**Huffman Kodlama ile Metin Verilerini Sıkıştırma**

**BLM6106 – Veri Sıkıştırma**

Recep Furkan Koçyiğit

22501048

furkan.kocyigit@std.yildiz.edu.tr

Bilgisayar Mühendisliği Bölümü

Elektrik Elektronik Fakültesi, Yıldız Teknik Üniversitesi

**Özet**

Bu ödevde, Huffman Kodlama ile metin verileri üzerinde bir sıkıştırma yapılacak ve daha sonrasında sıkıştırılan dosya açılarak doğru bir şekilde sıkıştırma olduğu teyit edilecektir. Sıkıştırma algoritması metindeki sembollerin frekansını bulurken, beraber kullanılan ikili sembollerin de frekansı kullanılacaktır. Bu sistemin geçekleştirilmesi için programlama dili olarak Python kullanılmıştır.

**Giriş**

Ödev kapsamında Yıldız Teknik Üniversitesi Kemik Doğal Dil İşleme Grubuna ait 69 Köşe yazarına ait yazılar sıkıştırma algoritması için kullanılmıştır. 69 köşe yazarına ait çeşitli sayıda yazı her yazar için birleştirilmiştir. 69 farklı köşe yazısı sıkıştırılarak metin özelinde sonuçlar alınması önlenmiştir. Bunun dışında ikili sembol oranı için verilen metnin uzunluğunun yüzde kaçı oranında olması gerektiğinin bulunması için farklı denemeler yapılmıştır. Tüm metnin %0,1 ile %2 arasındaki sıklıklarda ikili sembollerde yapılan bu denemede %1' den sonra sıkıştırma oranı değişmemiştir. Bunun sebebi oranın ikili sembol bulunma oranının yüksek olmasından dolayı tek sembol kullanılarak sıkıştırma yapılmasıdır. Yani, kullanılan veri kümesi için tek sembol kullanılan Huffman Kodlama sıkıştırması ikili sembol kullanılarak yapılan sıkıştırmadan daha başarılı sonuç üretmiştir.

**Yöntem**

Bu sistem aşağıdaki ana modüllerden oluşmaktadır:

1. **Metin Verilerinin Birleştirilmesi:** Kemik Grubuna ait köşe yazı verilerinin köşe yazına göre gruplanması.
2. **Frekans Tablosunun Oluşturulması:** Verilen dosya ve oran için tekli ve verilen orandan yüksek oranda olan ikili sembollerin bir tabloda saklanması.
3. **Huffman Ağacının Oluşturulması:** Verilen frekans tablosuna göre Huffman ağacının oluşturulması için Min-Heap kullanılmıştır. Algoritma şu şekilde çalışmaktadır:
    - Her sembolün ve frekansının olduğu bir Min-Heap oluşturulması.
    - Min-Heap' ten en küçük iki frekanslı düğümün çıkarılması.
    - Çıkarılan bu iki düğümün frekanslarının toplamını tutan, sol ve sağ düğümleri çıkarılan düğümler olan bir düğüm oluşturulması.
    - Sadece bir düğüm kalana kadar bu işlemin tekrar edilmesi.
4. **Sembol-Bit Tablosunun Oluşturulması:** Oluşturulan Huffman Ağacı kullanılarak metni içeren sembollere karşılık bu sembollerin sıkıştırılmış karşılıklarını ikili kodlama olarak tutan bir tablo oluşturulur.
5. **Metnin İkili Kodlama ile Kodlanması:** Verilen metin, sembol-bit tablosu kullanılarak 0 ve 1 ikilileri ile kodlanır.
6. **Huffman Ağacının Kodlanması:** Oluşturulan Huffman Ağacı sıkıştırılan dosyanın kullanımı için gerektiği için ön-sıralı (pre-order) okunmuş ve başlık olarak kodlanmıştır.
7. **Sıkıştırma Aşaması:** Verilen dosya ismi ve ikili sembol oranına göre dosyanın okunması, frekansların bulunması, Huffman ağacının oluşturulması, sembol-bit tablosunun oluşturulması, metnin 0 ve 1 olarak kodlanması, Huffman Ağacının ve şifrelenmiş verinin uzunluğunun başlık olarak kodlanması ve bu verilerin dosyaya yazılması.

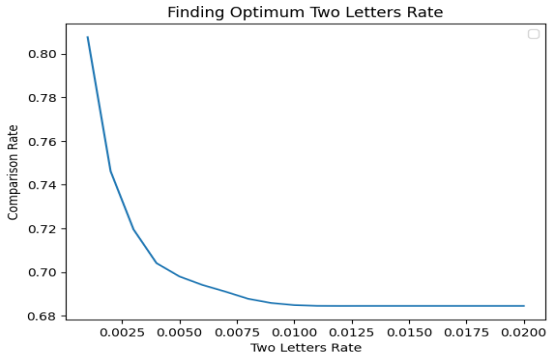
8. **Sıkıştırılmış Verinin Açılması:** Huffman Ağacı, 0 ve 1 ikililerini tutan veri, bu verinin uzunluğu ve açılan dosyanın kaydedileceği uzunluğu alır. Daha sonrasında Huffman Ağacını kullanarak veriyi açar ve verilen dosya ismi ile kaydeder.
9. **Sıkıştırma Oranının Bulunması:** Çıktı olarak elde edilen dosyanın boyutunun dosyanın orijinal boyutuna bölünmesi ile bulunur.
10. **Dosyaların Karşılaştırılması:** Sıkıştırıldıktan sonra tekrar açılan dosya ile orijinal dosyanın içeriklerinin kontrol edilmesi.
11. **Başlıktan Ağacın Oluşturulması:** Başlıkta ön-sıralı (pre-order) olarak tutulmuş olan Huffman Ağacının başlık okunarak tekrar oluşturulması.
12. **Açma Aşaması:** Sıkıştırılan dosyanın okunması, başlık ve veri kısımlarının ayrılması, başlıktan Huffman Ağacının oluşturulması ve verinin açılması aşamalarından oluşur.
13. **Sıkıştırma ve Açma Aşaması:** Verilen dosya adına göre dosya önce sıkıştırma, daha sonra açma aşamasından geçer ve dosya adı, ikili sembol oranı, sıkıştırma oranı, sıkıştırılıp açılan dosyanın aynı olup olmadığının kontrol bilgisi, sıkıştırma süresi ve açma sürelerini döner.
14. **Optimum İkili Sembol Oranının Bulunması:** 0,001 ve 0,2 arasında 0,001 artacak şekilde olan değerler denenmesi ve ortalamadaki sonuçların değerlendirilmesi.

**Uygulama**

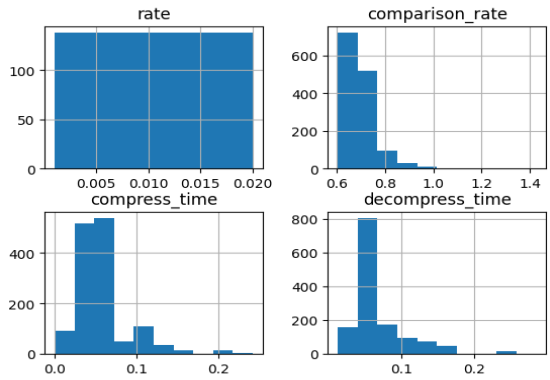
Birleştirilen 69 köşe yazarı için en uzun metin verisi 192 KB, en kısa metin verisi ise 11 KB'tır. 0,001 ve 0,2 arasında ikili sembol oranları kullanılarak sıkıştırma yapıldığında sonuçlar aşağıdaki gibi olmuştur.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| rate | 1380.0 | 0.010500 | 0.005768 | 0.001000 | 0.005750 | 0.010500 | 0.015250 | 0.020000 |
| comparison_rate | 1380.0 | 0.698208 | 0.068007 | 0.605499 | 0.658667 | 0.685927 | 0.711387 | 1.425548 |
| compress_time | 1380.0 | 0.055730 | 0.034149 | 0.000000 | 0.033372 | 0.049673 | 0.065376 | 0.241789 |
| decompress_time | 1380.0 | 0.070586 | 0.037552 | 0.013812 | 0.049838 | 0.063748 | 0.080573 | 0.284028 |

Bu tabloya baktığımızda en iyi sıkıştırma oranının 0,6 olduğu en kötü durumda ise 1,42 olduğu yani bir sıkıştırma işlemi yapılamadığı görülmüştür. Bir diğer göze çarpan şey sıkıştırma sürelerinin açma sürelerinden kısa olduğudur. İkili sembol oranlarına göre grupladığımızda sonuç aşağıdaki gibi olmuştur.




Görsele baktığımızda sembol oranı arttıkça ilk başlarda sıkıştırma oranlarında iyileşme olduğu ama 0,01'den sonra sabit kaldığı görülmektedir. Bunun sebebinin ikili oranın yüksek olmasından dolayı sembolleri tekli olarak kodlanmasıdır. Elde edilen sonuçların histogramı aşağıdaki gibidir.




Sıkıştırma histogramına baktığımızda verinin genel olarak 0,6 ile 0,7 oranlarında sıkıştırıldığı, ortalamada 0,05 sn'de sıkıştırıldığı ve yine 0,06 sn'de açıldığı görülmektedir.

**Sonuç**

69 köşe yazarına ait köşe yazılarının birleştirilmesiyle oluşturulan metinlerin, Huffman Algoritması kullanılarak, değişken ikili sembol oranına göre sıkıştırılması sonucunda, sıkıştırma oranının ikili sembol oranına bağlı olduğu görülmüştür. Oran arttıkça tekli sembol kullanımı artmış ve tekli sembol kullanılarak yapılan sıkıştırılmaların daha başarılı sonuç verdiği görülmüştür. Ortalamada 0,69 oranında bir sıkıştırma elde edilmiştir.