İnternet Sayfalarındaki Asıl İçeriği Gösterebilen Akıllı Bir Tarayıcı

Tarık YERLİKAYA *Erdinç UZUN*

Bilgisayar Mühendisliği Bölümü
Mühendislik Mimarlık Fakültesi
Trakya Üniversitesi, EDİRNE

Email: tarikyer@trakya.edu.tr

Bilgisayar Mühendisliği Bölümü
Çorlu Mühendislik Fakültesi
Namık Kemal Üniversitesi, Çorlu, TEKİRDAĞ

erdincuzun@nku.edu.tr

Özet

İnternet sayfaları, kullanıcının görmek istediği asıl içerik dışında birçok gereksiz bilgi içermektedir. Bu çalışmada, kural tabanlı bir sistem kullanarak gereksiz içerikleri çıkarabilen bir akıllı tarayıcı anlatılmıştır. Bu tarayıcı, 6 web sitesi için 5 defa eğitilmiştir. Birinci eğitimden sonra gereksiz içeriğin %79,28 temizlenirken beşinci eğitimden sonra bu oranın %89,32'e çıktığı görülmektedir. Beş eğitim sonunda oluşan veritabanı incelendiğinde web sitelerinin ortalama %85,07 oranında gereksiz etiket içerdiği tespit edilmiştir.

1. Giriş

İnternet sayfalarındaki açılır pencereler, reklamlar, gereksiz resimler ve menüler kullanıcının dikkatini asıl içerikten uzaklaştırır. Bu çalışmada, kullanıcıya büyük oranda asıl içeriği gösterebilecek akıllı bir tarayıcı geliştirilmiştir.

İnternet sayfalarındaki kullanışlı ve anlamlı bilginin otomatik süzülmesi için Internet Explorer, Mozilla Firefox ve Opera gibi tarayıcı uygulamaları çeşitli eklentilere sahiptir. Karmaşıklığı gideren veya içeriği daha okunaklı hale getirmek için font büyütme, resimleri kaldırma, javascriptleri engelleme ya da bu metotların kombinasyonlarını kullanma bu uygulamalar tarafından tercih edilmektedir. Kullanışlı ve anlamlı bilginin süzülmesi için genelde "kara liste" yaklaşımı ile bazı gereksiz içerikleri engellenir. Bu konuda diğer bir yaklaşımsa, OPERA[1] yazılımı içindeki "Small Screen Rendering (SSR)" uygulamasının yaptığı gibi web sayfalarının ekran genişliğine sığmasını sağlamak için sayfa formatını değiştirmektir. Özellikle, bu yaklaşıma yakın HTML kod değiştiricileri[2] mobil telefon için geliştirilen tarayıcılar tarafından kullanılmaktadır. Geliştirdiğimiz uygulama, bu yaklaşımlardan farklı olarak sadece gerekli içeriği kullanıcıya göstermeyi amaçlamaktadır.

Doğal Dil İşleme ve bilgi erişimi algoritmaları, içerik bulmada "standart kelime hata oranını" kullanarak metinlerdeki gürültüyü azaltmaya çalışmaktadırlar[1]. Hata oranı, asıl içerikten yanlış aktarılmış kelimelerin sayısıdır. İçerik bulmada kullanılan algoritmalar genelde internet sayfasından alınan süzülmüş içeriği girdi olarak kabul eder[2]. Bu girdiyi üretmek için genelleşmiş süzücülerde veya özelleşmiş sürücüler kullanılabilir. Genelleşmiş süzücüler her web sayfası için genel bir kuralı kullanırken, özelleşmiş süzücüler, her web sitesi için uzmanlaşmış süzücü veritabanı gerektirmektedir [3,4]. Beklenildiği gibi özelleşmiş süzücüler genelleşmiş süzücülere göre çok daha iyi sonuç üretirler. Yaptığımız uygulama, belirlediğimiz farklı web siteleri için özelleştirilmiş bir süzgeç veritabanı kullanmaktadır. Ayrıca, bu veritabanını oluşturmak için kullanıcıya bir ara yüz sağlamaktadır.

Büyükkökten ve ark. [5, 6], PDA ve hücresel telefonlar üzerinde "akordeon özetlemesi" adını verdikleri çalışmalarında HTML etiketlerinin analiziyle bir özetleme inşa etmişlerdir. Çalışmalarında, <P>, <TD>, <FRAME>, <H1>, <H2>, <H3> ve <B> etiketlerini kullanarak PDA ve hücresel telefonlara içeriği taşımışlardır. Kurdukları yapı hiyerarşisi, DOM (Document Object Model[3]) ağaç tabanlı modellemeye benzemektedir. Bu hiyerarşik yapı, kolay bir şekilde yeniden eski durumuna geri getirilebilir. Yaptığımız çalışmada, var olan etiketler değiştirilmeden sadece gereksiz görülen etiketler arasındaki içerikler otomatik olarak kaldırılmıştır.

[1] http://www.opera.com/mobile/
[2] http://www.novarra.com/solutions/html-micro-browsers-and-on-device-portal/
http://www.skweezer.com/
http://www.teashark.com/
[3] www.w3.org/DOM

Gupta ve ark. [7] bir teknikler serisi ve bu serileri kullanan algoritmalar geliştirerek benzer bir uygulama geliştirmişlerdir. Uygulamamız, kural tabanlı HTML etiketlerine dayalı bir uzman sistem olması açısından bu çalışmadan ayrılır.

Chen ve ark.[8], DOM ağaçlarını kullanarak tarayıcıdaki her bölümü parçalara böler ve organize eder. Kullanıcı tarafından bu kısım seçildiğinde, tam ekran olarak bölüm yakınlaştırılır. Kullanıcı, bir önceki bölüme dönebilir ve tekrar ilgili olduğu bölümleri yakınlaştırabilir. Bu çalışmada geliştirdiğimiz uygulama, tarayıcıda sadece gerekli içerik elde etmeyi amaçlamıştır. Bu yazıda, öncelikle kurulan uzman sistem ve sistemin uygulaması anlatılarak, çalışmanın deneyler kısmında geliştirdiğimiz uygulamanın test sonuçları verilecektir.

2. Uzman Sistemler

Uzman sistemler, bilgi işleme yeteneğinin makine tarafından otomatik olarak gerçekleştirilmesi amacı ile sürdürülen çalışmalardır. Başka bir ifade ile insanın temel özellikleri olan "düşünme, anlama, kavrama, yorumlama ve öğrenme yapılarının programlamayla taklit edilerek problem çözümüne uygulanması" olarak da ifade edilebilir. [9-10].

HTML etiketlerini çok iyi kullanan uzman bir kullanıcı, içeriği etiketleri kontrol ederek çıkarabilir. Örneğin bir haber sayfasının sadece haberle olan kısmını kopyalayıp metin olarak alma işlemini çok rahat bir şekilde yapabilir. Bu çalışmada, bu uzman kullanıcının işlemini yapabilecek bir akıllı tarayıcı uygulaması geliştirilmiştir. Bu akıllı tarayıcıyı Jackson [11] bileşenlerine uyarlarsak:

- Bilgi tabanı: Belirli bir web sitesine ait HTML etiketleri
- Çıkarım Mekanizması: HTML etiketlerine göre çıkarımlar yapan mekanizma
- Kullanıcı Ara yüzü: Kullanıcının işini kolaylaştıracak eğitim ve tarayıcı görsel modülleri
- Bilgi edinim modülü: HTML etiketlerini oluşturan modül

Uzman sistemlerde bilgi tabanı yaratmada güncel yaklaşımların en popüleri kural tabanlı sistemlerdir. Bu çalışmada, HTML etiketlerinin ilgi durumunu içeren kural tabanlı bir sistem kullanılmıştır.

Çıkarım mekanizmasının işlevi, bilgi tabanındaki HTML etiketlerini yorumlama ve kontrolüdür. Hangi durumda hangi kuralın uygulanacağı çıkarım mekanizması tarafından belirlenir. Çıkarım mekanizması, yeni kurallar oluşturmak içinde kullanılır. Bu işlemler için, veritabanından ve kullanıcıdan elde edilen verilerden yararlanılır [12]. Geliştirdiğimiz uygulamada çıkarım mekanizması iki bölümden oluşmaktadır. Birinci bölüm, web sayfasının ortak etiketlerini bulduğu ve kullanıcının bunların ilgili veya ilgisiz olarak işaretlediği kısımdır. İkinci bölüm ise, veritabanından alınan sonuçlara göre web sayfasını gösteren tarayıcı kısmıdır.

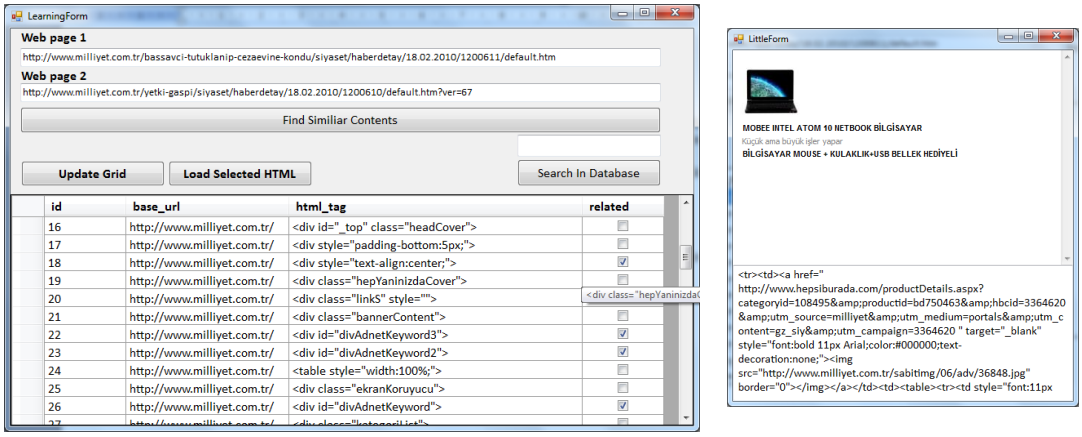
Çıkarım mekanizmalarının kolay kullanımı için görsel tabanlı kullanıcı ara yüzleri geliştirilmiştir. Özellikle, eğitim ara yüzü sayesinde farklı web siteleri eğitilebilir. Benzer bir şekilde mevcut web siteleri ilerideki ortaya çıkabilecek sorunlar için tekrar eğitilebilir. Eğitim ara yüzü, bilgi tabanının oluşturulması için bilgi edinim işlemini kolaylaştırır.

3. Akıllı Tarayıcı

Akıllı tarayıcı, iki modülden oluşmaktadır. Birinci modül, web sayfalarını filtresiz ve filtreli olarak gösteren tarayıcı bölümüdür. İkinci modül ise, web sayfalarındaki gereksiz bilgiyi öğrenmek için kullanılır. Öncelikle geliştirdiğimiz öğrenme modülü üzerinde duracağız.

3.1. Öğrenme Modülü

Web sayfaları, HTML etiketlerinden oluşur. Eğer HTML etiketlerinin içindeki içerik doğru bir biçimde çıkarılırsa istenmeyen içerik temizlenmiş olur. Web sayfalarını HTML kodunu incelediğinde içerik tutmak için "div", "table" ve "ul" etiketlerinin tercih edildiğini görülmektedir. Dinamik web sayfalarında, bu etiketler özellik bilgilerini otomatik olarak alırlar. Başka bir deyişle, reklamın, menülerin ve web sitesinin asıl içeriğinin etiketleri ve etiketin aldığı özellikler bellidir. Konusunda uzman bir kullanıcı bu içerikleri temizleyebilir. Öğrenme modülü daha az uzmanlığa sahip bir kullanıcı yol gönderici şekilde hazırlanmıştır. Öncelikle, kullanıcı benzer içeriğe sahip iki web sayfasının bağlantı adresini sisteme girer. Benzer içeriği tespit etmek istediğinde, iki web sayfasına ait ortak "div", "table" ve "ul" sahip HTML kullanıcıya gösterilir. Kullanıcı gerekli gördüğü etiketleri işaretleyerek filtreleme içeriğini belirler. Kullanıcı, seçili etiket içeriğini yükleye tıkladığında kullanıcıya web sayfasında yer alan içerik hem görsel hem de HTML olarak gösterilir. Bu bölüm, az uzman kullanıcıya yardımcı olması açısından önemlidir. Öğrenme modülü sadece bir kez değil HTML içeriğinde gereksiz işaretlemeler görüldüğü sürece tekrar kullanılabilir.



Şekil 1. Öğrenme Modülü ve İçerik Gösterici Formu

3.2. Etiketlerin tespiti ve etiketler arasındaki içeriğin çıkarımı

Bir internet sayfasında HTML içeriklerini ağaç şeklinde gösteren DOM HTML ayrıştırıcı kullanılabilir. Fakat, bu ayrıştırıcı HTML içindeki tüm etiket ayrıntılarını verir. Bu çalışmamızda sadece 3 etiketi ayrıştırmaya gerek vardır. DOM yerine, etiketlerin tespiti için bir metni düzenleme veya metin içersinden belli kurallara uyan alt metinleri elde edebilen düzenli ifadeler (regular expressions) olarak adlandırılan bir dil kullanılmıştır. Çoğu programlama dili tarafından desteklenmektedir.

Tablo 1. Çalışmada kullanılan düzenli ifadeler

<div.*?>
<ul.*?>
<table.*?>

Düzenli ifadeler, etiketleri belirlemek için kullanılabilir. Ancak, gömülü etiketleri belirlemek düzenli ifade ile zordur. Örneğin "div" içinde ikinci bir "div" etiketi açıldığında bitiş etiketinin tespitinin de düzenli ifade yetersiz kalır. Bu durum için etiket sayısını kontrol edip başlangıç etiketi ile o etikete ait bitiş etiketini bulup etiketin içeriğini getiren bir fonksiyon yazılmıştır. (Algoritma 1) Bu fonksiyon, hem içerik gösterici hem de içerik çıkarıcı bölümlerde kullanılmıştır.

Algoritma 1. Etiketler arasındaki veriyi çıkarıcı fonksiyon

fonksiyon İçerikÇıkarıcı
girdiler tüm_html_içeriği, islem_etiketi
degişken geçici_içerik;

eğer tüm_html_içeriği islem_etiketini içeriyorsa
değişken poziyonbilgisi = 0
değişken karakter_al = true
değişken geçici_etiket

geçici_içerik = islem_etiketini içerik içindeki başlangıçından tüm_html_içeriği sonuna kadar içeriği elde et

döngü tüm geçici_içeriği karakter bazında incele
  eğer karakter_al doğru ise
    gecici_etiket'e karakterleri al
  eğer geçici_içeriği[ ] "<"
    geçici_etiket'i temizle
    karakter_al doğru yap
  eğer sonu
  eğer geçici_içeriği[ ] ">"
    geçici_etiket'i temizle
    karakter_al false
  eğer sonu
  eğer geçici_etiket islem_etiketi ile aynı ise
    pozisyon_bilgisi bir artır
    geçici_etiket'i temizle
  eğer sonu
  eğer geçici_etiket "/"+islem_etiketi ile aynı ise
    posizyon_bilgisini bir azalt
  eğer geçici_etiket "/"+islem_etiketi ile aynı ve pozisyon_bilgisi = 0 ise
  geçici_içerik = geçici_içeriğin en başından döngü içindeki koordinata kadar içeriği al
  döngü sonuna git
  eğer sonu
döngü sonu

eğer sonu
geçici_içerik değerini döndür
fonksiyon sonu

Bu fonksiyonu örnek bir içerik üzerinden inceleyelim.

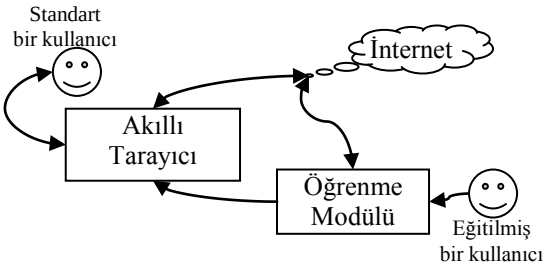
Örnek 1. HTML İçeriği

"
 İçerik Gömülü içerik 1 Gömülü İçerik 2 "

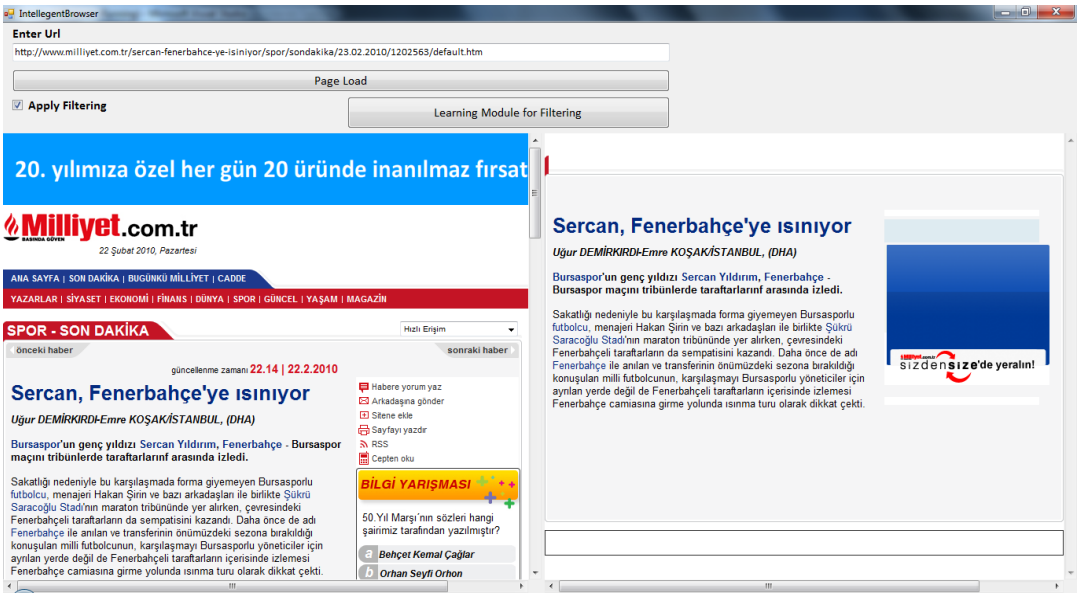
Bu içerikte düzenli ifade kullandığımızda " İçerik" sonucu ile karşımıza çıkar. Ancak yukarıdaki fonksiyonu kullandığımızda " İçerik Gömülü içerik 1 Gömülü İçerik 2" sonucu elde edilir. Görüldüğü gibi gömülü içerikler bu fonksiyon sayesinde elde edilmiştir. Gereksiz içeriğin temizlenmesinde de bu fonksiyondan dönen sonuçlar kullanılır.

3.3. Tarayıcı Modülü

Akıllı Tarayıcı, tüm içeriği ve filtrelenmiş içeriği yan yana gösteren sistemin ana modülüdür. Akıllı tarayıcı, internetten aldığı sayfaları öğrenme modülünden elde edilen kurallara göre HTML etiketlerini düzenler. Bu düzenleme işleminde gereksiz görülen HTML içerikleri filtreler. Şekil 3, bu filtreleme işlemi sonunda web sayfasının durumu gözükmektedir.



Şekil 2. Akıllı tarayıcının çalışması



Şekil 3. İçeriği filtrelenmemiş ve filtrelenmiş web sayfası örneği

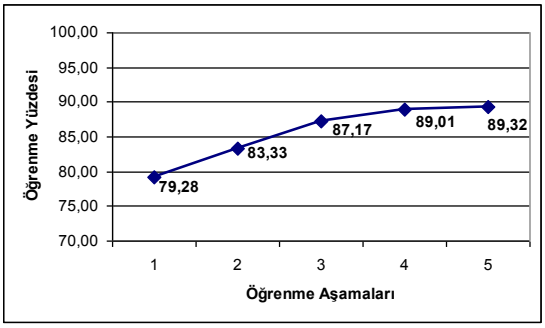
4. Deneyler

Deneylerde, 4 bilgisayar mühendisliği öğrencisine sistem kullanımı hakkında bilgi verildi. Öğrencilerin sistemi eğitmeleri ve test etmeleri için 6 adet web sitesi öğrencilere paylaştırıldı. Öğrenme işlemleri, sadece iki web sayfası için yapılmadığını belirtmiştik. Öğrenciler, kendi belirledikleri aynı web sitesine ait 10 farklı web sayfayı seçer ve sistemi 5 defa eğitir. Her eğitimde, diğer eğitimlerde öğrenilemeyen etiketleri ortaya çıkarılır. Öğrenme modülü sayesinde öğrenciler içerikle ilgili bilgileri işaretler. Kural veritabanı hazır olduktan sonra tarayıcı modülünü gidip her öğrenmenin sisteme katkısını incelediler. İlk deneyimiz, 6 web sitesi için ortaya çıkan kural veritabanını inceleyelim. Tablo 2'e göre bir web sitesi için ortalama 140,67 kural çıkarılırken bunların %14,93 ilgili etiket içerdiği görülmektedir. Kalan %85,07 gibi büyük oranda etiketin konu ile ilgisi olmayan menüleri, sayfa başlığı, reklam ve gereksiz içeriklerden oluştuğu görülmektedir.

Tablo 2. Kural veritabanında işaretleme sonrası etiket durumları

Web Siteleri	Öğrenilen Etiket Sayısı	İlgili Etiket Sayısı	İlgisiz Etiket Sayısı	İlgisiz Etiket (%)
www.hurriyet.com.tr	180	9	171	95,00
www.karakartal.com	144	30	114	79,17
www.milliyet.com.tr	137	16	121	88,32
www.sporx.com	121	23	98	80,99
www.vivivodo.com	106	21	85	80,19
www.izlesene.com	156	27	129	82,69
Ortalama	**140,67**	**21,00**	**119,67**	**85,07**

Toplamda öğrenilen etiket sayısı 844 iken bu etiketlerin %85,90 gibi oranının "div" etiketi olduğu görülmektedir. Kalan küçük bölüm ise, %11,14 ve %3,20 oranında sırasıyla "table" ve "ul" etiketi olduğu görülmektedir. Bu sonuç, web sitelerinin tasarımında genelde "div" etiketinin yüksek oranda kullanıldığını göstermektedir. İkinci deneyimiz ise oluşturulan kural veritabanının sonra akıllı tarayıcı bölümüne etkisi incelemektir. Bu inceleme için öğrencilere, akıllı tarayıcının filtresiz bölümü ile filtre uygulanmış bölümünün karşılaştırılması istenmiştir. Bu karşılaştırma işleminde öğrenciler gereksiz içerikleri (G.İ.), gereksiz olduğu halde gösterilen yani hatalı içerik (H.İ.) ve temizlenebilen gereksiz içerikler (T.İ) sayılarını tespit etmişlerdir. Daha sonra, 5 öğrenme aşamasını da aşamalı bir şekilde uygulayarak her öğrenmenin katkısını sisteme girmişlerdir.



Şekil 4. Öğrenme aşamaları sisteme katkısı

Altı adet web sayfası ortalama 13,42 gereksiz içerikten oluşmaktadır. 1. öğrenmeden ortalama 2,78 gereksiz içerik hala filtreleme aşamasını geçerken 5. öğrenme sonunda bu ortalama değer 1.43'e düşmüştür. (Yüzde oranları Şekil 4 verilmiştir.) Üçüncü deneyimiz, 5. öğrenme sonunda temizlenemeyen gereksiz içeriğin nelerden oluştuğunu incelemektir.

Tablo 3. 5. Öğrenme sonrası oluşan durum

	G.İ.	H.İ.	T.İ.	T.İ (%)
Menü	3,17	0,31	2,86	90,27
Sayfa Başlığı	1,34	0,12	1,22	91,29
Reklam	4,59	0,77	3,82	83,18
Diğerleri	4,32	0,24	4,08	94,52
Toplam	**13,42**	**1,43**	**11,99**	**89,32**

Tablo 3'e göre gereksiz içerikte en büyük hata oranının %83,18 ile reklamlarda olduğu görülmektedir. Bu sonuç, web sitelerinde farklı web sayfalarına bazı özel reklamlar için farklı etiket ismi tanımlanmasından kaynaklanmaktadır.

5. Sonuçlar

Bu çalışmada, bir web sayfasından gereksiz içerikleri çıkaran akıllı tarayıcı uygulaması ve deneysel olarak değerlendirilmesi anlatılmıştır. Gereksiz içerikleri çıkarabilen bu uygulama sayesinde görme engelliler için metinden seslendirme veya Braille çıktısı gibi uygulamaları gereksiz içerik olmadığı için daha iyi sonuç vereceklerdir. Ayrıca bu uygulamadan elde edilen içerik hücresel telefonlara çok rahat bir şekilde taşınabilir.

6. Referanslar

[1] W. Reichl, B. Carpenter, J. Chu-Carroll, W. Chou. "Language Modeling for Content Extraction in Human-Computer Dialogues", *In International Conference on Spoken Language Processing (ICSLP)*, 1998.

[2] E. Kaasinen, M. Aaltonen, J. Kolari, S. Melakoski, T. Laakko. "Two Approaches to Bringing Internet Services to WAP Devices", *In Proc. of 9th Int. World-Wide Web Conf.*, 2000.

[3] I. Muslea, S. Minton, C. Knoblock, "A Hierarchal Approach to Wrapper Induction". *In Proc. of 3rd Int.Conf. on Autonomous Agents (Agents'99)*, 1999.

[4] M. Kan, J.L. Klavans, K.R. McKeown. "Linear Segmentation and Segment Relevance". *In Proc.of 6th Int. Workshop of Very Large Corpora (WVLC-6)*, 1998.

[5] O. Buyukkokten, H. Garcia-Molina and A. Paepcke. "Accordion Summarization for End-Game Browsing on PDAs and Cellular Phones". *In Proc. of Conf. on Human Factors in Computing Systems (CHI'01)*, 2001.

[6] O. Buyukkokten, H, Garcia-Molina and A. Paepcke. "Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices", *In Proc. of 10th Int. World-Wide Web Conf.*, 2001.

[7] S. Gupta, G.E. Kaiser, P. Grimm, M.F. Chiang, J. Starren, "Automating Content Extraction of HTML Documents", *World Wide Web*, Vol. 8 , Issue 2, 2005, pp. 179-224.

[8] Chen, Y., Ma, W.Y., and Zhang, H.J. "Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices". Proc. WWW'03 Budapest, Hungary, May 2003.

[9] Ş. Sağıroğlu, E. Beşdok, M. Erler, "Mühendislikte Yapay Zeka Uygulamaları-1 Yapay Sinir Ağları", Ufuk Yayıncılık; Kayseri, 2003.

[10] İ. Kaya, Ş. Gözen, O. Engin, Kalite "Kontrol problemlerinin çözümünde uzman sistemlerin kullanımı", Havacılık ve Uzay Teknolojileri Dergisi, Cilt 1, Sayı 4, 2004, 87-101.

[11] P. Jackson, "Introduction to Expert Systems" 2nd Edition, Workingham: Addison-Wesley, 1990.

[12] İ. Vural, "Kalite Kontrolünde Uzman Sistemler", İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 1998.