Türkçe Metinlerin Kümelenmesinde Farklı Kök Bulma Yöntemlerinin Etkisinin Araştırılması

Examining the Impact of Different Stemming Methods on Clustering Turkish Texts

Volkan Tunalı, Turgay Tugay Bilgin

Yazılım Mühendisliği Bölümü
Maltepe Üniversitesi, İstanbul
volkantunali@maltepe.edu.tr, ttbilgin@maltepe.edu.tr

Özet

Doğal dil kullanılarak oluşturulan metinler üzerinde veri madenciliği tekniklerinin uygulanabilmesi için yapısal olmayan bu veriyi yapısal biçime dönüştüren bir önişleme adımı uygulanır. Kök bulma, önemli bir metin önişleme yöntemidir. Bu çalışmada temel olarak Zemberek, Ek Çıkaran ve Sabit Önek olmak üzere üç farklı kök bulma yönteminin Türkçe metin kümeleme üzerindeki etkisi deneysel olarak araştırılmıştır. Bu yöntemlerle kümeleme kalitesi bakımından birbirine oldukça yakın sonuçlar elde edilmiş olmasına rağmen, Zemberek ve Sabit Önek 5 yöntemlerinin diğer yöntemlere göre daha iyi sonuçlar ürettiği gözlenmiştir. Kümeleme kalitesinin yanı sıra, sağladıkları yüksek oranda boyut indirgeme nedeniyle de Zemberek ve Sabit Önek 5 yöntemleri Türkçe metin kümeleme uygulamaları için tercih edilebilir yöntemlerdir.

Abstract

In order to apply data mining techniques on texts that are written in natural language, a preprocessing step is used to transform this unstructured data into structured format. Stemming is an important text preprocessing technique. In this study, we empirically examined the impact of essentially three stemming methods on clustering Turkish texts: Zemberek, Affix Stripping, and Fixed Prefix. Although very similar results were obtained with all these methods in terms of clustering quality, Zemberek and Fixed Prefix 5 methods produced better results when compared to the others. Besides clustering quality, Zemberek and Fixed Prefix 5 methods are preferable stemming methods for Turkish text clustering applications due to high dimensionality reduction rate they provide.

1. Giriş

Bilgi ve iletişim teknolojilerindeki gelişmelere ve Internet'in yaygınlaşmasına paralel olarak özellikle metin biçiminde üretilen ve depolanan bilgi miktarında önemli bir artış olmaktadır. Tahminlere göre iş dünyasına ilişkin bilginin %85'i sayısal ortamda metin biçiminde saklanmaktadır [1]. Oluşan büyük hacimli doküman yığınlarıyla başedebilmek için verimli ve ölçeklenebilir araçlara ve tekniklere ihtiyaç duyulmaktadır [2]. Metin Madenciliği, yararlı, ilginç ve daha önce bilinmeyen bilginin, bilgi işlem yöntemleri ve teknikleri ile metin halindeki büyük miktarda veriden elde edilmesidir. Metin Madenciliğinin bir dalı olan Metin Kümeleme, metin biçimindeki çok büyük miktarda verinin otomatik olarak işlenmesi ve organize edilmesi bakımından büyük önem taşımaktadır. Metin Kümeleme, doküman koleksiyonlarının doküman benzerliklerine bağlı olarak gözetimsiz ve otomatik biçimde gruplara ayrıştırılmasıdır. Kümeleme sonucunda, aynı küme içerisindeki dokümanlar benzer bir konuda olurken, farklı kümelerdeki dokümanlar farklı konularda içeriğe sahiptirler [3-5].

Metin dokümanları doğal dil kullanılarak oluşturulurlar ve genellikle yapısal değildirler. Bu nedenle, metinler üzerinde veri madenciliği tekniklerinin uygulanabilmesi için yapısal olmayan veriyi yapısal biçime dönüştüren bir önişleme adımına gereksinim vardır. Metin dokümanlarını önişlemeden geçirmek için metin madenciliğinde sıklıkla kullanılan yöntemlerden bazıları dizgeciklere ayırma (tokenization), durak sözcük filtreleme (stopword filtering), kök bulma (stemming) ve terim ağırlıklandırmadır (term weighting) [1].

Bilgi erişim ve metin madenciliği alanlarındaki araştırmalar çoğunlukla İngilizce metinler üzerinde yoğunlaşmaktadır. Ancak, literatürde önişlemenin bilgi erişim ve metin madenciliği üzerindeki etkisini Türkçe dili bakımından araştıran önemli çalışmalar da yer almaktadır. Bu çalışmalarda genellikle bilgi erişim uygulamalarında kök bulma ve durak sözcük filtreleme yöntemlerinin yüksek başarım artışı sağladığı gösterilmektedir [6-8]. Türkçe üzerine yapılmış başka bir bilgi erişim çalışmasında kök bulmanın arama sonuçlarını iyileştirmesine rağmen arama sözcüklerinde kök bulma uygulanmasının arama başarımını olumsuz etkileyebildiği belirtilmektedir [9]. Ayrıca, Türkçe metinlerin sınıflandırılmasında kök bulmanın etkisinin çok az olduğu da bildirilmektedir [10]. Tarafımızdan yapılan bir çalışmada kök bulmanın kümeleme kalitesinde her zaman belirgin bir iyileşme sağlamadığı gösterilmiş, ancak buna rağmen doküman-terim matrisinin boyutunda sağladığı yüksek indirgeme oranı nedeniyle kümeleme uygulamalarında kullanılması önerilmiştir [11]. Bilindiği kadarıyla Türkçe

metinlerin kümelenmesinde farklı kök bulma yöntemlerinin etkinliğini sistematik ve kapsamlı olarak araştıran bir çalışma bulunmamaktadır. Bundan hareketle, bu çalışmada, önişleme adımında çeşitli kök bulma yöntemlerinin kullanılmasının Türkçe metinlerin kümelenmesine olan etkisi deneysel olarak analiz edilmektedir.

Makalenin 2. bölümünde Türkçe metinlerin önişlenmesinde kullanılan çeşitli kök bulma yöntemleri anlatılmıştır. 3. bölümde, kullanılan veri setleri ve araçlarla birlikte deney düzeneği açıklanmıştır. Deneysel sonuçlar ve sonuçlara ilişkin tartışma 4. bölümde yer almaktadır. 5. bölümdeki sonuç ve önerilerle makale tamamlanmaktadır.

2. Kullanılan Türkçe Kök Bulma Yöntemleri

Kök bulma (stemming), sözcükleri basit hallerine çevirme işlemidir. Örneğin çoğul ekinin isimlerden atılması, fiil çekim eklerinin fiilden ayrılarak fiil kökünün ayrılması gibi işlemler kök bulma olarak adlandırılır. İngilizce lisanı için en yaygın kural tabanlı kök bulma algoritması Porter tarafından geliştirilmiştir ve halen en sık başvurulan yöntemdir [12]. Türkçe için ise çeşitli yöntemler geliştirilmiş olup bu bölümde bu yöntemler hakkında bilgi verilmiştir.

Bu makale çalışmasında ilk kök bulma yöntemi olarak Zemberek doğal dil işleme kütüphanesindeki fonksiyonlardan yararlanılmış olup, bu yöntem bu çalışma kapsamında Zemberek olarak adlandırılmıştır. Zemberek, Türk dilleri ve özellikle Türkçe için geliştirilmiş, açık kaynak, platform bağımsız ve genel amaçlı bir Doğal Dil İşleme kütüphanesidir [13]. Zemberek, sözcüklerin analizi ve kök bulma için kök ve ek sözlüğü kullanan sözlük tabanlı bir kök bulma yöntemi sunmaktadır.

Ek çıkaran kök bulucu (affix stripping stemmer) olarak adlandırılan ikinci kök bulma yöntemi, sözlük tabanlı kök bulma yöntemlerinden farklı olarak Türkçe'nin kural tabanlı yapısından yararlanmaktadır. Bu kök bulma yöntemi, sözcüklerin biçimbilimsel (morfolojik) yapısını, sözcüklere getirilen eklerin sondan başlanarak başa doğru sözcüklerden çıkarılması (affix stripping) yaklaşımıyla çözümleyen bir çözümleyiciye dayalı olarak geliştirilmiştir [14]. Sözlük gerektirmemesi ve bu sayede sözlük tabanlı kök bulma yöntemlerine göre daha az işlemci zamanı kullanması bu yöntemin üstün yanlarıdır.

Kullanılan son kök bulma yöntemi olan sabit önek (fixed prefix stemming) yöntemi pseudo bir kök bulma yöntemi olup temel olarak sözcüklerin ilk *n* karakterinin sözcüğün kökü olarak kullanıldığı basit ve hızlı bir yöntemdir. *n* karakter ve daha kısa sözcükler ise olduğu gibi kullanılırlar. Bu yöntem karakter *n*-gram yaklaşımına benzerlik göstermekte olup, bir sözcükten üretilebilecek *n*-gramlardan sadece ilkinin kullanıldığı daha basit bir halidir [8]. Bu yöntem ayrıca sözcük kesme (word truncation) yöntemi olarak da bilinmektedir [9].

3. Deney Düzeneği

Farklı kök bulma yöntemlerinin Türkçe metinlerin kümelenmesindeki etkisini araştırmak için Milliyet ve NTV olarak adlandırılan iki Türkçe metin veri seti kullanılmıştır.

Milliyet gazetesinin internet sitesinde 2006 yılında yayınlanmış 1455 Türkçe haber makalesinden oluşturulmuş olan Milliyet veri seti, ekonomi, politika ve spor olmak üzere 3 farklı kategoriden 485'er doküman içermektedir [15].

NTV veri seti, 2009-2010 yılları arasında NTV haber kanalının internet sitesinden tarafımızca toplanmış 19476 Türkçe haber makalesinden oluşmaktadır ve her biri çok farklı sayıda doküman içeren 9 kategoriye sahiptir. Bu kategoriler şunlardır: bilim, dünya, ekonomi, sanat, sağlık, spor, Türkiye, yaşam ve "yeşil haber". NTV veri setini oluşturan kategoriler son derece dağınık olup en küçük kategoride 535, en büyük kategoride 5806 doküman bulunmaktadır ve ortalama kategori büyüklüğü 2164 dokümandır [11].

Türkçe dokümanların önişlenmesi amacıyla tarafımızdan Java ile geliştirilen PRETO aracı kullanılmıştır [16]. PRETO, 2. bölümde bahsedilen kök bulma yöntemlerini desteklemekte olup ayrıca durak sözcük filtreleme (stopword filtering), istatistiksel terim filtreleme (statistical term filtering) ve *n*-gram oluşturma gibi çeşitli önişleme seçeneklerine de sahiptir. Araştırmalar sırasında sıklıkla Türkçe olmayan klavye düzenleri kullanılarak oluşturulan dokümanlara rastlanılmıştır. Örneğin, "çağrışım" yerine "cagrisim", "ışıldama" yerine "isildama" kullanılması gibi. Dolayısıyla, Zemberek kütüphanesindeki ilgili fonksiyonlar kullanılarak bu tür sözcüklerdeki hatalı yazılmış harfleri düzelten, deasciification olarak adlandırılan önişleme seçeneği uygulanmıştır.

Deneylerde 182 sözcük içeren bir durak sözcük listesi kullanılarak durak sözcük filtrelemesi uygulanmıştır. Ayrıca, kök bulma işleminden önce ve sonra, dokümanlar arasında ayırıcı özelliği olmadığı gerekçesiyle 3 karakterden daha kısa sözcükler, 3 dokümandan daha az sayıda dokümanda bulunan sözcükler ve dokümanların %95'inden daha fazlasında bulunan sözcükler filtrelenerek sözlüğe dahil edilmemişlerdir. Son olarak, terimler yaygın olarak kullanılan TF-IDF (terim frekansı – ters doküman frekansı) yöntemiyle ağırlıklandırılmıştır. Sabit önek kök bulma yöntemi sadece 3, 5 ve 7 karakterli olarak uygulanmış olup bu yöntemler Sabit Önek 3, Sabit Önek 5 ve Sabit Önek 7 olarak adlandırılmıştır. Farklı kök bulma yöntemleri uygulanarak elde edilmiş veri setlerine ait karakteristik özellikler Çizelge 1'de görülmektedir. Kök bulma uygulanarak Milliyet ve NTV veri setlerinden oluşturulan doküman-terim matrislerinin boyutlarında %50-%75 oranında indirgeme sağlanmış olup, bu durum Türkçe'nin eklemeli biçimbilimsel yapısından kaynaklanmaktadır. İngilizce metinlerde Porter kök bulma yönteminin uygulanmasıyla genellikle çok daha düşük bir boyut indirgemesi sağlanmaktadır.

Veri setlerinin kümelenmesi amacıyla klasik K-Means algoritmasının çok boyutlu metin veri setlerinin kümelenmesi için geliştirilmiş bir çeşitlemesi olan Küresel K-Means (Spherical K-Means) algoritması kullanılmıştır [17]. Küresel K-Means algoritması için tarafımızdan C++ ile geliştirilmiş olan kümeleme aracı kullanılmıştır [4].

Küresel K-Means algoritması başlangıç küme ağırlık merkezi seçimine son derece bağımlı bir algoritma olduğu için kümeleme uygulaması her bir veri setinde rastgele olarak ayarlanmış 10 farklı başlangıç koşuluyla 10'ar kez çalıştırılmış ve elde edilen sonuçların ortalaması alınmıştır. Böylece bu algoritmanın başlangıç koşullarına olan bağımlılığının etkisi en aza indirgenmiş ve kümeleme kalitesi bakımından karşılaştırmaların adaletli olması sağlanmıştır.

Çizelge 1: Veri setlerinin karakteristik özellikleri

Veri Seti	Doküman Sayısı	Kategori Sayısı	Doküman-Terim Matrisi Boyutu					
			Kök Bulma Yok	Zemberek	Ek Çıkaran	Sabit Önek 3	Sabit Önek 5	Sabit Önek 7
Milliyet	1.455	3	19.759	7.273	12.459	627	7.971	14.282
NTV	19.476	9	85.008	23.498	39.814	1.190	18.570	41.198

Deney sonuçlarında kümeleme kalitesi ölçütü olarak Saflık (Purity), Entropi (Entropy), Normalize Ortak Bilgi (Normalized Mutual Information) ve F-Ölçütü (F-Measure) kullanılmıştır [18]. Bu ölçütler tablolarda sırasıyla Saf, Ent, NOB ve FÖ olarak kısaltılmıştır.

4. Deneysel Sonuçlar

Milliyet veri seti 3, NTV veri seti ise 9 doğal küme yani kategori içermektedir. Her iki veri seti üzerinde küme sayısı parametresi k=5, k=10 ve k=20 olmak üzere 3'er kümeleme deneyi gerçekleştirilmiştir. Bir gerçek hayat kümeleme probleminde doküman koleksiyonundaki kümelerin gerçek sayısı genellikle bilinmemektedir. Bu nedenle, deneylerde keyfi olarak seçilen k değerleri için bulunan kümelerin bilinen kategorilerle en yüksek kesinlik derecesiyle eşleşmesi beklenmektedir.

Farklı kök bulma yöntemleriyle önişlemeden geçirilmiş Milliyet veri seti üzerinde farklı k değerleri için gerçekleştirilen kümeleme sonuçları Çizelge 2'de verilmiştir.

Çizelge 3'te farklı kök bulma yöntemleriyle önişlemeden geçirilmiş NTV veri seti üzerinde farklı k değerleri için gerçekleştirilen kümeleme sonuçları görülmektedir.

Çizelge 2: Milliyet veri seti için kümeleme sonuçları

k	Kök Bulma Yöntemi	Saf	Ent	NOB	FÖ
5	Yok	0,999	0,003	0,856	0,881
	Zemberek	**1,000**	**0,001**	**0,877**	**0,916**
	Ek Çıkaran	1,000	0,002	0,868	0,896
	S. Önek 3	0,969	0,121	0,743	0,849
	S. Önek 5	0,999	0,003	0,874	0,903
	S. Önek 7	0,999	0,003	0,851	0,871
10	Yok	0,958	0,081	0,663	0,781
	Zemberek	**0,995**	**0,013**	**0,744**	**0,807**
	Ek Çıkaran	0,987	0,027	0,714	0,788
	S. Önek 3	0,957	0,131	0,579	0,613
	S. Önek 5	0,993	0,017	0,733	0,809
	S. Önek 7	0,982	0,038	0,716	0,802
20	Yok	0,665	0,508	0,290	0,376
	Zemberek	**0,968**	**0,064**	**0,589**	**0,685**
	Ek Çıkaran	0,907	0,161	0,517	0,640
	S. Önek 3	0,953	0,137	0,477	0,409
	S. Önek 5	0,958	0,084	0,571	0,673
	S. Önek 7	0,884	0,196	0,492	0,640

Çizelge 3: NTV veri seti için kümeleme sonuçları

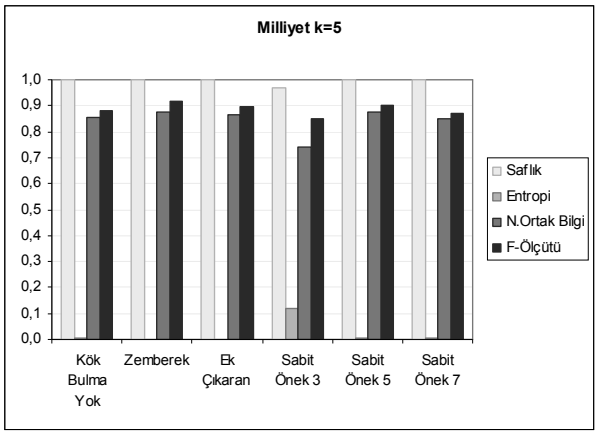
k	Kök Bulma Yöntemi	Saf	Ent	NOB	FÖ
5	Yok	0,618	0,514	0,440	0,538
	Zemberek	0,636	0,494	0,460	0,575
	Ek Çıkaran	0,616	0,507	0,445	0,545
	S. Önek 3	0,438	0,757	0,123	0,382
	S. Önek 5	**0,655**	**0,483**	**0,474**	**0,584**
	S. Önek 7	0,638	0,500	0,455	0,554
10	Yok	0,694	0,425	0,463	0,502
	Zemberek	0,706	0,411	0,473	0,532
	Ek Çıkaran	0,709	0,406	0,478	0,522
	S. Önek 3	0,434	0,751	0,109	0,295
	S. Önek 5	**0,708**	**0,403**	**0,480**	**0,516**
	S. Önek 7	0,699	0,417	0,469	0,516
20	Yok	0,741	0,366	0,451	0,447
	Zemberek	0,747	0,359	0,454	0,447
	Ek Çıkaran	0,750	0,359	0,455	0,447
	S. Önek 3	0,441	0,732	0,111	0,225
	S. Önek 5	0,746	0,359	0,455	0,441
	S. Önek 7	**0,750**	**0,354**	**0,460**	**0,438**

Şekil 1, 2, ve 3'te Milliyet veri setinde elde edilen kümeleme sonuçlarına ait grafikler sırasıyla k=5, k=10 ve k=20 için görülmektedir.

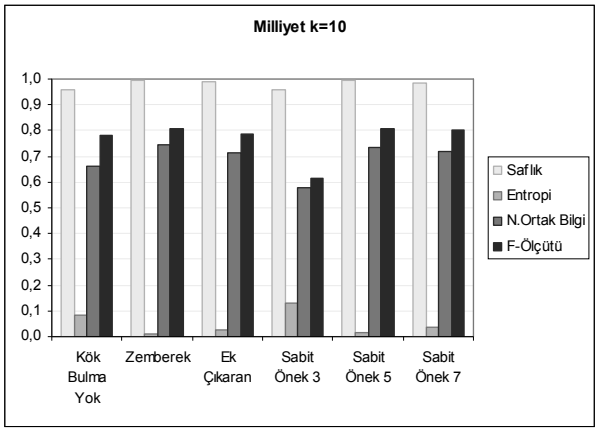
NTV veri setinde elde edilen kümeleme sonuçlarına ait grafikler k=5, k=10 ve k=20 için sırasıyla Şekil 4, 5, ve 6'da gösterilmiştir.

Deneylerde elde edilen yüksek saflık, düşük entropi, yüksek normalize ortak bilgi ve yüksek f-ölçütü değerlerine göre Milliyet veri setinde önişleme sürecinde kök bulma uygulanması kümeleme kalitesinde dikkate değer bir artış sağlamıştır. Kök bulma yöntemleri arasında Zemberek en iyi sonucu verirken, Sabit Önek 5 yönteminin neredeyse Zemberek kadar iyi sonuç verdiği gözlenmiştir. Sabit Önek 3 yöntemi ile en düşük kümeleme kalitesi elde edilmiştir.

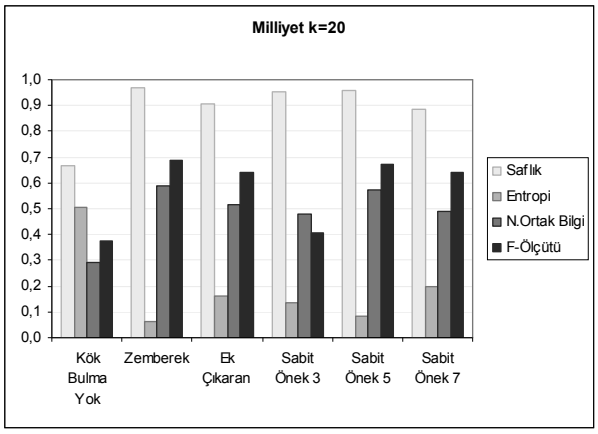
NTV veri seti üzerinde yapılan deneylerde elde edilen sonuçlara göre kümeleme kalitesi bakımından kök bulmanın belirgin bir yararının olmadığı görülmüştür. Birbirine çok yakın sonuçlar elde edilmiş olmakla birlikte kök bulma yöntemleri arasında ise genellikle Sabit Önek 5 yöntemi en yüksek kümeleme kalitesini sunmaktadır. Sabit Önek 3 yöntemi ile NTV veri setinde de son derece düşük bir kümeleme kalitesi elde edilmiştir.



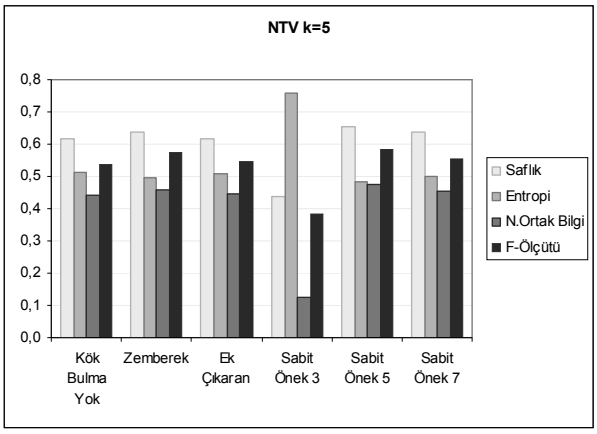
Şekil 1: Milliyet veri setinde *k*=5 için kümeleme sonuçları.



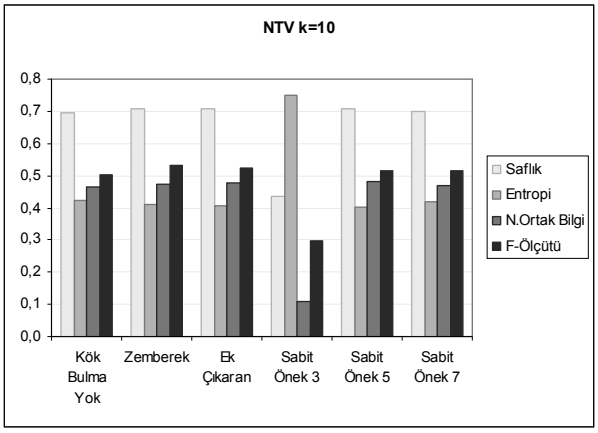
Şekil 2: Milliyet veri setinde *k*=10 için kümeleme sonuçları.



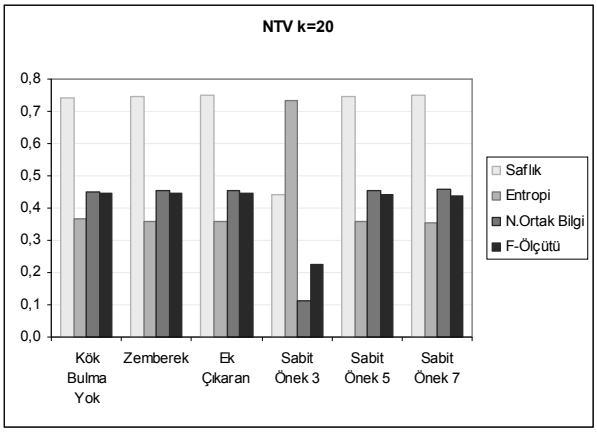
Şekil 3: Milliyet veri setinde *k*=20 için kümeleme sonuçları.



Şekil 4: NTV veri setinde *k*=5 için kümeleme sonuçları.



Şekil 5: NTV veri setinde *k*=10 için kümeleme sonuçları.



Şekil 6: NTV veri setinde *k*=20 için kümeleme sonuçları.

Kök bulmanın kümeleme kalitesine olan etkisinin yanı sıra kök bulma uygulanması sonucunda doküman-terim matrislerinin boyutlarında elde edilen yüksek indirgeme oranı da kümeleme uygulamaları bakımından büyük önem taşımaktadır. Farklı kök bulma yöntemleriyle değişen oranlarda boyut indirgeme sağlanmakla birlikte deneylerde kullanılan Milliyet ve NTV veri setlerinde genellikle en az yaklaşık %50 oranında bir indirgeme oranına ulaşılmıştır. Doküman-terim matrisinde elde edilen yüksek boyut indirgeme oranının etkisi doküman kümeleme uygulamalarında işlemci zamanı ve bellek gereksinimi bakımından son derece olumlu olmaktadır [2]. Dolayısıyla, kümeleme kalitesini belirgin bir biçimde arttırmasa bile sistem kaynaklarına olan gereksinim bakımından sağladığı olumlu katkı nedeniyle kök bulma uygulanması önerilmektedir.

Zemberek yöntemi ile elde edilen kümeleme kalitesi ve boyut indirgeme oranı, Zemberek'i iyi bir kök bulma seçeneği olarak ortaya çıkartmaktadır. Sabit Önek 5 yöntemi, dilbilimsel açıdan bir kök bulma yöntemi olmamasına rağmen Türkçe metin kümeleme uygulamalarında sağladığı kümeleme kalitesi ve boyut indirgeme oranı ile tercih edilebilir bir yöntem olarak görünmektedir. Sabit Önek 5 yönteminin temel olarak son derece basit ve hızlı oluşu da diğer bir tercih nedeni olabilir.

5. Sonuç ve Değerlendirme

Bu çalışmada farklı kök bulma yöntemlerinin Türkçe metin kümeleme üzerine etkisi deneysel olarak araştırılmıştır. Türkçe haber sitelerinden derlenen iki veri seti üzerinde kapsamlı deneyler yapılmıştır.

Deney sonuçlarına göre Türkçe doküman kümeleme uygulamalarında önişleme adımında kök bulma uygulanmasının kümeleme kalitesini her zaman iyileştirdiğine dair kesin bir bulgu yoktur. Bu çalışmada ele alınan kök bulma yöntemleriyle kümeleme kalitesi bakımından birbirine oldukça yakın sonuçlar elde edilmiş olup, Zemberek ve Sabit Önek 5 yöntemlerinin diğer yöntemlere göre daha iyi sonuçlar ürettiği gözlenmiştir. Kümeleme kalitesinin yanı sıra doküman-terim matrisinin boyutunda sağlanan yüksek indirgeme oranı nedeniyle Türkçe doküman kümeleme uygulamalarında kök bulma uygulanması kesinlikle önerilmektedir. Sağladıkları kümeleme kalitesiyle birlikte boyut indirgeme açısından da Zemberek ve Sabit Önek 5 yöntemleri tercih edilebilir yöntemler olarak görünmektedir.

6. Kaynaklar

[1] Hotho, A., Nürnberger, A. ve Paaß, G., "A Brief Survey of Text Mining", *LDV Forum-GLDV Journal for Computational Linguistics and Language Technology,* 20, 19-62, 2005.

[2] Han, J. ve Kamber, M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, USA, 2006.

[3] Feldman, R. ve Sanger, J., *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*, Cambridge University Press, 2007.

[4] Tunalı, V., "Metin Madenciliği İçin İyileştirilmiş Bir Kümeleme Yapısının Tasarımı Ve Uygulaması", Doktora Tezi, Fen Bilimleri Enstitüsü, Marmara Üniversitesi, İstanbul, 2011.

[5] Aliguliyev, R. M., "Clustering of Document Collection – a Weighting Approach", *Expert Systems with Applications,* 36, 7904-7916, 2009.

[6] Ekmekçioğlu, F. Ç. ve Willet, P., "Effectiveness of Stemming for Turkish Text Retrieval", *Program,* 34, 195-200, 2000.

[7] Sever, H. ve Bitirim, Y., "Findstem: Analysis and Evaluation of a Turkish Stemming Algorithm", *10th International Symposium on String Processing and Information Retrieval*, 2003, 238-251.

[8] Can, F., Koçberber, S., Balçık, E., Kaynak, C., Öcalan, H. Ç. ve Vursavaş, O. M., "Information Retrieval on Turkish Texts", *Journal of the American Society for Information Science and Technology,* 59, 407-421, 2008.

[9] Uzun, E., "Examining the Impacts of Stemming Techniques on Turkish Search Results by Using Search Engine for Turkish", *2nd International Symposium on Computing in Science and Engineering*, 2011, 33-39.

[10] Torunoğlu, D., Çakırman, E., Ganiz, M. C., Akyokuş, S. ve Gürbüz, M. Z., "Analysis of Preprocessing Methods on Classification of Turkish Texts", *International Symposium on Innovations in Intelligent Systems and Applications*, 2011, 112-117.

[11] Tunalı, V. ve Bilgin, T. T., "Examining the Impact of Stemming on Clustering Turkish Texts", *International Symposium on Innovations in Intelligent Systems and Applications*, 2012.

[12] Porter, M. F., "An Algorithm for Suffix Stripping", *Program,* 130-137, 1980.

[13] Akın, A. A. ve Akın, M. D., "Zemberek, an Open Source Nlp Framework for Turkic Languages", 2007.

[14] Eryiğit, G. ve Adalı, E., "An Affix Stripping Morphological Analyzer for Turkish", *International Conference on Artifical Intelligence and Applications*, 2004, 299-304.

[15] Işık, M., "Bölünmeli Kümeleme Yöntemleri İle Veri Madenciliği Uygulamaları", Y. Lisans Tezi, Fen Bilimleri Enstitüsü, Marmara Üniversitesi, İstanbul, 2006.

[16] Tunalı, V. ve Bilgin, T. T., "Preto: A High-Performance Text Mining Tool for Preprocessing Turkish Texts", *International Conference on Computer Systems and Technologies*, 2012, 134-140.

[17] Dhillon, I. S. ve Modha, D. S., "Concept Decompositions for Large Sparse Text Data Using Clustering", *Machine Learning,* 42, 143-175, 2001.

[18] Strehl, A., "Relationship-Based Clustering and Cluster Ensembles for High-Dimensional Data Mining", Doktora Tezi, The University of Texas at Austin, 2002.