma, kendinden düzenlenen haritalar, ders içerikleri.

CLASSIFICATION OF COURSE CONTENTS BY USING SELF-ORGANIZING MAPS

ABSTRACT

The number of electronic documents is growing at a high rate in today; therefore automatic document classification systems are becoming more important for the future of the information management. In this study, it is aimed to classify the technical documents automatically according to their contents. Course contents of computer engineering departments are used as technical documents, which contain many technical terms. In this study, a technical document classification system is proposed that is based on the Self-Organizing Map (SOM) algorithm, which is an effective unsupervised artificial neural network method. Before the classification process, some preprocessing steps have to be applied. First of all, stopwords are removed from documents. In order to increase the classification performance, the word stemming is needed. The words that are used in only one document are removed because of their less importance. Most frequently used words are not removed in contrary to other applications, because they are found to be important and meaningful in this data set. Next, term frequency and inverse document frequency data are used for calculation of normalized weighted vectors. By using these vectors of each course, document classification is performed by self-organizing map method. For comparison, the results are shown with the output of k-means algorithm. By using this classification study, the relations between the course contents of a department are very clearly visualized. Furthermore, different named and coded courses from different universities come successfully together in the final SOM map.

Keywords: Document classification, self-organizing map (SOM), course contents.

1. GİRİŞ (INTRODUCTION)

Günümüzde elektronik dokümanların sayısı büyük bir hızla artmaktadır. Bilgisayarlarda çok miktardaki dosyalar konularına göre elle oluşturulan çeşitli dizinlerin altında saklanmaktadır. Dosya sayısının artmasıyla yapılan gruplandırmalar nitelik kaybına uğramaktadır. İnternet üzerinde ise milyonlarca web sayfası bulunmakta ve İnternet kullanıcılarının artan içeriğin arasından aradıklarını bulması gittikçe güçleşmektedir. Aranan içeriğin daha kolayca ve isabetlice bulunabilmesi için geliştirilen açık dizinlerde web sayfaları konularına göre gruplandırılır. Bu açık dizinler insanlar tarafından elle oluşturulmaktadır. Açık dizinlere artan talebin karşısında elle sınıflandırma yetersiz kalmaktadır. Bu nedenle otomatik sınıflandırma sistemlerine ihtiyaç vardır. Otomatik sınıflandırma sistemlerinin, bilgi yönetiminin geleceği açısından çok kritik olduğu değerlendirilmektedir [1].

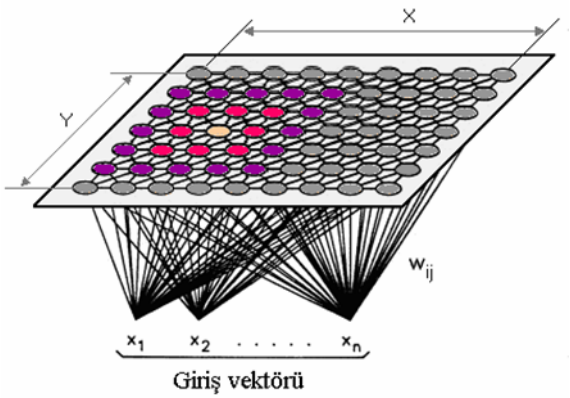
Otomatik doküman sınıflandırma için, danışmansız öğrenme tekniklerinden biri olan kendinden düzenlenen haritalar yaygın olarak kullanılmaktadır. Kendinden düzenlenen haritalar, bir veri kümesindeki var olan anlamsal (semantik) benzerlikleri başarılı bir şekilde ortaya çıkarmaktadır [2]. Kendinden düzenlenen haritalar, ilk olarak T. Kohonen (1997) tarafından geliştirilmiştir [3]. Kohonen (1998), 80 farklı Usenet grubundan elde ettiği 1 milyondan fazla mesajı kendinden düzenlenen haritalar kullanarak sınıflandırmıştır [4]. Bu çalışma benzer konulardaki mesajları gruplandırma konusunda başarılı olmuştur. Segal ve Kephart (1999), e-postaların otomatik sınıflandırılması için uyarlanır bir sınıflandırma geliştirmiştir [5].

Merkl ve Rauber (2000), veri kategorilerinin tek bir harita şeklinde gösterilmesinin yetersiz olduğunu iddia ederek çalışmalarını hiyerarşik kendinden düzenlenen haritalar üzerine yoğunlaştırmıştır [6]. Bu amaçla coğrafik haritalarda olduğu gibi genel bir harita ile başlayarak kullanıcının istediği noktalarda detay seviyelere inmesini sağlayacak bir algoritma üzerinde çalışmışlardır. Bunu yaparken yapay sinir ağlarının aslında belirli katmanlardan oluştuğunu ve her bir katmanın hiyerarşik olarak bir detay seviyesi olabileceğini dikkate almışlardır. Dolayısıyla her bir katmanın tekil bir kendinden düzenlenen harita olabileceği üzerinde durmuşlardır. Bu sayede verilerin istenilen detayda kategorize edilebileceğini göstermişlerdir. Tüm bu çalışmalarını danışmansız yapay sinir ağları üzerinde yapmış olan Merkl, 1990 yılına ait CIA Word Factbook'ta yer alan ülkelerle ilgili 245 adet dokümanı kullanarak değişik detay seviyelerinde bilgiler sunabilmiştir. Merkl, dokümanların gösteriminde vektör uzayı modelini kullanmıştır. Bu konudaki diğer detaylı bir çalışma ise Dittenbach ve arkadaşlarınındır (2002) [7].

Bu makalede, kendinden düzenlenen haritalar algoritması ile otomatik doküman sınıflandırma konusunda bir çalışma yapılmıştır. Çalışmada çeşitli üniversitelerin ders içerikleri kullanılmıştır. Deneysel çalışmalarda derslerin içeriklerine göre başarılı bir şekilde sınıflandırıldığı görülmüştür.

2. KENDİNDEN DÜZENLENEN HARİTALAR (SELF-ORGANIZING MAPS)

Kendinden düzenlenen haritalar, danışmansız öğrenen yapay sinir ağı modellerinden en önemli ve en yaygın kullanılanıdır [8]. Kendinden düzenlenen haritalar, dikdörtgen biçiminde 2 boyutlu düğümlerden oluşan bir ızgara ile gösterilir (Şekil 1). Her bir düğüm bir ağırlık vektörü ile ilişkilendirilmiştir. Ağırlık vektörlerinin boyutu ile ağa uygulanan giriş desenlerinin boyutları birbirine eşittir.



Şekil 1. Kendinden düzenlenen haritaların gösterimi (Representation of self-organizing maps)

Kendinden düzenlenen haritaların eğitilmesi işlemi kısaca ağa verilen giriş deseni ile ağırlık vektörlerinin uyarlanması şeklinde tanımlanabilir. Her bir eğitim iterasyonu *t*, rastgele bir giriş deseni *x(t)* seçimi ile başlar. Seçilen giriş deseni ağa alınır ve her bir birimin, $m_i(t)$, bu desen ile etkileşimi bulunur. Bu etkileşimi bulmak için genellikle her bir ağırlık vektörünün giriş desenine olan Öklit uzaklığı hesaplanır. Bu hesaplamayı yapay sinir ağlarında aktivasyon fonksiyonu olarak düşünebiliriz. Bu hesaplamalar sonucunda en az etkileşime sahip olan birim ilgili eğitim iterasyonunun kazanan birimi, *c* olarak kabul edilir (Eş. 1). Literatürde kazanan birim için BMU (Best Matching Unit) kısaltması çoğunlukla tercih edilmektedir [9-11].

$$c : m_c(t) = \min_i \left\| x(t) - m_i(t) \right\| \tag{1}$$

Daha sonra, kazanan birim ve çevresindeki bazı birimlerin ağırlık vektörleri güncellenmektedir. Bu güncelleme (uyarlama) işlemi, giriş deseninin ilgili bileşenleri ile ağırlık vektörünün farkına bir gradyan azaltımı uygulanmasıyla gerçekleştirilir (Eş. 2).

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot \left[x(t) - m_i(t) \right] \tag{2}$$

Bu fonksiyon iteratif bir fonksiyon olup, t iterasyon adımını ifade eder. Genel olarak formül bir düğüm için yeni ağırlığı, $m_i(t+1)$, mevcut ağırlığın, $m_i(t)$, bir fonksiyonu olarak göstermektedir. Formüldeki $x(t)$, t. iterasyondaki giriş desenini göstermektedir.

Eğitim boyunca, güncellenen birimlerin ağırlık vektörleri giriş desenine bir miktar yaklaştırılmış olmaktadır. Ağırlık vektörlerinin değişim hızı öğrenme katsayısı denilen $\alpha(t)$ ile belirlenir ve bu katsayı her bir iterasyonda üstel olarak azaltılarak sıfıra yaklaştırılır.

Etkileşime dahil edilecek birimler, komşuluk fonksiyonu denilen h_{ci} ile belirlenir. Etkileşime dahil edilen bu birimlerin sayısı da zamanla azalır ve eğitim işleminin sonuna doğru sadece kazanan birim etkileşime girer. Komşuluk fonksiyonu tipik olarak tek tepeli bir fonksiyon olup kazanan birimin bulunduğu yerin çevresinde simetrik ve kazanandan uzaklaştıkça tekdüze azalan bir yapıdadır. Komşuluk fonksiyonunu modellemek için bir Gauss fonksiyonu kullanılabilir:

$$h_{ci}(t) = \exp\left(-\frac{\left\|r_c - r_i\right\|^2}{2\sigma^2(t)}\right) \quad (3)$$

Bu eşitlikte, r_i, i biriminin ızgaradaki yerini gösteren 2 boyutlu bir vektördür. Eşitlikteki $\|r_c - r_i\|$ ise aktif eğitim iterasyonundaki kazanan birim c ile çıkış uzayındaki i birimi arasındaki uzaklığı göstermektedir. Eğitimin başında çıkış uzayının geniş bir alanı etkileşime uğramaktadır. Etkileşime giren birimlerin uzaysal genişliği (komşuluk yarıçapı) zamanla azalmaktadır. Bu strateji ile başlangıçta büyük kümelerin (cluster) oluşması ve eğitimin sonuna doğru çok daha küçük tanecikli ayrımların oluşması sağlanmaktadır [11]. Eğitimin başında, doküman matrisinin en (w) ve boy (h) değerinden büyük olanının yarısı alınarak yarıçapın başlangıç değeri bulunabilir. Bu komşuluk yarıçapı eğitim boyunca aşağıdaki formüle göre azaltılır:

$$\sigma = \frac{\text{toplam\_iterasyon\_sayisi}}{\log(\text{baslangic\_yariçapi})} \quad (4)$$

Ağırlık vektörlerinin hareketiyle giriş deseni ve ağırlık vektörü arasındaki Öklit uzaklığı sürekli azalır ve sonuçta ağırlık vektörleri giriş desenine çok benzer hale gelir. Böylece ilgili birimin sonraki iterasyonlarda kazanma olasılığı artmaktadır. Sadece kazanan birimin değil bu birime komşu diğer birimlerin de kazananla birlikte etkileşime dahil edilmesi neticesinde birbirine benzer desenlerin uzaysal kümelenmesi sağlanmaktadır. Böylece n boyutlu bir giriş uzayında bulunan giriş desenlerinden benzer olanları kendinden düzenlenen haritalar ile 2 boyutlu çıkış uzayında komşu olmaktadır. Çıkış uzayında benzer olan desenlerin coğrafik olarak birbirine yakın olacak şekilde kümelenmesi kendinden düzenlenen haritaların eğitim süreci ile sağlanmış olmaktadır.

Sınıflandırma işlemi bittikten sonra dokümanların anlaşılabilir bir şekilde görüntülenmesi için uygun bir şekilde etiketlenmesi işlemi de yapılabilir. Etiketleme işlemi genellikle dokümandaki en karakteristik kelimeler ile yapılır. Bu konuda çeşitli yöntemler bulunmaktadır. Bunlardan en yaygın olarak kullanılanı LabelSOM yöntemidir [12]. Bir dokümanı en iyi karakterize eden kelimeler, belirli bir dokümanın özeti gibi düşünülebilir. Etiketleme için her bir dokümandaki kelimelerin tekrar sayılarını tutan bir desen analiz edilmelidir. Bu çalışmada ise derslerin kodları ve adları yeterli olduğu için dokümanları başka bir şekilde etiketlemeye gerek kalmamıştır.

3. UYGULANAN YÖNTEM (APPLIED METHOD)

Bu çalışmada uygulanan yöntemin aşağıdaki adımlardan oluşması önerilmektedir.

3.1. Önişlemler (Preprocessing Steps)

Bu çalışmada farklı üniversitelerin bilgisayar mühendisliği bölümlerinin web sayfalarından alınan ders içerikleri toplanarak bir doküman kütüphanesi oluşturulmuştur. Bir dokümanda sınıflandırmaya doğrudan etkisi olmayan bazı verilerin temizlenmesi gerekir. Bu amaçla noktalama işaretleri, ASCII çizim karakterleri, çeşitli semboller ve sayısal ifadeler metinden temizlenmelidir. Her dilde doküman sınıflandırmaya etkisi olmayan bazı kelimeler, bağlaçlar ve harfler vardır. Dokümanlar analiz edildiğinde bu tür kelimelerin önemli ölçüde çok tekrar ettiği görülmektedir. Durak kelimeleri olarak adlandırılan bu kelimeler temizlenerek sınıflandırma daha doğru yapılabilmektedir ve hesaplama zamanı azalmaktadır. Sınıflandırma işleminin başarısını artırmak için kelimelerin kökleri bulunmaktadır ve bundan sonraki aşamalarda kökler kullanılmaktadır.

Doküman sınıflandırma için kullanılan veriler kelime sayıları dikkate alındığında çok yüksek boyutlara ulaşabilmektedir. Yüksek boyutlu veriler ise ağın eğitim süresinin önemli oranda uzun olmasına neden olmaktadır. Bu süreyi kısaltmak için çok az karşılaşılan terimler veya çok fazla tekrar edilen terimler de metinlerden temizlenir. Bu işlem için literatürde kabul edilen oran %10'dan az ve %90'dan çok geçen terimlerin temizlenmesi şeklindedir. Bu çalışmada sadece bir dokümanda geçen kelimeler atılmış, çok tekrar eden kelimeler anlamlı bulunduğu için atılmamıştır.

3.2. Kelimelerin İndekslenmesi (Indexing of Terms)

Tüm dokümanlarda geçen kelimeler ve her kelimenin kaç farklı dokümanda kullanılmış olduğu bilgisi bir tabloda tutulur. Tüm dokümanlardaki farklı kelime-

Çizelge 1. Kelimelerin aynı boyuta getirilmesi (Ealization of the size of the terms)

ilk durum:

D1	
Kitap	3
Defter	2
Kalem	1
Silgi	6

D2	
Masa	4
Defter	3
Silgi	7
Kalem	1

D3	
Silgi	9
Kağıt	3
Defter	6
Kalemtıraş	8

son durum:

D1	
Kitap	3
Defter	2
Kalem	1
Silgi	6
Masa	0
Kağıt	0
Kalemtıraş	0

D2	
Kitap	0
Defter	3
Kalem	1
Silgi	7
Masa	4
Kağıt	0
Kalemtıraş	0

D3	
Kitap	0
Defter	6
Kalem	0
Silgi	9
Masa	0
Kağıt	3
Kalemtıraş	8

lerin sayısını m ile gösterecek olursak bu tabloyu mx2 boyutlarında bir matris gibi ifade edebiliriz. Bu durumda 1.sütunda kelimeler 2. sütunda ise bu kelimelere karşılık gelen her bir kelimenin tekrar sayısı yer alır. Bu tablonun bütün dokümanlar için aynı boyuta getirilmesi işlemi Çizelge 1'de bir örnek üzerinde gösterilmiştir. Bu çizelgede ilk durum üst tarafta, son durum ise alt tarafta verilmiştir. Bu şekilde görüldüğü gibi bir dokümanda geçmeyen diğer dokümanlara ait kelimeler de bu dokümana sıfır değeri ile eklenmiştir. Örneğin D1 dokümanında hiç geçmeyen "Masa", "Kağıt" ve "Kalemtıraş" kelimeleri D1 dokümanı için kullanılan vektöre eklenmiştir. Benzer şekilde D2 dokümanında kullanılmayan "Kitap", "Kağıt" ve "Kalemtıraş" kelimeleri D2 doküman vektörüne, D3 dokümanında kullanılmayan "Kitap", "Kalem" ve "Masa" kelimeleri D3 dokümanına eklenmiştir.

3.3. Ağırlık Vektörlerinin Bulunması (Calculation of the Weight Vectors)

Dokümanlarda geçen kelimelerin ayrıştırma işlemi bittikten sonra her bir doküman vektöründeki terimlerin ağırlıklarından oluşan ağırlık vektörlerinin hesaplanmasına ihtiyaç vardır. Dokümanların modellenmesi için en yaygın kullanılan yaklaşım vektör uzayı modelidir [13]. Bu modelde, her bir doküman için bir vektör tanımlanır. Bu vektörün elemanları doküman içinde geçen kelimeler ve bu kelimelerin dokümandaki tekrarlanma sayısıdır (frekansıdır). Doküman vektörü bir ağırlıklı kelime histogramı olarak düşünülebilir. Bu amaçla kullanılan matristeki terimlerin değerlerini hesaplamak için kullanılan en yaygın formül [14]:

$$a_{ij} = f(TF_{ij}) \times g(IDF_{ij}) \tag{5}$$

burada, *TF*, Terim Frekansıdır, *IDF*, ise Ters Doküman Frekansıdır. Daha açık söylemek gerekirse; TF_{ij}, *j.* teriminin *i.* dokümandaki tekrar sayısıdır. *IDF* ise aşağıdaki gibi hesaplanmaktadır:

Çizelge 2. Ağırlık vektörlerinin esaplanması (calculation of the weight vectors)

D1			
	TF	*IDF*	*TFxIDF*
Kitap	3	1,477	4,431
Defter	2	1	2
Kalem	1	1,176	1,176
Silgi	6	1	6
Masa	0	1,477	0
Kağıt	0	1	0
Kalemtıraş	0	1	0

$$IDF_{ij} = \log\left(\frac{toplam\_dokuman\_sayisi}{j.\_terimi\_iceren\_dok\_say}\right) + 1 \tag{6}$$

Ağırlık vektörleri genellikle aşağıdaki sayıya bölünerek normalize edilirler ve sonra 0 ile 1 aralığına ölçeklenirler.

$$\sqrt{\sum (a_{ij}^{\ 2})} \tag{7}$$

burada, a_{ij} Eş. 5'teki ağırlık vektörüne karşılık gelmektedir. Eş. 7, Eş. 5'in normalizasyonu amacıyla kullanılan bir terimdir. Eş. 6'daki formüle göre Çizelge 1'de verilen D1 dokümanı için doküman vekörerinin ağırlıkları hesaplandıktan sonraki durumu Çizelge 2'de görülmektedir. Burada toplam doküman

sayısı 3 olarak alınmıştır.

Daha sonra, kelime ağırlıkları 0-1 arasına ölçeklenmektedir.

Bu işlemlerden sonra kendinden düzenlenen haritalar ile sınıflandırma yapılır. Kendinden düzenlenen haritalar algoritması kısaca şöyle ifade edilebilir:

1.Adım: 1.Adım: Çıkış işlem elemanlarına rastgele ilk değerler ver,
2.Adım: Eğitim kümesinden rastgele bir girişi seç,
3.Adım: Kazanan çıkış işlem elemanını belirle (Seçilen giriş desenine en yakın ağırlık vektörüne sahip işlem elemanıdır. Ağırlık vektörü ile giriş vektörü arasındaki uzaklık için genellikle Öklit uzaklığı kullanılır.),
4.Adım: Kazanan işlem elemanının ve çevresindeki komşularının ağırlık vektörlerini güncelleştir. Bu güncelleme ile ağırlık vektörleri giriş vektörüne yaklaştırılır. Bu yaklaştırma kazanan işlem elemanı için en fazla ve bu işlem elemanından uzaklaştıkça daha azdır. Öğrenme ilerledikçe komşuların sayısı azalmakta ve öğrenme sonunda sadece kazanan işlem elemanının ağırlık vektörü ayarlanmaktadır.
5.Adım: İterasyon sayısınca 2. adımdan itibaren tekrarla.

4. DENEYSEL SONUÇLAR (EXPERIMENTAL RESULTS)

Türkiye'deki üniversitelerin bilgisayar mühendisliği bölümlerinin lisans programları incelendiğinde, doğal olarak derslerin tamamen farklı kodlarla ve kısmen farklı adlarla anıldıkları görülmektedir. Her üniversite/bölüm kendi kodlama sistematiği ile derslere kod üretmektedir. Derslerin adları genel olarak birbirine benzemekle birlikte büyük farklılıklar da gösterebilmektedir; örneğin sayısal çözümleme ve nümerik analiz gibi. Öte yandan bir bölümde tek ders olarak okutulan içerik diğer bir bölümde iki ders kapsamında verilebilmektedir. Bir bölümde bir ders laboratuvar uygulaması ile birlikte anılırken, başka bir bölümde teorik kısmı ve ilgili laboratuvar uygulaması ayrı ders olarak gösterilebilmektedir. Bilgisayar mühendisliği alanında anlatılan konuların bir kısmı farklı ders başlıkları altında işlenebilmektedir. Bütün bu ve benzeri farklılıklar bölümlerin ders programlarını karşılaştırma açısından zorluklara neden olmaktadır. Ancak ders içeriklerine bakılarak doğru sonuçlar çıkarılabilir. Onlarca bölümün yüzlerce dersinin içeriklerini incelemek için ise otomatik bir sisteme gereksinim duyulmaktadır. Bu amaçla bu çalışmada ilk önce üniversitelerin bilgisayar mühendisliği bölümlerine ait web sayfalarından ders içerikleri alınmıştır.

Deneysel çalışmalarda iki farklı veri kümesi kullanılmıştır. Bunlardan biri bir bölümün derslerini (61 ders) içeren Türkçe bir doküman kütüphanesi, diğeri 6 farklı bölümün derslerini (toplamda 212) içeren İngilizce bir doküman kütüphanesidir. Bu dokümanların işlenmesi için, ilk önce noktalama işaretleri, ASCII çizim karakterleri, çeşitli semboller ve sayısal ifadeler metinden temizlenmiş ve metinler kelimelere ayrıştırılmıştır. Sınıflandırmanın başarısını artırmak için durak kelimeleri atılmıştır. Türkçe dokümanlar için kullanılan durak kelimeleri Çizelge 3'te listelenmiştir.

Daha sonra kelimelerin kökü (gövdeleri) bulunmaktadır. Türkçe kelimelerin kökleri Zemberek yazılımı aracılığıyla, İngilizce kelimelerin kökü ise Porter algoritması kullanılarak bulunmuştur. Son olarak, sadece bir dokümanda geçen kelimeler atılmıştır. Dokümanlardaki kelimelerden ağırlık vektörleri hesaplanmıştır; bu amaçla Eş. 5'teki formüle göre hesaplamalar yapılmış ve Eş. 7'deki terim ile normalizasyon yapılmıştır. Elde edilen veriler kendinden düzenlenen haritaların oluşturulmasında kullanılmıştır.

Çizelge 3. Türkçe dokümanlar için kullanılan durak kelimeleri (Turkish stop-words) (7x7 self-organizing map for 61 courses of a department after 1500 iterations, 13 clusters are colored according to k-means algorithm)

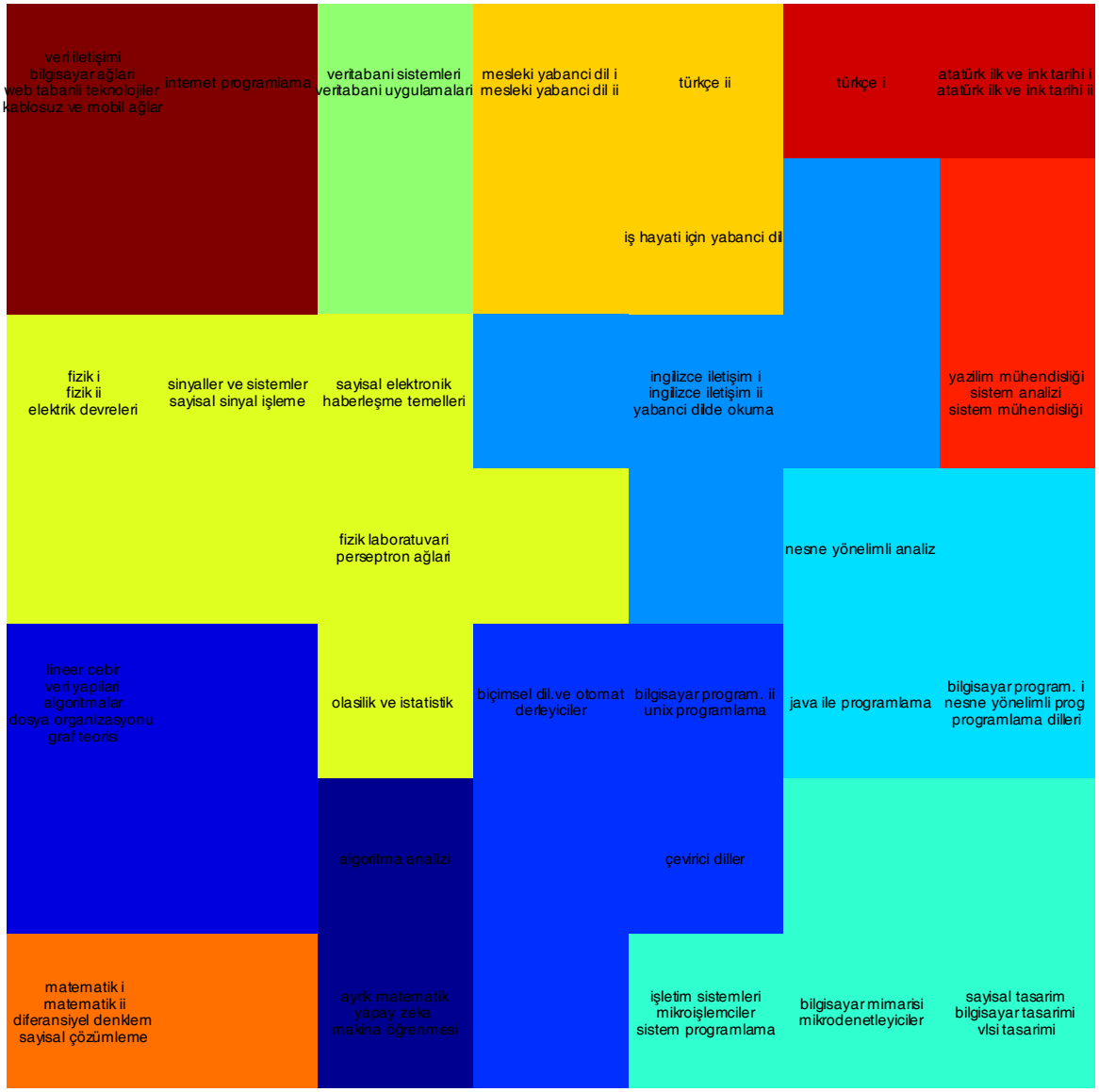
acaba	altı	altmış	ama	ancak	aynı	az	bana
bazı	belki	ben	bence	benden	beni	benim	beş
bin	bir	biri	birçok	birkaç	birşey	birşeyi	biz
bizden	bizi	bizim	bu	buna	bunda	bundan	bunu
bunun	çok	çünkü	da	daha	dahi	de	dedi
defa	diye	doksan	dokuz	dört	eden	elli	en
etti	fazla	gibi	hem	hep	hepsi	her	hiç
için	içinde	iki	ile	ilgili	ise	kadar	kez
kırk	ki	kim	kimden	kime	kimi	mı	mi
milyar	milyon	mu	mü	nasıl	ne	neden	nerde
nerede	nereye	niçin	niye	olarak	olsa	on	ona
ondan	onlar	onlardan	onları	onların	onu	otuz	önce
sanki	sekiz	seksen	sen	sence	senden	seni	senin
siz	sizden	sizi	sizin	son	şey	şeyden	şeyi
şeyler	şu	şuna	şunda	şundan	şunu	tek	tüm
üç	var	ve	veya	ya	yani	yedi	yetmiş
yirmi	yüz						

Bu çalışmada kullanılan kendinden düzenlenen haritalarda sıralı eğitim yöntemi tercih edilmiştir. Haritanın düğümlerine başlangıçta rasgele değerlere sahip vektörler atanmıştır. Kendinden düzenlenen haritaların sınıflandırma başarısı, öğrenme katsayısı ve iterasyon sayısına bağlıdır. Bu değerler doğru seçilerek yanlış sınıflandırmaların azaltılması sağlanabilir. Nitekim bu çalışmada yapılan ön incelemeler sonucunda doğru belirlenen öğrenme katsayısı ve iterasyon sayısı sayesinde yanlış sınıflandırmalar en aza indirgenmiştir. Yapılan çalışmalarda başlangıç öğrenme katsayısı 0.8 olarak alınmıştır.

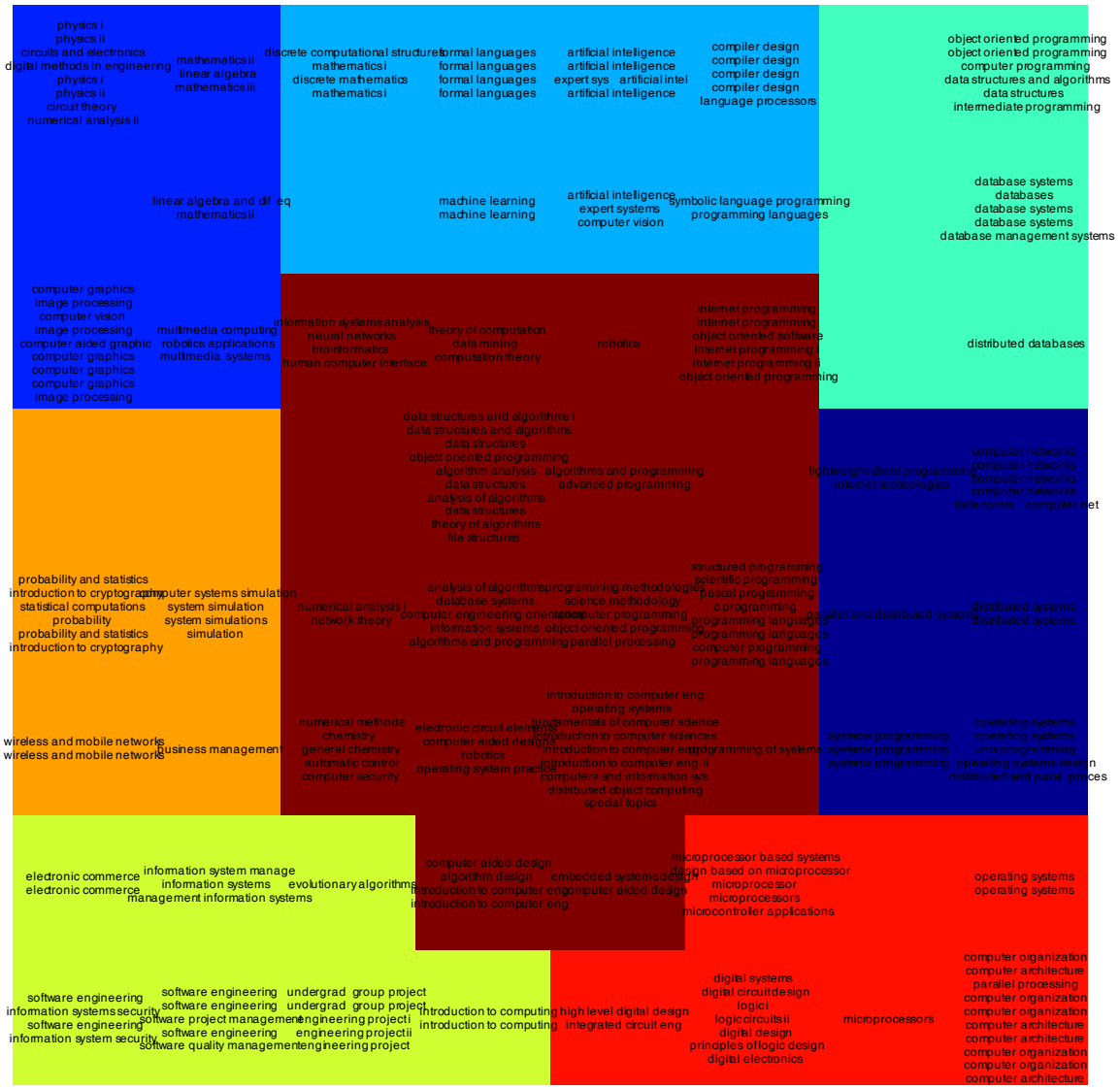
Deneysel çalışmada kendinden düzenlenen haritanın boyutları üzerinde bir inceleme yapılmıştır. Bu incelemenin sonucunda, haritanın boyutları küçük seçildiğinde birbirleriyle az ilgili olan dokümanların bile aynı düğüme girebildiği görülmüştür. Haritanın boyutları büyük seçildiğinde ise, çalışma zamanı oldukça artmıştır. Dolayısıyla bu çalışma için 7x7 veya 8x8 lik bir haritanın makul bir boyutta olduğuna karar verilmiştir.

İlk sonuç 7x7'lik bir haritanın 1500 iterasyonda eğitilmesi ile elde edilmiştir. Burada bir bölümün Türkçe olarak 61 dersi giriş olarak alınmıştır. Bu derslerin içeriklerindeki terimlerden ağırlık vektörleri hesaplanmış ve ağ buna göre eğitilmiştir. Sonuçlar Şekil 2'de görülmektedir. Önerilen yöntemin sonucunda birbirleriyle yakından ilgili olan dersler bir araya toplanmıştır. Elde edilen bu sonuçların başarısı, k-ortalama yöntemiyle bulunan kümelerle birlikte gösterilmiştir. Burada küme sayısı olarak 13 alınmıştır. Her küme farklı bir renk ile gösterilmiştir.

İkinci bir çalışmada, 6 farklı bölümün toplamda 212 İngilizce ders içeriği sınıflandırılmıştır (Şekil 3). Sonuçlar incelendiğinde farklı bölümlerde aynı içerikli dersler aynı karelerin içine, yakın içerikli dersler ise komşu karelere girmiştir. Buradan da



Şekil 2. Bir bölümün 61 dersinin 1500 iterasyon sonunda dağılımlarını gösteren 7x7'lik harita ve 13 küme(7x7 self-organizing map for 61 courses from 1 department after 1500 iterations, 13 clusters are colored according to k-means algorithm)



Şekil 3. 6 bölümün 212 dersinin 1500 iterasyon sonunda dağılımlarını gösteren 8x8'lik harita ve 8 küme (8x8 self-organizing map for 212 courses from 6 different departments after 1500 iterations, 8 clusters are colored according to k-means algorithm)

önerilen yöntemin başarılı bir şekilde çalıştığı görülmektedir. Elde edilen sonuçların başarısı, k-ortalama yöntemiyle bulunan kümelerle birlikte gösterilmiştir. Burada küme sayısı olarak 8 alınmıştır. Her küme farklı bir renk ile gösterilmiştir.

5. SONUÇLAR VE ÖNERİLER
(RESULTS AND DISCUSSION)

İnternet üzerinde web sayfalarının sayısı, büyük bir hızla artmaktadır. Artık otomatik arama motorları, arama sorgularına isabetli cevaplar vermekte yetersiz kalmaktadırlar. Dizin siteleri, bütün web sayfalarını değerlendirmeye yetişememektedir, dolayısıyla dizinlerin kalitesi ve kapsamı azalmaktadır. Ayrıca, bağlantılar güncelliğini kaybetmektedir. Öte yandan, bilgisayarlarda saklanan dokümanların sayısı ve hiyerarşisi de artmaktadır. Sonuç olarak web sayfalarının ve dokümanların otomatik olarak sınıflandırılması daha fazla önem kazanmaktadır.
Bu çalışmada, ders içeriklerinin otomatik olarak sınıflandırılması için bir sistem gerçekleştirilmiştir. Çalışmada özellikle yüksek boyutlu verilerde başarılı bir şekilde sınıflandırılma yapan kendinden düzenlenen haritalar yöntemi kullanılmıştır. Bu yöntem danışmansız olarak çalıştığı için, otomatik sınıflandırma için çok uygundur.

Uygulanan yöntemin aşamaları kısaca şu şekilde sıralanabilir: doküman kütüphanesinin hazırlanması, dokümanların okunması, durak kelimelerin temizlenmesi, kelimelerin indekslenmesi, az tekrar eden kelimelerin temizlenmesi, ağırlık vektörlerinin bulunması, normalizasyon, ağın eğitimi ve sınıflandırma sonucunun görüntülenmesi. Uygulanan yöntem ile ders içerikleri başarılı bir şekilde sınıflandırılmıştır.

Ders içerikleri sınıflandırılırken uygulanan sistemin başarısı değişik senaryolarla detaylıca irdelenmiştir. Benzer içeriğe sahip dersler başarılı bir şekilde yan yana gelmiştir.

Kendinden düzenlenen haritalarda dokümanların daha anlaşılabilir bir şekilde görüntülenmesi için uygun bir şekilde etiketlenmesi yapılabilir. Etiketleme işlemi genellikle dokümandaki en karakteristik kelimelerin bulunması ile yapılır. Bu çalışmada derslerin adları yeterli derecede açıklayıcı olduğu için ilave etiketlemeye gerek kalmamıştır.

Bu çalışmada uygulanan yöntem ile Web sayfaları ve haber gruplarındaki yazılar gruplanabileceği gibi elektronik posta mesajları kişinin özel ilgilerine göre otomatik olarak sınıflandırılabilir. Ayrıca resmi yazılar, kişisel dosyalar, tam metin veritabanları kolaylıkla sınıflandırılabilir. Bir işletmeye gelen yazı, makale vb. metinlerin ilgili kişilere otomatik dağıtımı yapılabilir. Örneğin bir ürün geliştirici ile pazarlama elemanının ilgileri birbirinden farklı olacaktır. İçerik patlamasının yaşandığı İnternet dünyasında bilginin otomatik sınıflandırılmasına olan ihtiyacın sürekli arttığı dikkate alınırsa, bu tür uygulamalara olan ihtiyaç daha iyi anlaşılacaktır.

KAYNAKLAR (REFERENCES)

1. Calvo, R.A., Lee, J.M., Li, X., "Managing content with automatic document classification", **Journal of Digital Information**, Cilt 5, No 2, 2004.
2. Amine, A., Elberrichi, Z., Bellatreche, L., Simonet, M., Malki, M., "Concept-based clustering of textual documents using SOM", **IEEE/ACS International Conference on Computer Systems and Applications**, 156-163, 2008.
3. Kohonen, T., "Self-organizing maps", Springer Series in Information Sciences, **Springer-Verlag**, New York, 30:1-426, 1997.
4. Kohonen, T., "Self-organization of very large document collections: State of the art", **Proceedings of the 8th International Conference on Artificial Neural Networks**, Skovde, Sweden, Cilt 1, 65-74, 1998.
5. Segal, R.B., Kephart, J., O., "MailCat: An Intelligent Assistant for Organizing E-mail", **Proceedings of the Third International Conference on Autonomous Agents**, Seattle, Washington, United States, 276-282, 1999.
6. Merkl, D., Rauber, A., "Document classification with unsupervised artificial neural networks", "Soft Computing in Information Retrieval: Techniques and Applications", (Editors: Fabio Crestani, Gabriella Pasi), **Springer-Verlag**, 102-121, 2000.
7. Dittenbach, M., Rauber, A., Merkl, D., "Uncovering hierarchical structure in data using the growing hierarchical self-organizing map", **Neurocomputing**, Cilt 48, No 1, 199-216, 2002.
8. Sağıroğlu, Ş., Beşdok, E., Erler, M., "Mühendislikte yapay zeka uygulamaları-I: yapay sinir ağları", ***Ufuk Kitap Kırtasiye Yayıncılık***, Kayseri, 23-116 (2003).
9. Koikkalainen, P., Oja, E., "Self-organizing hierarchical feature maps", **IJCNN International Joint Conference on Neural Networks**, Cilt 2, 279-284, 1990.
10. Vesanto, J., "SOM-based data visualization methods", **Intelligent Data Analysis**, Cilt 3, No 2, 111-126, 1999.
11. Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V. Saarela, A., "Self organization of a massive document collection", **IEEE Transactions on Neural Networks**, Cilt 11, No 3, 574-585, 2000.
12. Rauber, A., Merkl, D., "Automatic labeling of self-organizing maps: Making a treasure-map reveal its secrets", ***The Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining***, Beijing, China, 228-237, 1999.
13. Salton, G., McGill, M.J., "Introduction to modern information retrieval", **McGraw-Hill**, New York, 1986.
14. Salton, G., Buckley, C., "Term-weighting approaches in automatic text retrieval", **Information Processing and Management**, Cilt 24, No 5, 513-523, 1988.