Türkçe Arama Motoru Sonucu Kümeleme Çalışmaları

Search Result Clustering Studies in Turkish

Burak Dural
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi
İstanbul, Türkiye
burakdural@gmail.com

Banu Diri
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi
İstanbul, Türkiye
banu@ce.yildiz.edu.tr

***Özetçe*—Internet, içeriği her geçen gün katlanarak artan bir iletişim ağıdır. Internette aradığımız bilgilere ulaşmamızı sağlayan arama motorları, kullanılan sorguya bağlı olarak milyonlarca sonuç döndürebilmektedir. Gelen sonuçlar içerisinden, kullanıcının aradığını bulması da ayrı bir problemdir. Bu problemin çözümü için uygulanan yöntemlerden biri de arama sonucu kümeleme işlemidir. Arama sonucu kümelemede kullanılan en popüler yöntemlerden biri olan Sonek Ağacı Kümeleme oldukça başarılı sonuçlar vermektedir. Bu çalışmada, mevcut Sonek Ağacı Kümeleme algoritması geliştirilerek Türkçe sayfalar üzerinde yapılan aramalarda, sonuçların kullanıcıya daha başarılı bir şekilde döndürülmesi sağlanmıştır.**

Anahtar Kelimeler — Sonek ağacı kümeleme; arama motoru sonucu kümeleme; doküman kümeleme.

***Abstract*—Contents of the internet is rapidly growing day by day. To reach the information we search on the internet, we're using internet search engines, which can return millions of search results depends on the query. Finding the information needed on the search results can be a massive problem. Search result clustering is a useful operation to solve this problem. One of the most popular search result clustering algorithm is Suffix Tree Clustering can achieve successful results. In this work, we try to improve Suffix Tree Clustering to get more successful results on clustering Turkish web search results.**

Keywords — Suffix tree clustering; search result clustering; document clustering.

I. GİRİŞ

Internet üzerindeki içerik her geçen gün artmakta ve bu içeriği indeksleyerek üzerinde arama yapılabilmesini sağlayan arama motorları da bu bilgi artışına paralel olarak artmış ve gelişmiştir. Günümüzde, kullanılan arama motorları verilen sorguya göre zaman zaman binlerce sonuç döndürmektedir. Bu kadar büyük bir veri içerisinden kullanıcının, aradığı bilgiye en yakın içeriğe sahip sayfaya ulaşması da zaman almaktadır. Arama motorları, bu sorunu çözmek ve arama sonuçlarını kullanıcıya en verimli olacak şekilde yansıtabilmek için; Sıralı Liste Sunumu, Arama Sonucu Kümeleme, İlinti Geribildirimi gibi bir çok sunum arayüzü sunmaktadırlar [1].

Arama Sonucu Kümeleme, arama motorlarında kullanılan sunum arayüzlerinden biri olup, arama motorundan döndürülen sonuçlara kümeleme işlemi uygulayarak, bu sonuçların tematik olarak kümelere ayrılması işlemidir.

Arama sonucu kümeleme işlemi, doküman kümeleme işleminin alt başlıklarından birisidir. Arama sonucu kümeleme algoritmalarının, sınırlı girdileri olan sezgisel açıklamaları kullanarak hızlı bir şekilde kümeleme yapmaları gerekmektedir. Arama sonuçlarında, sayfalar ile ilgili sayfanın adresi, başlık ve bir iki cümlelik kısa bir açıklama yazısı (snippet) bulunur. Arama sonucu kümeleme algoritmalarının yalnızca, bu bilgiler üzerinden hızlı bir şekilde kümeleme yapması beklenmektedir.

Scatter/Gather [2] ile başlayan arama sonucu kümeleme çalışmalarında bir çok algoritma kullanılmıştır. Bunlardan en çok kullanılan ve en başarılı olanlarından birisi de Sonek Ağacı Kümeleme (SAK)–Suffix Tree Clustering (STC) algoritmasıdır.

Sonek Ağacı Kümeleme, Zamir vd. çalışmasıyla [3] ortaya çıkan; hızlı, artırımsal işlemeye olanak veren, çakışan kümeleri oluşturma yeteneğine sahip, kısa doküman parçalarıyla çalışmaya elverişli bir algoritmadır. Zamir [3]'in çalışmalarıyla ortaya çıkan Sonek Ağacı Kümeleme, daha sonra arama sonucu kümeleme alanında da kullanılmıştır. SAK'ı temel alan ve algoritmayı geliştirmeye yönelik bir çok çalışma yapılmıştır [4] [5].

Arama sonucu kümeleme işlemlerinde, özellikle Türkçe içerikli internet sayfalarında başarılı kümeleme yapmayı hedefleyen çok fazla çalışma bulunmamaktadır. SAK gibi kullanılan popüler yöntemler Türkçe veriler üzerinde çalıştırıldığında; başarım İngilizce dilindeki kadar yüksek olmamaktadır. Bu sebeple yaptığımız çalışmada, SAK algoritmasını kullanarak ve bu algoritma üzerinde yaptığımız geliştirmelerle, Google (http://www.google.com.tr) arama motorunun sonuçlarını kullanarak oluşturduğumuz Türkçe veri setimiz üzerinde başarılı bir arama sonucu kümeleme işlemi yapılmaya çalışılmıştır. Veriler üzerinde Zemberek [6] kütüphanesi kullanılarak, Türkçe'ye daha uygun bir ön işleme aşaması ve SAK'a eklediğimiz yeni bir adım ile daha başarılı bir kümeleme işlemi yapılmıştır.

Makalenin ikinci bölümünde SAK ve işlem adımları, üçüncü bölümde deneysel sonuçlar ve dördüncü bölümde de sonuç yer almaktadır.

II. YÖNTEM

Yapılan çalışma sonucunda geliştirdiğimiz yöntemde, Zamir vd.[3]'nin geliştirdiği klasik SAK algoritması üzerinde geliştirmeler yapılarak başarı arttırılmaya çalışılmıştır. Uyguladığımız yöntem 4 aşamadan oluşmaktadır. İlk aşama ön işleme aşamasıdır. İkinci aşama, temel kümelerin belirlenmesi, üçüncü aşama, belirlenen temel kümelerin belirli kurallara göre birleştirilerek sonuç kümelerinin oluşturulması aşamasıdır. Bu üç aşama klasik SAK algoritmasında da yer almaktadır. Uyguladığımız Geliştirilmiş SAK (GSAK) algoritmasında, üçüncü aşama sonunda oluşan kümeler üzerinde yeni eklenen bir adım daha olmuştur.

A. Ön işleme

Tüm doküman kümeleme çalışmalarında olduğu gibi, arama sonucu kümeleme işlemlerinde de ön işleme adımlarının başarıya katkısı büyüktür. Yapılan başarılı bir ön işleme adımı, kümelemede kullanılan veriden daha etkili bir şekilde yararlanılması sağlanmış olur.

SAK yöntemi için yapılan çalışmalarda kullanılan ön işleme adımı, diğer doküman kümeleme işlemlerinde kullanılanlar ile kıyaslandığında daha basittir. SAK yönteminde kelimelerin cümle içerisindeki sırası da önem taşıdığından, aradaki kelimeler veri setinden çıkartılmaz. Bağlaç gibi ayrıştırıcılığı olmayan kelime türleri dahi, cümle içerisindeki kelime gruplarını tamamlaycı niteliğe sahip olduğu için önemlidir.

Yaptığımız ön işleme çalışması, SAK algoritmalarında kullanılanlara benzer olarak üç aşamadan oluşmaktadır. Bu aşamalar: Filtreleme, Ayrıştırma ve Hataların Düzeltilmesi'dir.

Filtreleme: Filtreleme aşamasında, cümlelerin sonuna gelen nokta, soru işareti, üç nokta, ünlem dışındaki tüm noktalama işaretleri silinir.

Ayrıştırma aşamasında, kelimeler tek tek incelenmektedir. Klasik doküman kümeleme yöntemlerinde kullanılan kelimelerin sadece köklerinin alınması işlemi yerine, eğer varsa sadece kelime sonlarındaki çoğul ekleri (-lar, -ler) çıkartılır. SAK algoritmasında etiket ard arda geçen kelimeler ve kelime gruplarından oluşmaktadır. Kelimelerin sadece köklerinin alınması, küme etiketleri için anlamsal kayıplara neden olabilir.

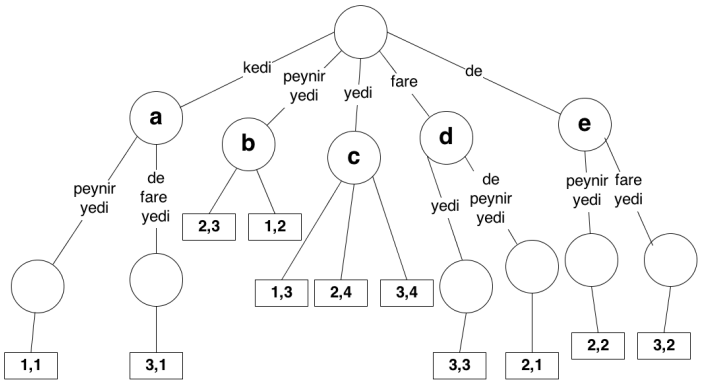
Hataların düzeltilmesi aşamasında, veri setinde yer alan kelimelerin hatalı yazılması, internet üzerinde sıkça karşılaşılan bir durum olan Türkçe karakter içeren bir kelimenin sadece İngilizce karakterle yazılması (balık yerine balik yazılması) gibi hatalar düzeltilir. Bu işlemler için Zemberek [6] kütüphanesinden yararlanılmıştır.

Ön işleme aşamasında, veri seti içerisinde çok fazla kullanılan ya da çok az kullanılan kelimelerin çıkarılmasına dayanan budama işlemi gerçekleştirilmemiştir. Bunun sebebi, SAK yönteminde kelime grupları da değerlendirmede önemli olduğundan, kelime gruplarının bölünmesinin istenmemesidir. Bu sebeple, yapılan ön işleme çalışmalarında cümle içerisindeki kelimelerin sıralarının bozulmamasına önem gösterilir.

B. Temel Kümelerin Belirlenmesi

Temel kümelerin belirlenmesi işlemi, ön işlemeden geçirilen veri seti üzerinde geçen ortak kelime ya da kelime gruplarının belirlenip, buna göre kümeleme yapılması işlemidir. Bunun için tüm veri setindeki dokümanlar (snipnet) kelime kelime bir son ek ağacına aktarılır. Son ek ağaçlarının genel kullanımından farklı olarak, SAK'de; son ek'ler yerine kelimeler, karakter katarları yerine ise dokümanlar kullanılır. Örnek bir son ek ağacı Şekil 1'de gösterilmektedir. Şekil 1'de "kedi peynir yedi", "fare de peynir yedi", "kedi de fare yedi" içerikli dokümanlar gösterilmektedir. Şekildeki düğümler üzerinde bulunan kutucuklardaki ilk sayı, düğümün hangi dokümanda bulunduğunu, ikinci sayı ise, düğüm etiketinin dokümanın kaçıncı kelimesinden başladığını göstermektedir. Bazı düğümlerde yer alan a, b, c, d, e harfleri oluşan temel kümeleri temsil etmektedir. Örneğin, a ile belirtilen kümenin içerisinde 1 ve 3 no'lu dokümanlar yer almaktadır.

C. Temel Kümelerin Birleştirilmesi



Şekil 1. Sonek Ağacı örneği

Dokümanlar birden fazla kelime grubu içerebilirler. Bunun sonucu olarak da, birbirinden farklı temel kümelerde bulunan doküman setleri kesişebilir. Hatta, birbirine eş temel kümeler bile oluşmuş olabilir. Bu durumun yaygın bir şekilde görülmesini engellemek için üçüncü aşama, yani temel kümelerin birleştirilmesi aşaması uygulanır.

Temel kümelerin birbirleriyle ne kadar kesiştiğini öğrenebilmek için, ikili bir benzerlik ölçümü yapılır. Verilen iki temel küme Bm ve Bn olsun. Bunlardaki doküman sayıları da |Bm| ve |Bn| olarak gösterilsin. |Bn∩Bm| de iki kümede birden olan doküman sayıları belirtilir. Eğer, (|Bm∩Bn|/|Bm|> 0.5) ve (Bm∩Bn|/|Bn|> 0.5) ise, iki temel küme arasındaki benzerlik değeri 1 olur. Aksi takdirde bu değer 0 olur. Benzerlik değeri 1 olan kümeler birleştirilir. Kullandığımız SAK algoritmasında varsayılan parametre değerleri referans alınarak kullanılmıştır. [3] 'teki çalışmada olduğu gibi Temel Küme Birleşim Eşik Değeri (Base Cluster Merge Threshold) 0,5 olarak seçilmiştir.

978-1-4673-5563-6/13/$31.00 ©2013 IEEE

D. SAK Geliştirmeleri

Klasik SAK algoritmasında aldığımız sonuçları, eklediğimiz yeni bir aşama üzerinde tekrar değerlendirerek, sonuç kümeleri oluşturulur. Bu aşamada oluşan sonuç kümelerini tekrar inceleyerek;

- İlk aşamada, en az %70 oranında aynı elemanları taşıyan kümeler tespit edilerek birleştirilir.
- İkinci aşamada, skoru 5'in altında olan kümeler tek tek incelenir. Eğer, bir kümenin elemanlarının %50'si diğer kümeler içerisinde yer alıyorsa, bu küme dağıtılır. Küme, diğer kümelerde yer almayan elemanlara sahip ise, *Diğer* ile etiketlenmiş kümeye eklenir.
- Son aşamada ise, elimizdeki küme etiketleri tekrar gözden geçirilerek, aynı kelimeleri içeren etiketler varsa bu kümeler arasındaki eleman benzerliğine bakılır. Eğer, benzerlik %50'nin üzerindeyse bu kümeler birleştirilir. Birleştirilen kümenin yeni etiket değeri, birleştirilen kümelerden skoru en yüksek olan kümenin etiket değeridir.

III. DENEYSEL SONUÇLAR

SAK algoritmasında yapılan geliştirmenin sonucunu görebilmek için, farklı anlamlarda kullanılan 5 sorgu kelimesi (Balık, Yüz, Takvim, Ocak, Dünya) belirlenerek, Google arama motoruna verilmiştir. Arama sonucu dönen sayfalardan her sorgu için ilk 50 tanesi alınarak veri seti oluşturulmuştur. Veri setimiz üzerinde klasik SAK (KSAK) ve üzerinde geliştirmeler yaptığımız geliştirilmiş SAK (GSAK) yöntemleri çalıştırılarak kümeleme işlemi gerçekleştirilmiştir. Ayrıca, veri setindeki veriler bir gönüllünün yardımıyla elle kümelenmiş ve bu işlemin sonuçları doğru kümeleme olarak kabul edilmiştir. KSAK ve GSAK yöntemlerinin başarısı el ile yapılmış kümeleme işleminin sonuçlarıyla karşılaştırılarak, F-Ölçüm kriteri üzerinden değerlendirilmiştir.

A. Yüz Sorgusu için Deney Sonuçları

Deney verilerimizde kullandığımız sorgulardan biri olan *yüz* için yapılan kümeleme işlemlerini incelediğimizde; KSAK'nin 9, GSAK'nin 6 adet küme oluşturduğu görülmektedir. Oluşan küme etiketleri Tablo 1' de gösterilmiştir.

Tablo 1. Yüz sorgusu için çıkartılan etiketler

Yöntem	Etiket Sayısı	Çıkarılan Etiket İsimleri
KSAK	9	Bakım, Türkiye, Wikipedi, Yüz Nakli, Oyun, Bakım Oyunu, Uzman TV, Bakım Ürünleri, Diğer
GSAK	6	Bakım Ürünleri, Oyun, Yüz Nakli, Wikipedi, Uzman TV, Diğer

Deneysel çalışmalarda yöntemlerin performans değerlendirilmesi yapılırken her metodun F-ölçüm değeri kriter olarak alınmıştır. F-ölçüm, Tutturma (1) ve Bulma (2) değerlerinin harmonik ortalaması alınarak hesaplanmaktadır. Tutturma; yöntemden getirilen verinin ne kadarının gerçekte getirilmesi gereken veri arasında olduğunu gösterir (1). Bulma ise; getirilmesi gereken verinin ne kadarının yöntem tarafından getirildiğini gösterir (2). F-ölçüm; Tutturma ve Bulma değerlerinden yararlanılarak eşitlik (3) 'deki gibi hesaplanır.

$$\text{Tutturma} = \frac{\{\text{getirilen sayfa}\} \cap \{\text{ilgili sayfa}\}}{\text{getirilen sayfa}} \quad (1)$$

$$\text{Bulma} = \frac{\{\text{getirilen sayfa}\} \cap \{\text{ilgili sayfa}\}}{\text{ilgili sayfa}} \quad (2)$$

$$\text{F} - \text{ölçüm} = \frac{2 * \text{Tutturma} * \text{Bulma}}{\text{Tutturma} + \text{Bulma}} \quad (3)$$

getirilen sayfa; yöntemin, incelenen etiket için getirdiği sayfaları, *ilgili sayfa* da; incelenen etiketin, elle sınıflandırma ile elde edilen sayfalarıdır.

Yaptığımız kümeleme işlemi için F-ölçüm sonuçlarını hesaplarken; öncelikle oluşan her kümenin F-ölçüm (3) değeri hesaplanmakta, daha sonra her kümenin boyutu da göz önünde bulundurularak, sorgunun ortalama F-ölçüm değeri (4)'teki gibi hesaplanmaktadır (*T* tüm kümeler seti; *c*, *T*'nin bir kümesi).

$$\text{Ort. F} - \text{Ölçüm} = \frac{\sum_{c \in T}(F_c * |c|)}{\sum_{c \in T}|c|} \quad (4)$$

KSAK sonuçlarını değerlendirirken her kümenin ayrı ayrı F-ölçüm puanları ve ortalama F-ölçüm değeri Tablo 2' de gösterilmektedir.

Tablo 2. *Yüz* sorgusu için KSAK F-ölçüm sonuçları

Etiketler	Tutturma	Bulma	F-ölçüm
Bakım	0	0	0
Türkiye	0	0	0
Wikipedi	1	1	1
Yüz Nakli	1	1	1
Oyun	1	1	1
Bakım Oyunu	0	0	0
Uzman TV	1	1	1
Bakım Ürünleri	1	1	1
Diğer	0.79	0.89	0.83
Ortalama	**0.71**	**0.74**	**0.72**

GSAK ile yapılan kümeleme işlemi için de alınan başarı sonuçları Tablo 3'te gösterilmektedir.

Tablo 3. *Yüz* sorgusu için GSAK F-ölçüm sonuçları

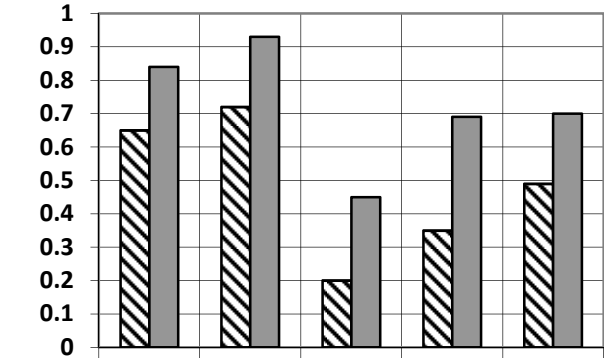
Etiketler	Tutturma	Bulma	F-ölçüm
Bakım Ürünleri	1	1	1
Oyun	0.92	1	0.96
Yüz Nakli	1	1	1
Wikipedi	1	1	1
Uzman TV	1	1	1
Diğer	0.79	0.88	0.83
Ortalama	**0.90**	**0.96**	**0.93**

Yüz sorgusu için sonuçları incelediğimiz de, Ort. F-ölçüm değerinin KSAK için 0.72 iken, GAK yöntemi uygulandığında 0.93'e yükseldiği görülmektedir. Buradan GSAK yöntemi ile *Yüz* sorgusu için başarının arttırıldığı gözlemlenmiştir.

B. Genel Değerlendirme Sonuçları

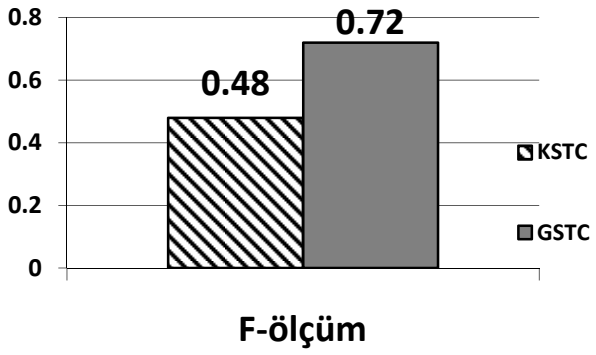
Denemelerimiz için seçtiğimiz tüm sorgu kelimelerini, *yüz* sorgusunda olduğu gibi çalıştırıp, F-ölçüm değerlerini hesapladığımızda da elde edilen sonuçlar Şekil 2'de gösterilmektedir.

Tüm veri setimiz için elde ettiğimiz sonuçların ortalamasını aldığımızda Şekil 3'teki gibi bir sonuç elde edilir. Yapılan geliştirme işleminin başarıma ne kadar katkı sağladığı açıkca görülmektedir.



	Balık	Yüz	Takvim	Ocak	Dünya
KSTC	0.65	0.72	0.2	0.35	0.49
GSTC	0.84	0.93	0.45	0.69	0.7

Şekil 2. Tüm sorgular için F-ölçüm sonuçları



Şekil 3. Tüm veri seti için ortalama F-ölçüm sonuçları

IV. SONUÇ

Arama sonucu kümeleme işlemleri, arama motorlarının kullanılmaya başlanması ve yaygınlaşmasıyla ortaya çıkmıştır. Arama motorlarının döndürdüğü sonuçlar arttıkça dönen sonuçların, kullanıcı tarafından daha verimli bir şekilde kullanılabilmesi için arama sonucu kümeleme yönteminin de önemi artmaktadır.

Yaptığımız çalışmada hedefimiz, Türkçe içerikli internet sayfalarının arama sonucu kümeleme işleminin başarılı bir şekilde yapılmasını sağlamaktır. Bunun için arama sonucu kümeleme alanında tercih edilen bir yöntem olan SAK algoritması üzerinde durulmuş ve Türkçe içerikli arama sonuçları üzerinde SAK algoritmasıyla daha başarılı kümeleme yapabilmek için algoritmaya bazı değişiklikler ve eklemeler yapılmıştır. Geliştirilen Sonek Ağacı Kümeleme, GSAK, yönteminin başarısını sınamak için elde edilen F-ölçüm değerleri Zamir vd.[3]'nin sunduğu klasik SAK algoritması (KSAK) ile karşılaştırmalı olarak verilmiştir.

KSAK ve GSAK yöntemi ile elde ettiğimiz sonuçlar karşılaştırıldığında, GSAK ile başarının arttırıldığı rahatlıkla görülmekte, ortalama başarı KSAK için %48 iken, GSAK ile başarı %72'ye çıkmıştır.

KAYNAKÇA

[1] Osinksi, S., "*An Algorithm For Clustering of Web Search Results*", Master Thesis, Poznan University of Technology, Polonya, 2003.

[2] Cutting, D. R., Karger, D. R., Pedersen, J.O. ve Tukey, J.W., "Scatter/gather: A cluster-based approach to browsing large document collections", *In Proceedings of The 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM New York, NY, USA, 1,318–329, 1992.

[3] Zamir, O. ve Etzioni, O., ."Web document clustering: A feasibility demonstration", *In Proceedings of The 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 46–54. ACM New York, NY, USA, 1998.

[4] Zamir, O. ve Etzioni, O.,"Grouper: a dynamic clustering interface to Web search results", *Computer Networks-the International Journal of Computer and Telecommunications Networkin*, 31(11):1361–1374, 1998.

[5] Wang, J.ve Li, R,."A New Cluster Merging Algorithm of Suffix Tree Clustering", *In Intelligent information processing III: IFIP TC12 International Conference on Intelligent Information Processing* (IIP 2006), September 20-23, Adelaide, Australia, 197, 2006.

[6] Zemberek 2 is an open source NLP library for Turkic languages. http://code.google.com/p/zemberek/, 9 Nisan 2012.