Biçimbilime Dayalı Doküman Sıkıştırma
Morphology Based Text Compression

[1]Hayriye Göksu, [2]Banu Diri

Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, İstanbul
[1]hayriyegoksu@gmail.com, [2]banu@ce.yildiz.edu.tr

Özetçe

İnternet' in yaygınlaşmasıyla elektronik ortamdaki doküman sayısı oldukça artmıştır. Gittikçe artan bu bilgiye daha kolay ve hızlı erişmek amacıyla doküman sıkıştırma önem kazanmaktadır. Son yıllarda, doküman sıkıştırma alanında yapılan çalışmaların bir kısmı, dilin biçimbilimsel yapısını kullanmayı amaçlayan çalışmaları kapsamaktadır.

Bu çalışmada, Türkçe ve İngilizce dokümanların sıkıştırılma verimlerinin belirlenmesinde farklı ayrıştırma yöntemleri ve bu yöntemlerin sıkıştırma oranına etkileri araştırılmıştır. Dokümanlar Türkçe ve İngilizce' nin biçimbilimsel yapısı kullanılarak ayrıştırılmıştır. Sonraki aşamada ayrıştırılan dokümanlardaki yapılara Huffman algoritması ile sıkıştırma işlemi uygulanmıştır. Sonuçta, 10 farklı ayrıştırma tekniği oluşturulmuş ve bunlar ile farklı külliyatlar üzerinde denemeler yapılmıştır.

Abstract

With the rapid growth of online information, the number of documents in electronic media is very common increased. Easy and quick access to this information gets more important for the purpose of text compression. In recent years, a portion of the work in the field of text compression covers study aimed to the morphological structure of the language.

In this study, Turkish and English documents are compressed in the determination of the different decomposition methods and efficiency, this method has been to investigate the effects of compression. Turkish and English documents are parsed by using morphological structure. The next stage in the parsed document structure is applied to the compression process with Huffman compression method. As a result, created 10 different parsing techniques with which attempts were made on a different corpus.

1. Giriş

Zamandan ve yerden kazanç sağlamak için, kayıplı ve kayıpsız yöntemlerle veri üzerinde işlem yapmak anlamına gelen veri sıkıştırma, verinin sayısal ortamda daha fazla yer tutup, maliyet artımına yol açtığında ve belli bir zaman diliminde iletişim kanallarından transfer edilen verinin miktarı söz konusu olduğunda çok büyük bir önem kazanmaktadır. Bilişim teknolojilerindeki gelişmeler veri sıkıştırma yöntemlerinin hem yazılım hem de donanım elemanları ile gerçekleştirilmesini sağlamıştır. Veri sıkıştırmanın temel özellikleri, var olan verinin daha az yer kaplayacak şekilde yeniden düzenlenmesi, zaman ve boyuttan kazanım sağlayarak maliyetin minimize edilmesi olarak özetlenebilir.

Veri sıkıştırma işlemleri temel olarak kayıpsız ve kayıplı sıkıştırma yöntemleri olarak iki gruba ayrılır. Ses, görüntü ve video gibi verilerin sıkıştırma işlemleri kayıplı olurken, dokümanların sıkıştırılması kayıpsız olmaktadır. Kayıpsız sıkıştırma algoritmalarını istatistiksel yöntemler ve sözlük tabanlı yöntemler olarak iki gruba ayırabiliriz. LZ77, LZ78, LZW, Huffman Kodlama, Aritmetik Kodlama, Run-Length Kodlama ve Dinamik Markov Zinciri bu algoritmalara örnek olarak verilebilir[1].

Bu çalışmada Türkçe dilinin biçimbilimsel yapısının sıkıştırılmaya uygunluğu araştırılmaktadır. Bu araştırma esnasında karşılaştırma yapmak amacıyla elverdiği ölçüde İngilizce dili için de benzer bir çalışma yapılmıştır. Türkçe dilinin yapısını kullanan veya dilden bağımsız olarak n-gram' ları kullanarak [2] yapılan literatürde bazı çalışmalar mevcuttur. Diri [3], Türkçe dilinin biçimbilimsel yapısından yararlanarak doküman içerisindeki her kelimeyi kök-ek, hece ve karakter gibi farklı sembollere ayırıp kayıpsız sıkıştırma gerçekleştirmiştir. Bu makale [3] ' in daha genişletilmiş bir halidir. Çelikel [4] ise hem Türkçe hem de İngilizce için sabit ve esnek kelime tabanlı bir yöntem, Rueda [5] ise İkili Arama Ağacı ve Fano kodlamasını kullanarak yeni bir sıkıştırma yöntemi geliştirmiştir. Topaloğlu [6], İngilizce dili için n-gram' ları kullanarak dokümanların kayıpsız olarak sıkıştırılmasını incelemiştir.

Çalışmanın ikinci bölümünde Entropi Kodlamadan, üçüncü bölümde deneysel sonuçlardan ve son bölümde de çalışmanın değerlendirilmesi yapılmıştır.

2. Entropi Kodlama

Bu çalışmada, entropy kodlamada en çok tercih edilen Huffman Kodlama yöntemi seçilmiştir. *Entropi(E-Entropy)* ve *Artıklık(R-Redundance)* kavramları bilgi teorisinde önemlidir. a_i gibi tek bir sembolün entropy' si, $-P_i \log_2 P_i$ olarak ifade edilmekte olup, P_i, doküman içerisindeki a_i sembolünün oluşma olasılığı olarak tanımlanmaktadır. Doküman içerisinde n tane farklı sembol var ise, a_i sembolü için ihtiyaç duyulan ortalama bit sayısı Eşitlik 1' deki gibi ifade edilmektedir [7].

$$-\sum_{1}^{n} P_i \log_2 P_i \qquad (1)$$

Eşitlik 1 ile ortalama ihtiyaç duyulan bit sayısı ifade edilmiştir. a_i sembolünün entropy' sine bağlı olan P_i ifadesi, tüm n olasılıkları eşit olduğunda en küçük değerini almaktadır. Bu da gerçek verideki R artıklığını tanımlamak için kullanılmaktadır. Bu ifade entropi ve en küçük entropi arasındaki fark alınarak Eşitlik 2' deki gibi tanımlanmaktadır.

$$R = \left(-\sum_{1}^{n} P_i \log_2 P_i \right) - \log_2 \left(1/P \right) = -\sum_{1}^{n} P_i \log_2 P_i + \log_2 n \qquad (2)$$

Huffman algoritmasında yarı adaptif kodlama tercih edilmiştir.

978-1-4244-9671-6/10/$26.00 ©2010 IEEE

İlk adımda sıkıştırılacak olan doküman içerisiden, seçilen ayrıştırma yöntemine göre karakter, kök, ek, hece, isim, sıfat, fiil, n-gram, vs. gibi sembollerin kullanım frekansları hesaplanır. Daha sonra kodlama ağacı oluşturularak sembollerin Huffman kodları elde edilir. Sıkıştırılmış dosyanın başına da kod çözme sırasında ihtiyaç duyulacak olan başlık (header) bilgisi eklenir. Başlık bilgisi içerisinde sembol ve onların kullanım sıklıkları gibi bilgiler gönderilir. Ancak sıklık bilgisinin gönderilmesi çok yer tutacağından ve kod çözme işleminde ağacın tekrar oluşturulması zaman alacağından bu bilgilerin yerine başlıkta oluşturulan kodlama ağacının kendisi gönderilir. Ağaçtaki her sembol ve sembole ait kod değerleri sol-düğüm-sağ (left-node-right-LNR) notasyonu ile okunarak bir bit katarı oluşturularak gönderilir. Başlığın oluşturulması hakkındaki detaylı bilgiye [3]' ten ulaşılabilinir. Bu çalışmadaki amaç bir doğal dilin biçimbilimsel yapısının sıkıştırma performansına olan etkisini incelemektir.

3. Türkçe' nin Biçimbilimsel Yapısı

Biçimbilgisi, bir dilin kelimelerini, kelime yapılarını, türeme yollarını, çekim bilgilerini inceleyen dilbilgisinin bir koludur. Biçimbilimsel yapısı incelenecek olan dil olarak Türkçe ve İngilizce seçilmiştir. Türkçe Ural-Altay dil ailesine bağlı olup, sondan eklemeli, İngilizce ise Hint-Avrupa dil ailesine bağlı olup, bükümlü bir dildir. Bu nedenle Türkçe ile İngilizce arasında sözdizim (sentaks), anlambilim (semantic), sesbilim (fonetic), sözcükbilim (lexicology), biçimbilim (morfolojic) yönlerinden önemli farklar vardır. Biz bu çalışmada Türkçe' deki her kelimeyi karakter, hece, isim ve isme eklenen ekler, fiil ve fiile eklenen ekler, sıfat ve sıfata eklenen ekler, ekleri tek parça, ek, hece veya karakter halinde ayrı ayrı ele alarak, ayrıca dokümanın 2-gram' larını da çıkartarak inceledik.

Türkçe dokümanların biçimbilimsel yapılarına ayırma işleminde *Zemberek* [8] isimli açık kaynak kodlu program kullanılmıştır. Kelimenin heceleri, kökü, ek/ekleri, isim, fiil veya sıfat olduğu bu araç yardımıyla bulunmaktadır. Ancak bazı çözümlemelerde problemler yaşanmaktadır. Bunlardan biri hece düşmelerinin göz ardı edilmesidir. Örneğin, '*bağrına*' kelimesi çözümleme işlemine tabi tutulduğunda, kök "*bağır*", "*ına*" eki olarak analiz edilmektedir. Kökü ve eki bu şekilde kodlayacak olursak kodçözme işlemi sırasında orijinal metnin aynısını elde edemeyiz. Yazılan program içerisinde bu gibi durumlar saptanarak Zemberek' ten gelen sonuçlar kontrol edilerek gerekli düzeltmeler yapılmıştır.

İngilizce dokümanların biçimbilimsel yapılarına göre ayırma işleminde *Porter Stemming* algoritması [9] kullanılmıştır. Bu algoritma İngilizce kelimelerde yer alan biçimbilimsel ve çekimsel ekleri ayırma işlemini gerçekleştirmektedir. Bu çalışmada İngilizce' deki her kelime karakter, kök ve eki ek, kök ve eki karakter olmak üzere farklı şekillerde ayrıştırılarak kodlanmıştır.

Son aşamada da hem Türkçe hem de İngilizce dokümanların 2-gram' larını çıkaran bir kod yazılmış ve sadece 2-gram' lara göre kodlama gerçekleştirilmiştir.

4. Deneysel Sonuçlar

Bu bölümde öncelikli olarak denemelerin yapıldığı 8 farklı külliyat oluşturulmuştur. Bu külliyatlardan K1-K7 arası Türkçe için, K8 ise İngilizce için hazırlanmıştır. K1 külliyatı; popüler hayat, güncel, ekonomi, spor, dünya ve sağlık gibi farklı konularda yazan 18 erkek yazara ait 376 farklı dokümandan, K2 külliyatı; sadece güncel konularda yazan 7 erkek yazara ait 237 dokümandan, K3 külliyatı; popüler hayat, güncel, ekonomi, spor, dünya ve sağlık gibi farklı konularda yazan 18 erkek ve 4 kadın yazara ait 516 dokümandan, K4 külliyatı; siyaset konusunda yazan 12 erkek ve 1 kadın yazara ait 271 dokümandan, K5 külliyatı; popüler hayat, güncel, ekonomi, spor, dünya ve sağlık gibi farklı konularda yazan 4 kadın yazara ait 140 dokümandan, K6 külliyatı; güncel konularda yazan 2 kadın yazara ait 98 dokümandan, K7 külliyatı; Türkiye Anayasa metni ile 6 öykü ve 1 romana ait 46 dokümandan, ve İngilizce için K8 külliyatı ise güncel konularda yazılmış 290 farklı dokümandan oluşturulmuştur. K1-K6 arasındaki külliyatlar günlük gazetelerin köşe yazılarından toplanmıştır. Farklı külliyatlar oluşturulmasının sebebi sadece kadın yazarların veya erkek yazarların ya da tek bir konuda veya farklı konularda yazılmış dokümanların biçimbilimsel analizi yapılarak sıkıştırıldığında performansın nasıl etkilendiğini görmektir. Herbir külliyat için 64Kb, 256Kb ve 1024Kb' lık dokümanlar oluşturulmuştur.

Türkçe için tüm külliyatlara 10 ayrıştırma yöntemi uygulanmıştır. Bunlar sırasıyla KET-kelimenin kökü ve ekler ek bazında, H-kelimenin heceleri, K-kelimeyi oluşturan karakterler, İET-isim kökü ve ekler tek parça, FET-fiil kökü ve ekler tek parça, SET-sıfat kökü ve ekler tek parça, İEK-isim kökü ve ekler karakter bazında, FEK-fiil kökü ve ekler karakter bazında, SEK-sıfat kökü ve ekler karakter bazında, 2-Gr, kelimenin bigram olarak analiz edilmesidir. Bu ayrıştırma yöntemlerine örnek verecek olursak:
KET: kitaplarda (kitap, -lar, -da)
H: kitaplarda (ki, -tap, -lar, -da)
K: kitaplarda (k, i, t, a, p, l, r, d)
İET: kitaplarda (kitap, -larda)
FET: yüzecek (yüz, -ecek)
SET: güzellik (güzel, -lik)
İEK: kitaplarda (kitap, l, a, r, d)
FEK: yüzecek (yüz, e, c, k)
SEK: güzellik (güzel, l, i, k)
2-Gr: soru (so, or, ru)

Tablo 1' de ayrıştırma yöntemlerinden 7 tanesinden alınan sıkıştırma oranlarına yer verilmiştir. İEH, FEH ve SEH ayrıştırma yöntemleri diğerlerine yakın veya daha kötü sonuçlar verdiğinden araştırmanın dışında tutulmuştur. Sıkıştırma oranı Eşitlik 3' deki gibi hesaplanmaktadır.

$$So = 100 * \left(1 - \left(cikisdosyaboyu / girisdosyaboyu\right)\right) \qquad (3)$$

Yedi külliyatta görebildiğimiz şey hangi ayrıştırma yöntemini kullanırsak kullanalım dosya boyutu arttıkça sıkıştırma oranı da artmaktadır. Kısaca sıkıştırma oranı, dosya boyu ile doğru orantılıdır. K1 ve K2 külliyatlarının her ikisi de erkek yazarlardan oluşmuştur. Aynı veya farklı konularda yazılmış olsalar da sıkıştırma oranları birbirine çok yakındır. En başarılı yöntem kök ve ekler ek olarak değerlendirildiğinde olmuştur. Ortalama sıkıştırma oranı başarısı K1 için %48, K2 için de %47.3' tür. Bunu izleyen diğer yöntem de her iki külliyat için hecelerin sıkıştırılmasıdır. K3 ve K4 külliyatları hem erkek hem de kadın yazarlardan oluşmakta, genel ve sadece tek bir konudan oluşmaktadır. K3 külliyatında ortalama en başarılı yöntem KET olmasına karşılık, isim ve ekler karakter bazında

kodlanmış olan İEK yönteminde %56' lık bir başarı alınmıştır.

Bu bize kadın yazarların erkeklere göre yazılarında daha fazla isim kullandığını ve bunların tekrarladığını göstermektedir. K4 külliyatında en yüksek başarı hem ortalamada hem de dosya bazında KET yönteminde olmuştur. Aynı konularda yazılmış olan K2 ve K4 külliyatları arasında 1024Kb' lık dosyadaki farkın yine kadın yazarlardan kaynaklandığı düşünülmektedir. Başarı daha yüksek olduğu için tekrar eden aynı kök sayısı fazladır. Bu da kadın yazarların daha az kelime dağarcığı kullandığını göstermektedir. Yine K3 ve K4 külliyatlarında ikinci başarılı yöntem H' dir. K5ve K6 külliyatları ise sadece kadın yazarlara aittir ve her ikisinde en başarılı yöntem KET olmuştur. Diğer yöntemlerin başarıları birbirlerine oldukça yakındır. Külliyatlar ister karışık, isterse tek bir konuda yazılmış olsa bile kadın yazarların yazım stilleri birbirlerine oldukça yakındır. K7 külliyatı ise anayasa, öykü ve romandan oluşmakta olup, içerik olarak gazete köşe yazılarından oldukça farklıdır. Bu külliyat için dosya bazında en başarılı ayrıştırma yöntemi H olmuştur. Ortalama başarıda ise KET ve H ayrıştırma yöntemleri başarılıdır. Genelde tüm ayrıştırma yöntemlerinde 64Kb gibi kısa dosya boyutunda başarı sonuçları düşüktür ancak, külliyat K7' de en yüksek sıkıştırma oranı alınmıştır. Bunun nedeninin K7 içerisinde yer alan dokümanların daha edebi yapıtlar olduğu ve dilin güzel kullanılmış olduğu düşünülmektedir.

Bu çalışmanın benzeri İngilizce için elimizde bulunan dil çözümleyicisinin imkanları dahilinde yapılmıştır. Metinler biçimbilimsel yapılarına göre, ekler ek bazında olmak üzere kök–ek(İKET), ekler karakter olmak üzere kök–ek(İKEK), karakter(İK) ve bigram(2-Gr) olarak 4 farklı şekilde ayrıştırılmışlardır. Güncel konularda yazan farklı kişilere ait 290 farklı dokümandan oluşan K8 külliyatı için başarı sonuçları Tablo 2' de verilmektedir. İngilizce' de en başarılı ayrıştırma yöntemi kök ve ekler karakter halinde kodlanan İKEK yöntemidir. Ortalama başarısı %52.7' dir. İkinci ayrıştırma yöntemi de karakter bazında yapılan İK kodlama yöntemidir.

Bu iki dili karşılaştıracak olursak, KET ve İKET ayrıştırma yöntemlerinde Türkçe İngilizce' ye göre daha yüksek sıkıştırma başarısı vermiştir. Türkçe'nin sondan eklemeli bir dil olması sebebiyle toplam kök sayısı, İngilizce diline göre daha azdır. Türkçe' de kelime köküne eklenen yapım ekleri ile farklı kelimeler elde ediliyorken, İngilizce için böyle bir durum söz konusu değildir. Karakter bazında ayrıştırma yapan K ve İK yöntemlerini karşılaştırdığımızda ise İngilizce dili için daha başarılı sıkıştırma performansı elde edilmiştir. Bigram ayrıştırma yöntemi K4 külliyatı haricinde, İngilizce dilinde daha iyi sıkıştırma performansı vermiştir.

Tablo 1. Türkçe için 8 farklı külliyata uygulanan 7 ayrıştırma yönteminin sıkıştırma oranları(%)

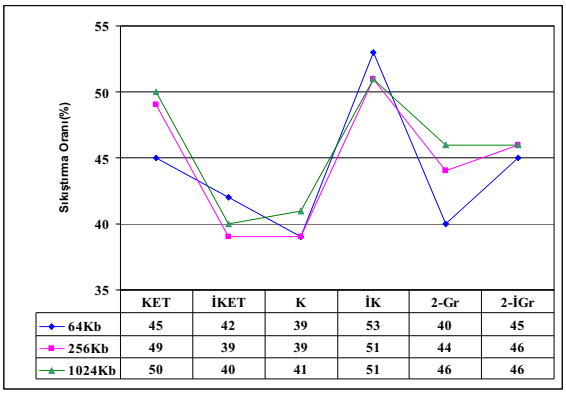
	KET	**H**	**K**	**İEK**	**FEK**	**SEK**	**2-Gr**
64Kb	45	43	39	37	38	39	40
256Kb	49	47	39	42	39	39	44
1024Kb	**50**	48	40	43	41	39	45
K1(Ort.)	**48**	46	39.3	40.7	39.3	39	43
64Kb	44	43	39	37	38	39	38
256Kb	48	46	39	42	39	39	43
1024Kb	**50**	48	40	43	39	40	46
K2(Ort)	**47.3**	45.7	39.3	40.7	38.7	39.3	42.3
64Kb	45	43	39	37	38	39	40
256Kb	49	47	39	42	39	39	44
1024Kb	50	48	41	**56**	41	40	46
K3(Ort)	**48**	46	39.7	45	39.3	39.3	43.3
64Kb	44	43	39	36	38	39	41
256Kb	55	54	47	49	47	47	52
1024Kb	**61**	60	51	56	52	52	56
K4(Ort)	**53.3**	52.3	45.7	47	45.7	46	49.7
64Kb	44	43	39	36	38	39	40
256Kb	48	47	39	41	39	39	45
1024Kb	**51**	49	39	42	39	39	47
K5(Ort)	**47.7**	46.3	39	39.7	38.7	39	44
64Kb	43	43	39	35	37	39	41
256Kb	48	47	40	41	39	40	45
1024Kb	**49**	48	41	43	40	40	47
K6(Ort)	**46.7**	46	40	39.7	38.7	39.7	44.3
64Kb	45	44	40	37	38	40	42
256Kb	49	49	41	42	40	41	45
1024Kb	50	**51**	41	44	42	41	48
K7(Ort)	**48**	**48**	40.7	41	40	40.7	45

K4 külliyatının başarılı olmasının sebebi tek bir konu hakkında yazıldığında elde edilen bigram' ların sayısı daha az ve kullanım frekans değerlerinin yüksek olmasıdır.

Tablo 2. İngilizce için hazırlanan külliyata uygulanan 4 ayrıştırma yönteminin sıkıştırma oranları(%)

	İKET	**İKEK**	**İK**	**2-İGr**
64Kb	42	**54**	53	45
256Kb	39	52	51	46
1024Kb	40	52	51	46
K8(Ort)	40.3	**52.7**	51.7	45.7

Türkçe' de başarılı sonuç vermediğinden bu çalışmada yer verilmeyen kök ve ekler karakter olarak alınan İKEK ayrıştırma yöntemi İngilizce için %52.7 gibi ortalama başarı vermiştir.



	KET	İKET	K	İK	2-Gr	2-İGr
64Kb	45	42	39	53	40	45
256Kb	49	39	39	51	44	46
1024Kb	50	40	41	51	46	46

Şekil 1. Her iki dilin 3 farklı ayrıştırma yönteminin karşılaştırılması

Şekil 1' de her iki dil için, kök ve ekler ek halinde, karakter ve bigram olarak ayrıştırılmış, 3 dosya boyutundaki sıkıştırma performansları karşılaştırmalı olarak verilmektedir. Şekil 1' de görüldüğü gibi kök ve ekler ek halinde olan ayrıştırma yöntemi Türkçe dili için daha başarılı olmuşken karakter tabanlı ayrıştırma İngilizce dili için oldukça yüksek bir performans sergilemiştir. Bigram ayrıştırma yöntemi ise dosya boyutu arttıkça her iki dil için de yaklaşık aynı sıkıştırma başarısını vermektedir.

5. Sonuçlar

Bu çalışmada, Türkçe ve İngilizce dokümanlarda dilin biçimbilimsel yapısı kullanılarak farklı ayrıştırma yöntemleri ile entropi kodlama kullanılarak kayıpsız sıkıştırma uygulanmıştır. Türkçe için yedi farklı külliyat, genel veya tek bir konuda, erkek, kadın veya karışık yazarlardan oluşturulmuştur. Biçimbilimsel olarak kelime kökü ve köke eklenen ekler ek, karakter, hece, isim ve isme eklenen ekler tek parça, karakter, hece, fiil ve fiile eklenmiş ekler tek parça, karakter, hece, sıfat ve sıfata eklenmiş ekler tek parça, karakter, hece bazında ve bigramlar olmak üzere yapılmıştır. İngilizce içinse genel bir külliyat oluşturulmuş, sadece kelime kökü ve köke eklenen ekler ek, kök ve köke eklenen ekler karakter bazında, karakter ve bigram' lar olmak üzere dört farklı ayrıştırma yöntemi tercih edilmiştir. Herbir ayrıştırma yöntemiyle elde edilen sıkıştırma sonuçlarının karşılaştırılması yapılmıştır. Hangi ayrıştırma yönteminin hangi dilde daha başarılı olduğu tartışılmıştır.

Bu çalışmanın amacı yüksek sıkıştırma oranına sahip bir yöntem bulmak değildir. Amaç doğal bir dilin biçimbilimsel yapısı kullanılarak bir sıkıştırma yapılırsa nasıl bir sonuç alınacağı ve dillerin hangi özellikleri avantaj sağlarken hangi özelliklerinin dezavantaj sağlayacağını görmektir. Elde edilen deney sonuçlarının daha iyileştirilmesi için dile bağlı kullanılan biçimbilimsel analiz yapan araçların serbest kullanıma açılmasıdır. Yine çalışma içerisinde sıkıştırma sürelerinden bahsedilmemiştir. Çünkü biçimbilimsel analizin yapılması ve kayıpsız sıkıştırmayı gerçekleştirebilmemiz için analiz sonucundaki çıktıların dilin özelliklerine bağlı olarak tekrardan işleme sokulması gerekmektedir. Sıkıştırılmış dosyanın geri açılması ise daha hızlı olarak gerçekleşmektedir.

Bu çalışma, bugüne kadar Türkçe dokümanların biçimbilimsel yapılarını kullanarak sıkıştırma işleminin yapılmasını sağlayan en kapsamlı çalışmadır.

6. Kaynakça

[1] Nelson, M., The Data Compression Book, NewYork, USA, (1996)

[2] Çebi, Y., Dalkılıç, G., "Turkish Word N-gram Analysing Algorithms for a large Scale Turkish Corpus-TurCo", *IEEE International Conference on Information Technology*, Vol:2, pp.226-240, (2004)

[3] Diri, B., "A Text Compression System Based on the Morphology of Turkish Language", *International Symposium on Computer and Information Sciences (ISCIS) XV*, 11-13 October, Istanbul (2000)

[4] Çelikel, E., Dalkılıç, M.E. ve Dalkılıç, G., "Word-Based Fixed and Flexible List Compression", *Computer and Information Sciences-ISCIS,* LNCS 3733, Springer Verlag, pp. 780-790, (2005)

[5] Rueda, L., Oommen, B.J., "Efficient Adaptive Data Compression Using Fano Binary Search Trees", *Computer and Information Sciences-ISCIS*, Vol. 3733, p. 768-779, (2005)

[6] Topaloğlu, U. ve Bayrak, C., "Polymorphic Compression", *Computer and Information Sciences-ISCIS*, Vol. 3733, p. 759-767, (2005)

[7] Salomon, D., Data Compression, Springer-Verlag, NewYork, (1997)

[8] https://zemberek.dev.java.net/

[9] http: //tartarus.org/~martin/PorterStemmer/