Html İçindeki Gereksiz Kelimeleri Çıkaran Benzer Metin Tespit Uygulaması

Erdinç UZUN

Bilgisayar Mühendisliği Bölümü
Çorlu Mühendislik Fakültesi
Namık Kemal Üniversitesi, Çorlu, TEKİRDAĞ
Email: erdincuzun@nku.edu.tr

Özet

Kelimelerin metin içinde bulunma sıklığını kullanarak arama yapan arama motorlarından elde edilen sonuçlar, HTML içindeki gereksiz kelimelerden etkilenmektedir. Bu çalışmada, herhangi bir eğitim verisi kullanmadan metinlerdeki benzerliklerini tespit edip gereksiz metinleri çıkaran bir uygulama ve bu uygulamadan elde edilen sonuçlar anlatılmaktadır. Bu uygulama sayesinde HTML dokümana göre %90,59 oranında gereksiz kelime temizlenmiştir. Ayrıca, HTML etiketleri ayrıştırılıp içindeki kelimelere kök bulma işlemi uygulandığında sadece kelimelerin %20,38 oranında kökü bulunurken benzer metin tespit uygulaması sayesinde elde edilen kelimelerin %69,55 oranında kelime kökü tespit edilebilmiştir.

1. Giriş

Arama motorlarının web sayfalarını gezen ve bu sitelerden bilgi toplayan programlarına arama örümceği (crawler) denir. SET[*] (Searh Engine for Turkish) projesi kapsamında Milliyet haberlerini toplayan bir arama örümceği geliştirildi. Bu örümceği kullanarak Milliyetin 2003-2007 arasındaki verilerine çok rahat şekilde ulaşıldı. Ancak, 2007'den sonra Milliyet'in HTML içeriğinin değişmesi arama örümceğimizin çok fazla sayıda gereksiz bilgi almasına sebep oldu.

Web sayfaları, farklı etiketlere sahip birçok HTML dokümanlarından oluşmaktadır. Düzenli ifade (Regular Expression) veya DOM *(Document Object Model[*])* kullanarak sadece web sayfasının metinlerini çekmek mümkündür. DOM tüm HTML etiketlerini bir ağaç şeklinde ürettiği için çalışmamız için daha uygun olan düzenli ifade dili kullanıldı. İnternetten çekilen HTML dokümanları düzenli ifadeleri kullanarak ayrıştırdığımızda elde edilen metinler içinde hala reklam, menü ve çeşitli bağlantılarını içeren gereksiz metinlerin çok fazla olduğunu gördük. Bu çalışma, gereksiz metinlerden kurtulmak için kendi kendine öğrenebilen bir uygulamayı anlatmaktadır.

Gereksiz metinleri veya kelimeleri dokümanlardan çıkarmak için kullanılan tekniklerden biri derlem istatistiklerini kullanmaktır [1, 2]. Diğer bir yöntem ise, "standart kelime hata oranı" gerekli içeriğe ulaşmada kullanılan diğer bir yoldur [3]. Bu yaklaşımlar, büyük derlemlere ihtiyaç duymaktadırlar. HTML dokümanları çok sık değişen ve her web sitesinin kendine ait özellikleri olan metinlerden oluşur. Bu sebepten dolayı bu teknikler yeterli olmayacaktır.

İçerik çıkarımında diğer bir teknik daha önceden çıkarılan içeriği girdi olarak kabul ederek sonra gelen web sayfalarında bu bilgiyi kullanarak çıkarım yapabilen algoritmalardır[4, 5]. Daha özelleştirilmiş içerik çıkarıcılarında, her web alanı için özel çıkarıcılar ve bu alanlarını destekleyici algoritmalar kullanmaktır[6]. Bu çalışmalar kural tabanlıdır. Geliştirdiğimiz benzer metin tespiti yapan uygulamamız, kuralını benzer bir başka dokümana bakarak otomatik olarak çıkaran bir yaklaşımdır.

Bu çalışma için kullanılabilecek diğer bir teknik ise gözetimli öğrenme metotlarıdır [7, 8]. Gözetimli öğrenme metotları eğitim verisine ihtiyaç duyarlar. Hazırlanması zaman alıcıdır. Ayrıca web sitesinde bir değişiklik olduğunda eğitim verisi dışındaki durumlar gözden kaçırma ihtimalleri yüksektir.

Web siteleri günümüzde dinamik web programlama tekniklerini kullanarak web sayfasını kullanıcıya sunar. Benzer bir web sitesinin sayfa içeriğinde benzer menüler, reklamlar ve diğer metinlerden oluştuğu görülmektedir.

[*] SET, Türkçenin bilgi erişimi konusunda özelliklerini test etmek ve etkilerini araştırmak için geliştirilmiş bir arama motoru projesidir. *http://bilgmuh.nku.edu.tr/SET/* adresinden projeyi takip edebilirsiniz.
[*] www.w3.org/DOM

Geliştirdiğimiz uygulama, sisteme sunulan dokümanları karşılaştırır ve benzer bölümleri dokümanlar içinden çıkarır.

2. Çalışmanın doğuşu

Bir metni düzenlemek veya metin içersinden belli kurallara uyan metinleri elde etmek için düzenli ifade dilinden yararlanılır. Bu dil sayesinde Milliyet'in 2003-2007 arasındaki haber sayfalarının sadece haberle ilgili içerikleri çekilmiştir. Bu haberleri çekme için kullanılan düzenli ifade

<!--.PRINT.BASLADI.-->(.*?)<!--.PRINT.BİTTİ.-->

şeklindedir. 2007'den sonraki haberlerde bu etiketlerin kullanılmadığı görülmektedir. Bizde, HTML etiketleri temizlemek için sıklıkla kullanılan

<(.|\n)+?>

şeklindeki düzenli ifadeyi kullandık. Bu ifade bize HTML etiketlerini döndürdü. Buradan dönen etiketleri doküman içinden çıkardığımızda HTML etiketsiz dokümana ulaştık. Ancak, elde edilen dokümanları incelediğimizde içerikle alakası olmayan gereksiz birçok metin olduğu görüldü. Özelikle de, bu içerik ayrıntılı olarak incelendiğinde içinde reklamlardan, javascript kodlarından ve menülerden kalan gereksiz metinlerin olduğunu tespit edildi. Bu çalışma, bu gereksiz metinlerden kurtulmayı amaçlamaktadır.

3. Sistemin temelleri

SET projesi için geliştirilen arama örümceği üç modülden oluşmaktadır:

- İnternet üzerinde Milliyet sitesi içinde dolaşıp HTML doküman toplayıcı
- Web sayfalarının benzerliklerini tespit edip gereksiz metin çıkarıcı
- Ortaya çıkan metni, arama sonuçlarını hızlandırmak için indeksleyici

İnternet üzerinden diğer sayfalara ulaşmak için

<a[^>]*href=(.*?)>

düzenli ifadesi kullanılır. Buradan dönen sonuçlar indirilecek bağlantılar arasına alınır. Web sayfasından alınan HTML etiketleri temizlenerek metinlere ulaşılır. Elde edilen metinler de ayrıca temizlenerek saf metne ulaşmak için benzerlik tespit uygulaması geliştirilmiştir. Benzerlik tespit uygulamasından çıkan metinler indekslenmeye hazırdır. Bu aşamada eğer o dili özgü bir kök bulma algoritması kullanılırsa sadece kök kelimeleri de indekslemek mümkündür. Bu sayede sistem kaynaklarında kazanç ve arama sonuçlarında iyileşme sağlanabilir [9-12].

3.1. Benzerlik Tespit Modülü

Benzerlik tespit modülünde, internetten indirilen iki dokümanın karakter karakter karşılaştırması yapılmaktadır. Karşılaştırması yapılacak dokümanlar aynı web sitesine ait alt sayfalar olmalıdır. Bu çalışmada Milliyet'in haber bölümü kullanılmıştır. Uygulama, indirilen iki sayfayı kıyaslama şeklinde devam eder.

Algoritma 1. Benzer Metinler Listesini Döndüren Fonksiyon

fonksiyon BenzerlikTesbiti
girdiler: Karşılaştırması yapılacak iki HTML dokümanı
metin 1, metin 2: Düzenli ifade <(.|\n)+?>: İki HTML dokümanın metinlerini temizle

döngü 1: metin 1'deki tüm karakterleri al
  döngü 2: metin 2'deki tüm karakterleri al
    eğer metin[] 1 ile **metin[] 2** karakterlerini aynı ise
      döngü 1 metin[] 1 konumu arttırarak kontrole devam et
      döngü 1 karakter değişim miktarını hesapla
    eğer sonu
    eğer metin[] 1 ile **metin[] 2** karakterlerini farklı ise
          Eğer Karakter değişim miktarı 50'den fazla İse Benzer Metinler listesine ekle
          Eğer karakter değişim miktarı 50'den az ise **Metin[] 1** eski konumuna değişim miktarına göre dön
    eğer sonu
  döngü 2 sonu
döngü 1 sonu

Benzer Metinler Listesi, metin uzunluğuna göre sırala
Benzer Metinler Listesini döndür
fonksiyon Sonu

Bu fonksiyonda iki önemli nokta ortaya çıkmaktadır. Birinci nokta metin benzerliği bulunduktan sonra karakter değişim miktarının tutulmasıdır. Bu değişim miktarı kullanılarak 50 karakterden az bir benzerlik söz konusu ise 1. metinde eski konuma geri dönülür. 50 karakter benzerlik miktarı HTML içeriği içinde benzer kelimelerin, kelime gruplarının veya cümlelerin yanlışlıkla çıkarılmasını engellemek içindir. 50 değeri seçilirken Türkçedeki en uzun kelime ve 10 karakterlik hata payı düşünülmüştür.

İkinci önemli nokta, benzer metin listesinin metin uzunluğuna göre sıralanması olayıdır. Küçük metin listeleri büyük metin listelerinin alt kümesi olabilir. Küçük metin listesinin öncelikle sistemden çıkarılması büyük metin listesinin sistemden çıkarılamamasına sebep olacaktır. Bu sebepten dolayı benzer metin listesi karakter uzunluğuna göre sıralanır.

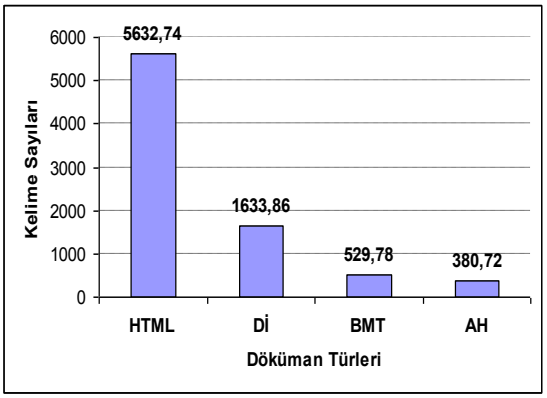
3.2. İndeksleme İşlemi

İndeksleme, HTML dokümandan elde edilen kelimeler ile web sayfası bilgilerini ilişkisel olarak tutulduğu bir işlemdir. İndekslemede, çoğunlukla daha az kelimeyi bellekte tutmak için ve arama sonuçlarını daha iyi duruma getirmek için kök bulma teknikleri kullanılır. Kök bulma teknikleri üzerine Türkçede çeşitli çalışmalar vardır [13, 14]. Ayrıca Can ve ark.[9] Türkçe için farklı kök bulma algoritmalarının arama sonuçlarına etkisi karşılaştırmıştır. Bu çalışmada, kök bulma işlemi için Zemberek* projesi kullanılmıştır. Değerlendirme bölümde Zemberek uygulamasının kaç kelimeye kök döndürüp döndüremediği test edilmiştir. Örneğin "işletmecilik" kelimesi aratıldığında Zemberek işletmecilik, işletme, işle ve iş kelimelerini sonuç olarak sisteme döndürür. Ancak, "Ahmet, İstanbul, Beşiktaş, Google" gibi bir özel isimler ve "TÜBİTAK, BJK, FB, GS" gibi kısaltmalar için sonuç döndürmemektedir. *

4. Deneyler

Bu çalışmada, Milliyet sitesinden rastgele 50 web sayfası örneği seçilip örnekler ikili gruplara ayrılmıştır. Deneylerde, indeksleme işlemi için önemli olan kelime sayıları kullanılmıştır. Karşılaştırma işlemleri HTML dokümanı (HTML), düzenli ifadeden dönen doküman (Dİ), benzer metinlerin tespitinden sonra dönen doküman (BMT) ve asıl haber dokümanı (AH) olmak üzere 4'e ayrılmıştır. İlk deney, bu 4 farklı dokümanı kelime sayıları üzerinden değerlendirilmesidir. İkinci deney ise otomatik olarak elde edilen dokümanların içindeki kelimelere Zemberek uygulamasının sonuç döndürüp döndüremediğinin test edilmesi olacaktır.

Bu sonuçlara göre, asıl haber dokümanı - HTML bir dokümana göre fiziksel olarak 85,21'lik daha az bir alan kapladığı görülmektedir. Düzenli ifadeler kullanılarak HTML dokümanına göre %70,99 bir iyileşme sağlanabilmiştir. Benzer metinlerin tespiti ve dokümanda çıkarılması sayesinde düzenli ifadelerden elde edilen sonuçlara göre %67,57 bir iyileşme sağlanmıştır.



Şekil 1. Oluşan 4 farklı doküman için ortalama kelime sayıları

Benzerlik tespiti, asıl haber dokümanına göre 1,39 daha fazla alan kaplamasına rağmen HTML dokümana ve düzenli ifadeden elde edilen dokümanlara göre tercih edilebilecek bir doküman türüdür. 4 farklı doküman arasında iyileşme (Tablo 1.'de sağ kenar) ve dosya boyutu (Tablo 1'de sol kenar) arasındaki oranlar aşağıdaki tabloda verilmiştir.

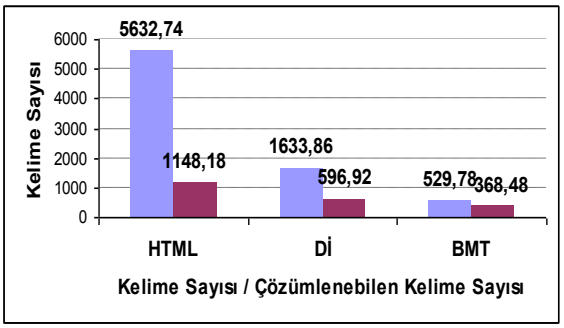
Tablo 1. İyileşme ve dosya boyutu oran ilişkileri

Doküman Türü	HTML	Dİ	BMT	AH
HTML	-	%70,99	%90,59	%93,24
Dİ	3,45	-	%67,57	%76,70
BMT	9,44	3,08	-	%28,14
AH	14,79	4,29	1,39	-

Dokümanlar incelendiğinde, benzer metinlerin tespitinden elde edilen dokümanlarla asıl haber dokümanları arasındaki fark web sayfası içindeki bu habere bağlı haber bağlantılardan ve kullanıcı yorum bölümünden kaynaklanmakta olduğu görülmektedir. İkinci deney, otomatik olarak elde edilen dokümanların kelimelerinin kök bulma işleminin etkinliğini test etmektir. Bu testlerde elde edilen kelimeler Zemberek fonksiyonlarından değer dönüp dönmediğine bakılmaktadır. Yapılan inceleme sonunda asıl haber dokümanına yaklaşıldıkça çözümlenebilen kelime sayısında arttığı gözlenmektedir. Şekil 2, doküman içinde bulunan kelime sayısı ve çözümlenebilen kelime sayısını vermektedir.

* Zemberek, açık kaynak koda sahip, platform bağımsız, Türk dilleri için tasarlanmış kütüphane ve araçlara sahip bir uygulamadır. Daha fazla bilgi için: *http://code.google.com/p/zemberek/*

* Zemberek uygulamasındaki bazı sözlük eksikliklerini otomatik olarak ekleyen Zemberek kütüphaneleri kullanan bir uygulama geliştirilmiştir. Uygulamayı kullanmak için: *http://bilgmuh.nku.edu.tr/SET/semcoz.aspx*



Şekil 2. Otomatik elde edilen dokümanlar için kelime sayıları

Bu deney sonuçlarına göre HTML dokümanın, düzenli ifadeden elde edilen dokümanın ve benzerlik tespiti sonucunda çıkarılan dokümanın sırasıyla %20,38 - %36,53 ve %69,55 oranında kelimelerin çözümlenebildiği görülmektedir. %69,55 ile bu deneyde de benzerlik tespitinin çok iyi bir iyileştirme sağladı görülmektedir. %30,45'lik çözümlenemeyen kısım içinde Zemberek veritabanında bulunmayan özel isimler ve kısaltmalar gibi kelimelerden kaynaklandığı görülmektedir.

5. Tartışma

Geliştirilen uygulama, benzer içeriklere sahip internet sitelerinin arama motorlarında indeksleme işlemini başlatmadan önce kullanabileceği sade bir yaklaşımdır. Bu çalışmadan elde edilen deney sonuçlarına göre:

1. HTML dokümandan elde edilen kelimeleri indekslemede kullanmak bilgisayarın fiziksel kaynaklarının tükenmesine neden olacaktır.
2. Düzenli ifadeleri kullanmak her ne kadar fiziksel kaynakları gibi görünse de asıl haber içeriğine bakıldığında hala çok sayıda reklam, menü ve bunlar gibi gereksiz içeriklerin fiziksel olarak tutulduğu görülmektedir.
3. Benzerlik tespit uygulaması sayesinde sayfalarda ortak olan gereksiz metinlerin büyük bir kısmının temizlendiği görülmektedir.
4. Benzerlik tespit uygulamasının diğer bir yararı ise indekslemede kök bulma işlemine katkısının daha çok arttığı görülmektedir. Bu sonuca göre, daha büyük oranda Türkçeye uygun kelimeye rastlandığı anlaşılmaktadır.

Bu konudaki diğer çalışmalarda gereksiz metin veya kelimeleri çıkarmak için eğitim verisi kullanılmaktadır. Bu çalışmada, eğitim verisine ihtiyaç duymayan sadece iki dokümanı karşılaştırıp sonuçları bulan bir uygulama geliştirilmiştir.

6. Sonuçlar

Bu çalışmada, Milliyet gazetesinin 2007 yılında sonraki içeriğini indekslerken karşılaştığımız sorunlar ve bu sorunlara çözüm olması için geliştirdiğimiz uygulama anlatılmaktadır. Bu uygulama sayesinde, asıl haber içeriğine çok yaklaşıldığı görülmektedir. Başka bir deyişle, reklamlardan ve menülerden gelen kelimeler sistemden temizlendiği için bu gereksiz metinler arama sonuçlarını etkilemeyecektir. Çalışmanın diğer bir özelliği ise, istatistiksel değerlendirmeye veya eğitim verisine ihtiyaç duymamasıdır. İhtiyaç olunan tek şey, benzer özelliklere sahip iki web sayfasıdır.

Bundan sonraki çalışmalarımızda, bu uygulamamızı geliştirerek farklı web ortamlarında da test etmeyi düşünüyoruz. Ayrıca, SET projesinin üzerine uygulamalar geliştirmeye ve karşılaştığımız sorunlara çözümler bulmaya devam etmeyi düşünmekteyiz.

6. Referanslar

[1] Y. Yang, J. Wilbur, "Using corpus statistics to remove redundant words in text categorization", Journal of the American Society for Information Science, *John Wiley & Sons, Inc., New York*, USA, Vol. 47, Issue 5, 1996, pp. 357-369.

[2] Y. Yang, "Noise reduction in a statistical approach to text categorization", *18th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, Washington, USA, 1995, pp.256-263.

[3] 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.

[4] A. F. R. Rahman, H. Alam, R. Hartono. "Understanding the Flow of Content in Summarizing HTML Documents", *In Int. Workshop on Document Layout Interpretation and its Applications*, DLIA01, Sep., 2001.

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

[6] M.Yen 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.

[7] C. Apte, F. Damerau, Sholom M. Weiss, "Automated learning of decision rules for text categorization", *ACM Transactions on Information Systems (TOIS)*, Vol. 12, Issue 3, 1994, pp. 233-251.

[8] E. Riloff, W. Lehnert, "Information extraction as a basis for high-precision text classification", *ACM Transactions on Information Systems (TOIS)*, Vol. 12, Issue 3, 1994, pp. 296-333.

[9] F. Can, S. Kocberber, E. Balcik, C. Kaynak, H.C. Ocalan, O.M. Vursavas, "Information retrieval on Turkish texts", *Journal of the American Society for Information Science and Technology*. Vol. 59, No. 3, pp. 407-421, 2008.

[10] C.G. Figuerola, R. Gomez, A.F.Z. Rodriguez, J.L.A. Berrocal, "Stemming in Spanish:a first approach to its impact on information retrieval.", *Working Notes for the CLEF 2001 Workshop 3 September*, Darmstadt, Germany, edited by Carol Peters. Retrieved September 3, 2006.

[11] J. Savoy, "Light stemming approaches for the French, Portuguese, German and Hungarian languages", *ACM SAC' 06*, pp. 1031-1035, Dijon: ACM, 2006.

[12] P. Ahlgren, J. Kekäläinen, "Swedish full text retrieval: Effectiveness of different combinations of indexing strategies with query terms", *Information Retrieval*, Vol.9, No. 6, pp. 681-697, 2006.

[13] G. Eryigit, E. Adali, "An Affix Stripping Morphological Analyzer for Turkish", *Proceedings of the IAESTED International Conference ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, Innsbruck, Austria, 2004.

[14] K. Oflazer, "Two-level description of Turkish morphology", *Literary and Linguistic Computing*, 9(2), pp. 137-148, 1994.