The Turkish Online Journal of Design, Art and Communication - TOJDAC
ISSN: 2146-5193, January 2023 Volume 13 Issue 1, p.193-224

R İLE TWITTER VERİSİ ANALİZİ: VERİ TOPLAMA, SOSYAL AĞ ANALİZİ VE METİN ANALİZİ AŞAMALARI

Naim ÇINAR
Anadolu Üniversitesi, Türkiye
naimcinar@anadolu.edu.tr
https://orcid.org/0000-0002-1824-4076

Atıf	Çınar, N. (2023). R ile Twitter Verisi Analizi: Veri Toplama, Sosyal Ağ Analizi ve Metin Analizi Aşamaları. The Turkish Online Journal of Design Art and Communication, 13 (1), 193-224.

ÖZ

Enformasyon ve iletişim teknolojilerindeki hızlı gelişmeler çevrim içi davranışları anlamak için büyük veri setlerine erişme imkanını da beraberinde getirdi. İnternetin yaygınlaşmasıyla birlikte çok daha fazla sayıda birey, topluluk ve kurum sosyal medya platformlarında dijital sosyal etkileşimler kurmaya başladı. Bu dönüşüm sayesinde, yapılandırılmamış ya da yarı-yapılandırılmış yapıdaki ve çok zengin bir içerik çeşitliliğine sahip olan sosyal büyük veri (*big social data*) her an birikerek artıyor. Dijital sosyal ağların, büyük oranda internet kullanıcıları tarafından oluşturulan içerik yığınını doğal ortamında gözlemleme imkânı sağlaması araştırmacılara çok çeşitli konularda çalışma gerçekleştirmek için ideal bir ortam sağlıyor. Bruns (2020: 65)'un da belirttiği gibi büyük sosyal veri üzerine yapılan çalışmalar aynı zamanda iletişim, kültürel çalışmalar, sosyal bilimler ve bilgisayar bilimi gibi çalışma alanlarının arasında yeni bağlantılar kuruyor. Büyük sosyal veri üzerine yapılan çalışmalarda, içeriğin yapısı, çeşitliliği, erişim imkanları ve karşılıklılık şartı aramayan kullanıcılar arası ilişki yapısı nedeniyle Twitter araştırma yapmak için ideal bir platform olarak ön plana çıkıyor. Bu çalışmada R programlama dili kullanılarak Twitter verisinin toplanması, verinin analize hazır hale getirilmesi, temizlenen veriye otomatik metin analizi ve sosyal ağ analizi yapılması adımlarını örnekler ile açıklayan bir rehber oluşturulması amaçlanmıştır.

Anahtar Kelimeler: *Hesaplamalı Sosyal Bilimler, Hesaplamalı İletişim Araştırmaları, R, Twitter, Sosyal Ağ Analizi, Metin Analizi.*

ANALYSIS OF TWITTER DATA WITH R: DATA COLLECTION, SOCIAL NETWORK ANALYSIS, AND TEXT ANALYSIS STAGES

ABSTRACT

The constant development in information and communication technologies has enabled the opportunity to access and analyze large datasets to understand human behavior in the digital era. Following the widespread use of the internet, social media platforms have become the most popular environments where individuals, communities, and institutions interact. It has led to the emergence of extensive amounts of unstructured or semi-structured big social data that is very rich in content variety. Digital social networks provide an opportunity to observe the online behavior of users in a natural environment which makes it an ideal place for researchers to study a wide variety of topics. Bruns (2020:65) states that big social data approaches connect core disciplines that use big data methods -media, communication and cultural studies, the social sciences, and computer science. Twitter stands out as an ideal platform for research on big social data because of the structure and diversity of the content, data access opportunities, and the structure of the relations between users that does not require reciprocity. This study aims to provide a guideline for data collection from

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

Twitter, data cleaning, social network analysis, and automated text analysis with R programming language.

Keywords: *Computational Social Science, Computational Communication Research, R, Twitter, Social Network Analysis, Text Analysis.*

GİRİŞ

Çevrim içi ortamda kullanıcıların sürekli olarak birbirleriyle etkileşim halinde olması sonucunda kullanıcının ürettiği içerik (*user-generated content*) devasa boyutlara ulaşmış durumdadır. Olshannikova, Olsson, Huhtamäki ve Kärkkäinen (2017: 1)'nin; "insanlarla ilgili veya onların dijital ortamdaki davranışlarını ve teknoloji aracılı sosyal etkileşimlerini tanımlayan, yapılandırılmamış/yarı-yapılandırılmış ancak anlamsal olarak oldukça zengin olan büyük hacimli veri" olarak tanımladığı *sosyal büyük veri*, büyük verinin çok büyük bir bölümünü oluşturmaktadır. Bu verinin devasa boyutu ve anlamsal zenginliği sosyal bilimler çalışmalarına da yön vermektedir. Sosyal büyük veriye erişim imkanları, bu çalışmada örnek uygulamalarla açıklanan hesaplamalı metin analizi ve sosyal ağ analizinin sosyal bilimlerde giderek önemi artan çalışma alanları olmasının temel nedenidir. Özellikle büyük veri, sosyal ağ analizi, sosyal medya içeriği ve konumsal veriyi içeren sosyal bilimler çalışmalarında, hesaplamalı teknikler ve yaklaşımlar kullanılmasının genel adı hesaplamalı sosyal bilimler (*computational social sciences*) olarak ifade edilmektedir (Sagepub, t.y.) ve kısaca, toplum ve insan davranışına hesaplamalı analizler perspektifinden bakan disiplinler arası bir araştırma alanı olarak tanımlanmaktadır (Nature, t.y.).

2000'li yılların ortasından itibaren sosyal medya, insanların neredeyse her konuyla ilgili içerik oluşturma, diğerlerinin oluşturduğu içeriğe erişme, paylaşma, manipüle etme ve değiştirmeye olanak sağlayan bir ortama dönüştü. Instagram, Facebook, TikTok ve Twitter gibi çok çeşitli sosyal medya platformlarının her biri içerik üretimi ve kullanım açısından farklı özelliklere sahip olmakla birlikte, dijital medyanın önemli unsurları olarak toplumsal dönüşümü sürekli olarak etkiliyorlar. Sosyal medya platformlarını birbirinden ayıran en temel özelliklerden birisi içeriğin biçimi olmakla birlikte bir diğeri de o platformdaki kullanıcılar arasındaki ilişkinin yapısıdır. Örneğin Facebook ve LinkedIn gibi sosyal ağlarda her kullanıcının, ağdaki bir diğer kullanıcıyla arasında karşılıklı ilişki (*reciprocated relationship*) mevcuttur. Bu tip yapılara yönsüz ağ (*undirected network*) denir. Buna karşılık Twitter ağındaki kullanıcılar arasında karşılıklı ilişki olması gerekli değildir (Anber, Salah ve El-Aziz, 2016: 241). Hem yönsüz (*undirected*), hem de tek yönlü (*one-way directed)* ve çift yönlü (*two-way directed*) ilişkiler mevcuttur.

Mikroblog formatına sahip ve dünya genelinde en sık kullanılan sosyal ağ platformlarından biri olan Twitter'da yüz milyonlarca kullanıcı, birçok farklı konuda kendini ifade etmektedir. Ortaya çıktığı 2006 yılından günümüze bu platformda üretilen devasa içerik, çok çeşitli çalışma konuları için önemli bir veri kaynağına dönüştü. 280 karakterle sınırlı kısa ve özet içeriklerden oluşan yapısı, zorunlu karşılıklılık olmaksızın diğer kullanıcıları takip edebilme özelliği ve diğer sosyal medya platformlarına kıyasla daha açık uygulama programlama ara yüzüne (API) sahip olması Twitter'i çevrim içi davranışları incelemek için ideal bir platform haline getirmektedir (Grandjean, 2016: 2). Daha da ötesi, Phua, Jin ve Kim (2017) Twitter'in diğer sosyal medya platformlarına kıyasla en fazla köprü kuran sosyal sermayeye (*bridging social capital*) sahip olduğunu belirtmektedir. Köprü kuran sosyal sermaye, bireyler arasında enformasyon paylaşımı için fırsatlar yaratan zayıf ve mesafeli ilişkilere atıfta bulunur. Başka bir deyişle, Twitter'daki enformasyonun sosyal grupların sınırları ötesinde paylaşılma olasılığı yüksektir (Yu, 2020: 2).

Büyük sosyal veri, sosyal bilimler araştırmaları için kritik bir kaynak haline dönüşmüştür. Öte yandan, büyük veriye erişim, verinin kullanışlı hale getirilmek üzere temizlenmesi ve düzenlenmesi, verinin analizi hesaplamalı sosyal bilimler alanıyla ilgili yeterlilikler gerektirmektedir. Bu çalışmada R programlama dili aracılığıyla Twitter verisine erişim ve analizine yönelik uygulamalara yer verilmiştir. Çalışmanın ilk adımında *academictwitteR* paketi aracılığıyla veriye erişim ve verinin

sonraki adımlardaki analizler için hazır hale getirilmesi süreçlerine yer verilmiştir. İkinci adımda, sosyal ağdaki aktörler arasındaki etkileşimleri incelemeye dayalı bir yaklaşım olan sosyal ağ analizi üzerinde durulmuştur. Ağı oluşturan düğümlerin ağ içindeki konumu ve ağdaki enformasyon akışındaki rollerini ortaya koymak için dereceye dayalı merkezilik ölçümleri (derece merkeziliği, özvektör merkeziliği) ve en kısa yola dayalı merkezilik ölçümleri (arasındalık merkeziliği, yakınlık merkeziliği) yapılmıştır. Sosyal ağ görselleştirmesinin R ile nasıl gerçekleştirilebileceği açıklanmış, buna ek olarak ağ görselleştirmesinde sıklıkla kullanılan bir yazılım olan Gephi'nin kullanımından da bahsedilmiştir. Son adımda, tweetlerden oluşan metin yığının içeriğini anlamak üzere metin analizi gerçekleştirilmiştir. Metin yığınındaki kelimelerin seçimi, bağlamları ve aralarındaki ilişkiler ortaya koyularak, iletişim içeriği çözümlenmiştir.

R ile Twitter Verisinin Analizi

Twitter verisinin toplanması, işlenmesi ve analiz edilmesi için çeşitli sosyal medya araştırma araçlarından yararlanılabildiği gibi (bkz: Wasim, A, 2021), işleme ve analiz aşamalarında çok geniş fırsatlar sunması ve açık kaynak olması nedeniyle Python ve R gibi programlama dilleri yoğun şekilde kullanılmaktadır (Steinert-Threlkeld, 2017: 7). Bell Laboratuvarları'nda John Chambers, Rick Becker ve Allan Wilks tarafından geliştirilen S adında bir başka programlama dilini temel olarak alan R, Yeni Zelanda Auckland Üniversitesi'nde çalışan iki akademisyen Ross Ihaka ve Robert Gentleman tarafından 1990'ların başında geliştirilmiştir (Chambers, 2020). Zaman serileri analizi, modelleme, görselleştirme ve diğer veri analizleri gibi yaygın Veri Bilimi görevlerinde en sık kullanılan programlama dillerinden biri olan R, aynı zamanda Twitter API'si ile birlikte de sıklıkla kullanılmaktadır.

Programlama amacıyla kullanılacak Twitter verisine, Twitter'in sağlamış olduğu API'ler (uygulama programlama arayüzleri) aracılığıyla erişilebilmektedir. API özetle iki yazılım bileşeni arasında enformasyon talebi ve akışına olanak sağlayan mekanizmalardır. Stylos ve Myers (2017: 51)'in belirttiği gibi, diğer programcılar belirli programlama görevlerini yerine getirmek için, derlenmiş formattaki mevcut kodlardan oluşan API'leri talep edebilirler. API'ler yeniden kullanıma olanak sağlayarak, yazılım geliştirme sürecinde verimliliği önemli ölçüde artırmaktadır (Qiu, Li ve Leung, 2016: 81). Twitter kendi web sayfasında API'yi bilgisayar programlarının birbirleriyle konuşma biçimi olarak tanımlamaktadır. Bu konuşma, bir yazılım uygulamasının uç nokta (*endpoint*) olarak bilinen şeyi aramasına izin vererek yapılır. Uç nokta, Twitter'in sağladığı belirli bir enformasyon türüne karşılık gelen bir adrestir ve uç noktalar genellikle telefon numaraları gibi benzersizdir. Twitter verisi, kullanıcılarının halka açık olarak paylaşmayı tercih ettiği enformasyonlardan oluşur. Twitter API'si de kullanıcıların paylaşmayı tercih ettikleri herkese açık Twitter verilerine geniş erişim sağlar. Twitter'in mevcut API sürümü, 2021 yılında geliştiricilerin kullanımına açtığı Twitter API v2'dir. Mevcut API'de, *essential, elevated, academic research ve enterprise* olmak üzere farklı erişim seviyeleri mevcuttur (Twitter, t.y.). Enterprise, Twitter'in ticari hedeflere yönelik sunduğu, en ileri düzey erişim imkanları sunan ücretli erişim seviyesi olduğu için bu çalışma içeriğinin dışında tutulmuştur. *Essential* seviyesine erişim için sadece kayıtlı bir Twitter hesabınızın olması yeterlidir. Bu seviyenin, tweet arama için kullanabildiği uç nokta *recent search* (son arama), son bir haftada paylaşılmış kamuya açık tweetlere erişim sağlamaktadır. *Essential* seviyesinde ayda 500bin tweete erişilebilmektedir. Bir üst sıradaki Elevated seviyesine erişim için Twitter geliştirici portalından (*developer portal*) başvuru yapılması gerekmektedir. Benzer şekilde *recent search* uç noktası kullanılabilen seviyede ayda 2milyon tweete erişilebilmektedir. Bu çalışmada paylaşılan örnek veriye erişim ise Academic research (akademik araştırma) erişim seviyesi ile sağlanmıştır. Sadace akademik araştırmalar için kullanılabilen bu erişim seviyesi tüm Twitter API v2 uç noktalarına erişime sahiptir ve aylık tweete erişim sayısı üst limiti (*Twitter caps*) daha yüksektir. Bu seviyenin, tweet arama için kullanabildiği uç nokta *full-archive search* (tüm arşivi arama), Mart 2006 tarihinde atılan ilk tweetten günümüze kadar tüm kamuya açık tweetlere erişim sağlamaktadır. *Academic research* seviyesinde ayda 10 milyon tweete erişilebilmektedir. *Academic research* erişim seviyesine sadece akademik araştırmacılar ve lisansüstü düzeyde tezi üzerine çalışan öğrenciler başvuru yapabilmektedir. Akademik araştırmanın içeriği hakkında detaylı bilgi, Twitter verisinin ne amaçla

Research Article - This article was checked by iThenticate

ve nasıl kullanılacağına yönelik açıklamalar ve bir eğitim kurumunda çalıştığınızı kanıtlayan kişisel bilgilerle başvuru yapılmakta ve değerlendirme sonucunda gerekli kriterler sağlanıyorsa *Academic research* seviyesine erişim onaylanmaktadır.

Twitter Verisini Toplama Aşaması

Bu çalışmada RStudio (versiyon 2022.02.3) IDE (tümleşik geliştirme ortamı) yazılımı aracılığıyla R (versiyon 4.1.3) programlama dili kullanılmıştır. *Academic research* erişim seviyesindeki Twitter API anahtarı kullanılarak, *academictwitteR* paketi (Barrie ve Ho, 2021) ile Twitter verisine erişilmiştir.

Öncelikle, Twitter API'sini kullanmak için *Twitter Developer Portal*'dan ulaşılabilen, *bearer token* adı verilen API'ye erişim anahtarı kullanılmalıdır. set_bearer() fonksiyonu *bearer token*'i ".Renviron" adlı dosyaya kaydetmeye yarar. ".Renviron" API anahtarı gibi hassas bilgilerin saklandığı bir dosyadır.

```
library(academictwitteR)
set_bearer()
```

Açılan ".Renviron" adlı sekmede bulunan TWITTER_BEARER=YOURTOKENHERE metninde, YOURTOKENHERE yerine önceden kopyalanan *bearer* token'in yapıştırılması gerekmektedir. Bu yapıldıktan sonra RStudio yeniden başlatılmalıdır. Artık *academictwitteR* paketinin Twitter API'sine erişimi sağlandığı için veri toplama aşamasına geçmek mümkündür.

Twitter API'sine sorunsuz bir şekilde erişilebildiğinin kontrolü get_bearer() fonksiyonuyla yapılabilmektedir.

```
get_bearer()
```

Araştırılan konuyla ilgili tweetleri indirmeden önce, belirlenen tarihler arasında atılan o konuyla ilgili Tweetlerin sayısına ulaşmak ve bu veriyi kullanarak zaman serisi grafiği oluşturmak, çalışacağınız veri seti hakkında bir ön izleme imkanı verir. count_all_tweets fonksiyonu ile tweet sayılarına ulaşılabilir.

Bu çalışmada örnek olay olarak İstanbul Sözleşmesi konusu belirlenmiştir. İstanbul Sözleşmesi, kadına yönelik şiddet ve aile içi şiddeti önlemek ve bunlarla mücadeleye odaklanan bir uluslararası bir insan hakları sözleşmesidir (Comeforo ve Görgülü, 2022). Sözleşme 01 Ağustos 2014'te 10 ülkenin onaylaması ile yürürlüğe girmiş ve ilk imzalayıcı ülke Türkiye olmuştur ("İstanbul Sözleşmesi kadınları", 2021). Yine Türkiye, imzaya açıldığı günden beri tartışmaların odağında olan İstanbul Sözleşmesi'nden yapılan aleyhte kampanyaların sonucunda 20 Mart 2021 tarihinde çekilmiştir (Çamurcu, 2021: 64). Bu süreçte yaşanan tartışmalar Twitter platformuna da yoğun bir şekilde yansımıştır. Çalışmada kullanılan örnek veri, tartışmaların yoğunlaştığı dönemi kapsayacak şekilde sözleşmeden çekilme tarihinin bir sene öncesinden günümüze kadar olan tweetleri kapsamaktadır. Aşağıdaki kod parçacığı özetle, 20 Mart 2020 saat 00:00'dan 01 Temmuz 2022 saat 00:00'a kadar olan süre içerisinde, içeriğinde "#istanbulsözleşmesi", "#istanbulsözleşmesiyaşatır" ya da "istanbul sözleşmesi" ifadelerinden en az bir tanesi geçen tweetlerin sayısını saat bazında (*granularity = "hour"*) ortaya koymaktadır. Zaman aralığı gün (*granularity = "day"*), saat ya da dakika (granularity = "minute") olarak belirlenebilir. Örneğin kısa bir zaman aralığında atılan tweetlerin sayısı incelenirken her dakika ya da her saat atılan tweet sayısındaki değişimi görmek araştırmacı için anlamlı olabilir.

```
istanbulsozlesmesi <- count_all_tweets(query = c("#istanbulsözleşmesi",
                                                 "#istanbulsözleşmesiyaşatır",
                                                       "istanbul sözleşmesi"),
                                       start_tweets = "2020-03-20T00:00:00Z",
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
                                        end_tweets = "2022-07-01T00:00:00Z",
                                           bearer_token = get_bearer(),
                                           granularity = "hour",
                                           n = 5000000)
```

istanbulsozlesmesi adını verdiğimiz *dataframe* (veri çerçevesi), 3 sütundan oluşmaktadır. "tweet_count" sütunu her gün atılan tweet sayısını göstermektedir.

```
> colnames(istanbulsozlesmesi)
[1] "end"         "start"       "tweet_count"
```

"tweet_count" sütunundaki sayıları toplayarak toplam tweet sayısına ulaşılabilir. Bu örnekte toplam 2228413 adet tweete ulaşılmıştır.

```
sum(istanbulsozlesmesi$tweet_count, na.rm=TRUE)
```

Veri çerçevesindeki "end" ve "start" sütunları saat aralığının başlangıcını ve bitişini göstermektedir. Ancak Twitter'dan elde ettiğimiz bu veriler "character" (karakter) formatındadır. Sonraki aşamalarda yapılacak analizlerde sorun yaşamamak için karakter (*character*) ya da sayısal (*numeric*) formattaki tarih bilgisi içeren sütunlar, POSIXct adı verilen tarih/saat sınıfına dönüştürülmelidir.

Lubridate paketini kullanarak formatı POSIXct sınıfına dönüştürebilir ve veri Türkçe dilinde paylaşılan tweetlerden oluştuğu için saat dilimini İstanbul'un yerel saat dilimine göre ayarlanabilir.

```
library(lubridate)

istanbulsozlesmesi$end = ymd_hms(istanbulsozlesmesi$end,
                                 tz = "Europe/Istanbul")
```

Zaman serisi grafiği oluşturmak için tarih/saat formatına dönüştürdüğümüz "end" sütunu ve her saat atılan tweet sayısını gösteren "tweet_count" sütunları yeterli olduğundan sadece bu iki sütunu bırakılmıştır. Daha anlaşılır bir grafik için "end" sütunu "tarih", "tweet_count" sütunu "tweet_sayisi" olarak yeniden adlandırılmıştır.

```
istanbulsozlesmesi = subset(istanbulsozlesmesi, select = c(end,
tweet_count))
names(istanbulsozlesmesi) <- c("tarih", "tweet_sayisi")
```

Düzenlenen *istanbulsozlesmesi* adlı veri çerçevesi artık zaman grafiği oluşturmaya hazır durumdadır. Grafiği oluşturmak için *ggplot2* paketi kullanılmıştır.

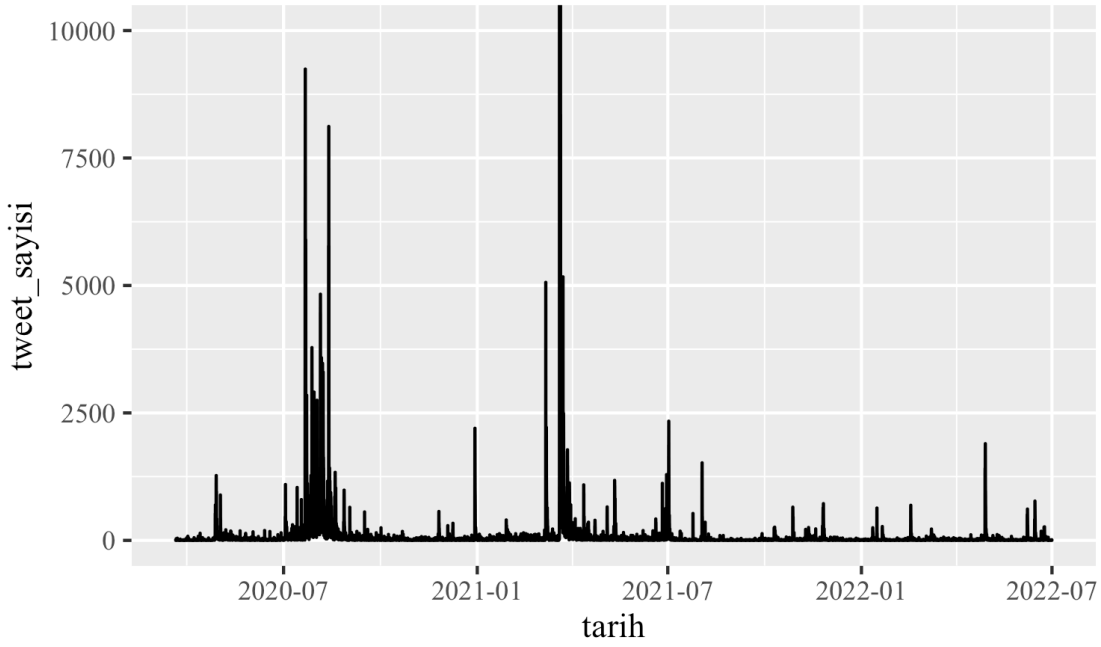
```
library(ggplot2)

gg1 <- ggplot(data = istanbulsozlesmesi,
       aes(x = tarih, y = tweet_sayisi)) +
  geom_line(lwd = 0.5) +
  coord_cartesian(ylim = c(0, 10000)) + #Grafikte görünen saatlik tweet
sayısı üst limitini 10000 olarak belirledik.
  theme(text = element_text(family = "Times New Roman"))
```

Veride ortalamanın çok üzerinde tweet sayısı atılan saat aralıkları olduğu için (örn: 2021-03-20 tarihinde 13:00 ve 14:00 saatleri arasında toplam 76410 tweet), görsel olarak daha anlaşılır bir grafik oluşturmak adına tweet sayılarının olduğu y ekseni için üst limit (10000) belirlenmiştir. Eğer ortalamadan çok farklı değerler yoksa bu satır kullanılmayabilir.

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate



Şekil 1: Saatte atılan tweet sayısını gösteren zaman serisi grafiği

Yukarıdaki grafik iki yıllık süreçte atılan tweet sayılarındaki yoğunlukla ilgili genel bir izlenim vermekle birlikte, hafta, gün ve saat gibi daha dar zaman aralıklarındaki değişimleri görüntülememize olanak sunmamaktadır. Zaman serisi grafiği için bir alternatif yöntem de interaktif bir görselleştirme yapılmasıdır. İnteraktif grafikte zaman dilimleri daha detaylı bir şekilde incelenebilir. Ayrıca interaktif grafikler çevrim içi ortamlarda paylaşım imkanları konusunda da birçok olanak sunmaktadır. Aşağıdaki interaktif grafik *ggplot2 ve plotly* paketleri ile oluşturulmuştur.

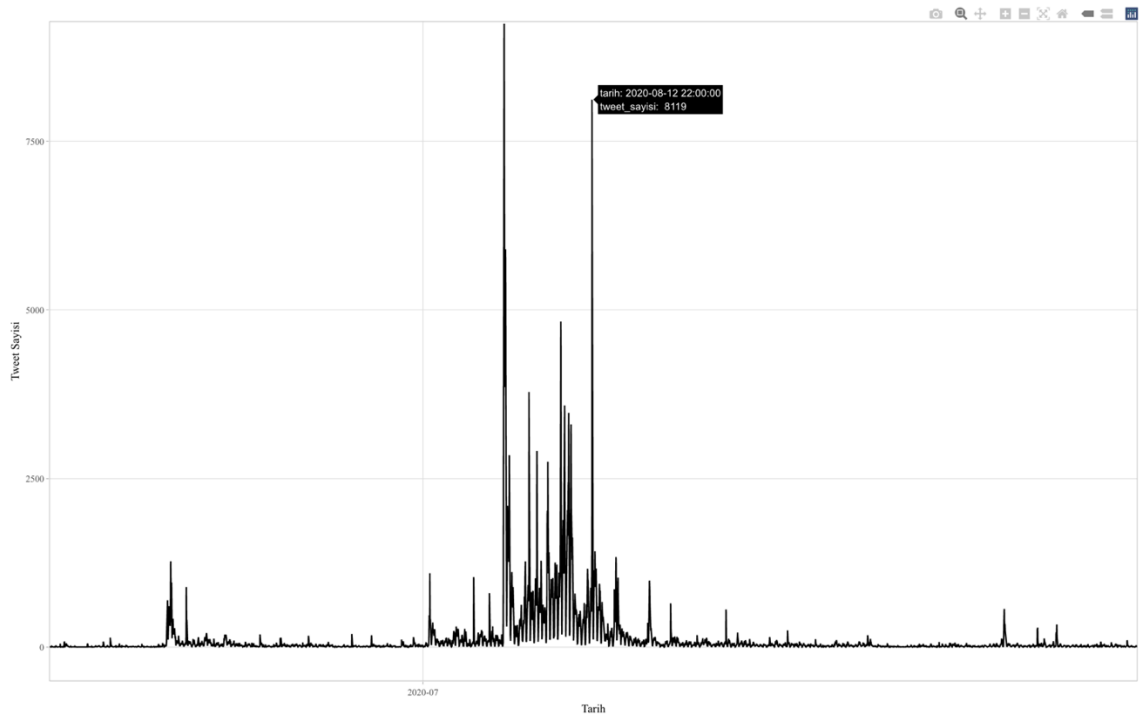
```
library(plotly)

gg2 <- istanbulsozlesmesi %>%
  ggplot(mapping = aes(x = tarih, y = tweet_sayisi)) +
  theme_light() +
  geom_line() +
  xlab(label = 'Tarih') +
  ylab(label = 'Tweet Sayisi') +
  coord_cartesian(ylim = c(0, 10000)) +
  theme(text = element_text(family = "Times New Roman")) +
  ggtitle(label = 'Saatte atılan tweet sayısı')

gg2 %>% ggplotly()
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate



Şekil 2: Plotly ile oluşturulan interaktif zaman serisi grafiğinden detay ekran görüntüsü

Araştırılan konu ile ilgili tweetlerin zaman serisi grafiği ile ön izlemesi yapıldıktan sonra, ilgili tweetlere get_all_tweets fonksiyonuyla erişilebilir. Academic Research erişim seviyesinin aylık tweet erişim üst seviyesi 10 milyon olduğu için, belirlenen zaman aralığındaki ilgili tweetlerin tamamına erişmek mümkündür. Twitter'da İstanbul Sözleşmesi ile ilgili tartışmalar geniş bir zaman dilimi ve çok fazla sayıda tweeti kapsamaktadır. Ancak bu çalışmada temel amaç veri toplama ve analiz süreçlerini açıklamak olduğu için sadece 60245 adet tweetten oluşan örnek bir veri seti üzerinde çalışılmıştır. Zaman serisi grafiğinde de görülebileceği gibi 2020 senesi Temmuz ve Ağustos aylarında konuyla ilgili yoğun bir tartışma yaşanmıştır. *istanbulsozlesmesi* veri çerçevesinin tweet sayısı sütunu (`istanbulsozlesmesi$tweet_sayisi`) ya da interaktif zaman serisi grafiği incelendiğinde bu yoğun tartışmanın 07.21.2020'de başlayıp takip eden bir ay boyunca devam ettiği görülebilir. İstanbul Sözleşmesi'nden çekilmenin tartışıldığı o tarihlerde yeni bir kadın cinayeti haberi ülkenin gündemine oturmuştu. Beş gündür kayıp olan Pınar Gültekin'in 07.21.2020 tarihinde Cemal Metin Avcı adında bir erkek tarafından öldürüldüğü ortaya çıkmıştı. Çalışmada kullanılan 60245 adet tweeti içeren veri, 07.21.2020 ve 07.28.2020 tarihleri arasındaki bir haftalık kısa süreçte yapılan paylaşımları kapsamaktadır.

Belirlenilen parametrelere göre tweet sorgulaması yapılması için build_query fonksiyonu kullanılabilir. Fonksiyondaki mevcut parametreler sayesinde erişilmek istenen tweetler daha net şekilde tanımlanabilir. Böylelikle araştırma amaçları doğrultusunda daha rafine bir veri seti oluşturulur.

```
my_query <- build_query(c("#istanbulsözleşmesi",
"#istanbulsözleşmesiyaşatır", "istanbul sözleşmesi"),
                        remove_promoted = TRUE,
                        lang = "tr")
```

Yukarıdaki kod parçacığında görüldüğü gibi tweet sorgulaması sadece Türkçe tweetlerle sınırlandırılmıştır ve Twitter'da promosyon amacıyla paylaşılan tweetlerin veri setinin dışında tutulmasını tercih edilmiştir. Fonksiyondaki diğer parametler kullanılarak daha da odaklı tercihler yapılabilir. build_query fonksiyonu bu örnekteki gibi ayrı kullanılabileceği gibi get_all_tweets fonksiyonu içinde de kullanılabilir.

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

İndirilen veri setini bilgisayarda saklamak, hızlıca geri getirmek (*restoring*) ve sonraki aşamalarda görüleceği gibi farklı formatlara kolayca dönüştürmek için, RDS ve JSON formatlarında kaydetmek güvenli seçenektir. *file* parametresi veriyi RDS formatında çalışma klasörünün içine kaydedecektir. Eğer *data_path* parametresiyle bir veri yolu (*data path*) tanımlanırsa veri otomatik oluşturulacak bir klasör içinde tweet ve kullanıcı verilerinden oluşan JSON formatlı veri dosyaları olarak da kaydedilir.

```
get_all_tweets(
  query = my_query,
  start_tweets = "2020-07-21T00:00:00Z",
  end_tweets = "2020-07-28T00:00:00Z",
  file = "istanbulsozlesmesi",
  data_path = "tweet-data",
  n = 60000,
)
```

academictwitteR paketi kaydedilen JSON dosyalarını üç farklı veri çerçevesi formatına ("vanilla", "tidy" ve "raw") dönüştürme imkanı vermektedir. Bunlardan birincisi "vanilla" formatıdır. Bu formatta, metin içeren sütunlar düzgün şekilde görüntülenir ancak bazı veriler liste sütunlarında iç içedir ve düzenlenmesi gerekir. Twitter kullanıcı bilgilerini de dışarı çıkarmak için aşağıdaki kod parçacığına `user = TRUE` eklenmelidir. Bu örnekte "vanilla" ve "tidy" formatları birleştirilip çalışma amaçlarına uygun bir veri çerçevesi oluşturulacağı için bu parametre kullanılmamıştır. Formatla ilgili son özellikle ise dışarı çıkarılan veri çerçevesinin *tibble* formatında olmamasıdır. Bu nedenle veri çerçevesi tibble formatına çevrilmiştir. Tibble formatına çevirmek için *tidyverse* paket koleksiyonunda mevcut olan *dplyr* paketi kullanılmıştır. *Tidyverse* veri analizinde sıklıkla kullanılan *ggplot2, dplyr, tidyr, readr, purrr, tibble, stringr* ve *forcats* paketlerini içeren bir paket koleksiyondur ve içeriğindeki paketler bu çalışmanın sonraki aşamalarında sıklıkla kullanılmıştır.

```
library(tidyverse)
vanilla.output <- bind_tweets(data_path = "tweet-data") %>% as_tibble
```

"Vanilla" formatını oluşturan sütunlar şunlardır:

```
> colnames(vanilla.format)
 [1] "public_metrics"      "possibly_sensitive"  "id"                  "lang"
 [5] "source"              "entities"            "author_id"           "text"
 [9] "created_at"          "conversation_id"     "referenced_tweets"   "geo"
[13] "attachments"         "withheld"            "in_reply_to_user_id"
```

İkincisi, "tidy" formatıdır. Paketi geliştiren Barrie ve Ho (2021) bu formatın çoğu sosyal medya araştırması için gerekli tüm sütunları içerdiğini belirtmektedir. "Tidy" varsayılan olarak *tibble* formatındadır.

```
tidy.output <- bind_tweets(data_path = "tweet-data", user = TRUE,
output_format = "tidy")
```

"Tidy" formatını oluşturan sütunlar şunlardır:

```
> colnames(tidy.output)
 [1] "tweet_id"              "user_username"         "text"
"possibly_sensitive"
 [5] "lang"                  "source"                "author_id"
"created_at"
 [9] "conversation_id"       "in_reply_to_user_id"   "user_description"
"user_name"
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
[13] "user_protected"         "user_verified"          "user_created_at"
"user_location"
[17] "user_profile_image_url" "user_url"               "user_pinned_tweet_id"
"retweet_count"
[21] "like_count"             "quote_count"            "user_tweet_count"
"user_list_count"
[25] "user_followers_count"   "user_following_count"   "sourcetweet_type"
"sourcetweet_id"
[29] "sourcetweet_text"       "sourcetweet_lang"       "sourcetweet_author_id"
```

Üçüncüsü ise, "raw" formatıdır. Bu format Twitter API'si ile erişilen tüm veriyi kapsayan veri çerçevesi listelerinden oluşmaktadır. Çalışmada bu format kullanılmadığı için detaylarına yer verilmemiştir.

Veriyi Temizleme Aşaması

Araştırma için gerekli olan veri toplandıktan sonra, analiz için hazır hale getirilmelidir. Bu çalışmada gerçekleştirilen analizler için gerekli veriyi bir araya getirmek için "vanilla" ve "tidy" formatındaki veri çerçeveleri birleştirilerek yeni bir veri çerçevesi ("joined") oluşturulmuştur.

```
addconsecutivenumbers1 <- cbind(vanilla.format,
"observation"=1:nrow(vanilla.format))
addconsecutivenumbers2 <- cbind(tidy.output,
"observation"=1:nrow(tidy.output))

joined <- merge(addconsecutivenumbers1, addconsecutivenumbers2,
by="observation")
```

Daha önce de belirtildiği gibi bazı liste halindeki sütunlarda veriler iç içe olduğu için, buradaki verilerin dışarı aktarılarak düzenlenmesi gerekmektedir. Öncelikle `joined$entities` sütunundaki iç içe verileri dışarı aktarılmıştır. Bu veriler sosyal ağ analizi ve metin analizi aşamalarında kullanılmıştır.

```
joined_entities <- as.data.frame(joined$entities)#entities sütununun
dışarı aktarımı

joined_entities.hashtag <- joined_entities %>%
  select(hashtags) %>%
  hoist(hashtags, hashtag = 3)#tweet içinde var olan hashtaglarin dışarı
aktarımı.

joined_entities.url <- joined_entities %>%
  select(urls) %>%
  hoist(urls, url= 3)#tweet içinde var olan internet linklerinin dışarı
aktarımı.

joined_entities.mentions <- joined_entities %>%
  select(mentions) %>%
  hoist(mentions, mention_username= 3)#mention yapılmışsa, mention
yapılan kullanıcı adı/adlarının dışarı aktarımı.

joined_entities.mention_ids <- joined_entities %>%
  select(mentions) %>%
  hoist(mentions, mention_id= 4) #mention yapılmışsa, mention yapılan
kullanıcı/kullanıcıların id'lerinin dışarı aktarımı.
```

Ardından dışarı aktarılan sütunlar bir araya getirilip yeni bir veri çerçevesi oluşturulmuş ve daha sonra bu veri çerçevesi, orijinal veri çerçevesi ("joined") ile birleştirilmiştir.

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
df.extracted <- data.frame(joined_entities.hashtag, joined_entities.url,
joined_entities.mentions,joined_entities.mention_ids)
df.extracted = subset(df.extracted, select = c(hashtag, url,
mention_username, mention_id))

df.extracted = mutate_all(df.extracted, list(~na_if(.,"NULL"))) #NULL
yazılı olan tüm boş değerleri NA olarak dönüştürülmüştür..

addconsecutivenumbers3 <- cbind(df.extracted,
"observation"=1:nrow(df.extracted))
joined <- merge(joined, addconsecutivenumbers3, by="observation")
#oluşturulan veri çerçevesi, orijinal veri çerçevesi ile
birleştirilmiştir.
```

Artık birleştirilen veri çerçevesinin içinden ihtiyaç olan sütunlar seçilip, diğerleri çıkartılabilir. Bu aşamada araştırma amaçları doğrultusunda ihtiyaç duyulan sütunlar bırakılmış ve ayrıca birleştirme sürecinde oluşan birden fazla tekrarlanan (*duplicate*) sütunlar temizlenmiştir.

```
joined.clean = subset(joined, select = c(id, created_at.x, retweet_count,
like_count, quote_count, url, hashtag, mention_username, mention_id,
sourcetweet_type, sourcetweet_id, sourcetweet_text,
sourcetweet_author_id, text.x, possibly_sensitive.x, author_id.x,
user_username, user_name, user_description,user_profile_image_url,
user_url, user_verified, user_location, user_followers_count,
user_following_count, user_tweet_count, user_list_count, source.x,
in_reply_to_user_id.x))

joined.clean <- joined.clean %>% rename_with(~str_remove(., "\\.x$")) %>%
as_tibble() # Bazı sütun isimlerinin sonuna birleştirme sonucu eklenen .x
uzantıları silinmiştir.
```

Twitter platformunda farklı iletişim kurma biçimleri mevcuttur. Bunlar doğrudan tweet atma, tweete yanıt verme (*reply to*), başka kullanıcıya ait bir tweeti kullanıcının kendi duvarında paylaşması (*retweet*), başka kullanıcıya ait bir tweeti kullanıcının kendi yorumunu da ekleyerek duvarında paylaşması (*quote*) olarak sınıflandırılabilir. Veri setindeki her bir tweetin hangi sınıfa ait olduğuyla ilgili veri sonraki analizler için gereklidir. Öncelikle hangi sınıfta kaç adet tweet olduğu görüntülenebilir.

```
joined.clean %>%
  count(sourcetweet_type, name = "frequency")
```

```
# A tibble: 3 × 2
  sourcetweet_type frequency
  <chr>                <int>
1 quoted                1153
2 retweeted            45453
3 NA                   13639
```

Görüldüğü gibi 60245 adet tweetten oluşan veri seti, 1153 adet quoted tweet, 45453 adet retweet içermektedir. 13639 adet tweetin kaynağı ile ilgili ise bilgi mevcut değil (*NA*) görünmektedir. Tweet türü (*sourcetweet_type*) NA görünen bu tweetler doğrudan tweet ve yanıt tweetleri (*reply to*) kapsamaktadır. Bu sayı içerisinde ne kadarının yanıt tweet olduğu sonraki aşamalarda netleştirilmiştir. Aşağıdaki kod parçacığında her bir sınıf için ayrı veri çerçeveleri oluşturulmuştur.

```
retweeted <- filter(joined.clean, sourcetweet_type == "retweeted")
quoted <- filter(joined.clean, sourcetweet_type == "quoted")
uniques <- joined.clean %>% filter(is.na(sourcetweet_type))
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

Sosyal ağ analizi için retweet, quote ya da reply to yapılan tweetin sahibinin kullanıcı adı bilgisine sahip olmak gerekir. *academictwitteR* ile elde edilen veri setinde retweet ve quote yapılan tweetin sahibinin kullanıcı numarası *sourcetweet_author_id* sütununda, yanıt verilen (*reply to*) tweetin sahibinin kullanıcı numarası ise *in_reply_to_user_id* sütununda bulunmaktadır. Bu sütunlardaki kimlik numaralarından yararlanarak kullanıcıların adları saptanabilir. Bu da R aracılığıyla Twitter verisi toplamak için kullanılan bir diğer paket *rtweet*'de bulunan lookup_users fonksiyonuyla gerçekleştirilebilir. Öncelikle distinct fonksiyonu ile *retweeted* veri çerçevesinin *sourcetweet_author_id* sütununda yer alan benzersiz kullanıcı kimlik numaralarını belirlenmiş, daha sonra lookup_users fonksiyonuyla bu kullanıcı numaralarının kullanıcı adları saptanmıştır. Kullanıcı adları *sourcetweet_usernames_retweeted* veri çerçevesinin "screen_name" sütununda görülebilir.

```
unique_sta_ids_retweeted <- distinct(retweeted, sourcetweet_author_id,
.keep_all=TRUE)

library(rtweet)

sourcetweet_usernames_retweeted <-
lookup_users(as_userid(unique_sta_ids_retweeted$sourcetweet_author_id))
```

Ardından bu veri kullanılarak, r*etweeted* veri çerçevesinin içinde *screen_name* adında, retweet yapılan tweetin sahibi kullanıcı adlarını içeren yeni bir sütun oluşturulmuştur.

```
retweeted <- retweeted %>%
  left_join(
    select(sourcetweet_usernames_retweeted, sourcetweet_author_id =
user_id, screen_name)
  )
```

Bu işlem sonunda *screen_name* sütununda değeri NA olan az sayıda satır kalmış olabilir. Aşağıda görüldüğü gibi bu örnekte 7 satırda *screen_name*'de kullanıcı adı yoktur ve NA olarak görülmektedir.

```
retweeted %>%
  count(is.na(sourcetweet_author_id), name = "frequency")
#is.na FALSE: 454453

retweeted %>%
  count(is.na(screen_name), name = "frequency")
#is.na FALSE: 45446, TRUE= 7
```

Son olarak temiz bir veri çerçevesi oluşturmak adına, *screen_name* değeri NA olan 7 satır çıkarılmıştır. Artık temizlenen *retweeted* veri çerçevesi sonraki analizler için hazır durumdadır.

```
retweeted <- retweeted %>% filter(!is.na(screen_name))
```

Retweeted veri çerçevesi için gerçekleştirilen bu temizleme aşaması benzer şekilde *quoted* veri çerçevesi için de uygulanmıştır.

```
unique_sta_ids_quoted <- distinct(quoted, sourcetweet_author_id,
.keep_all=TRUE)

sourcetweet_usernames_quoted <-
lookup_users(as_userid(unique_sta_ids_quoted$sourcetweet_author_id))

quoted <- quoted %>%
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
  left_join(
    select(sourcetweet_usernames_quoted, sourcetweet_author_id = user_id,
screen_name)
  )

quoted %>%
  count(is.na(sourcetweet_author_id), name = "frequency")
#is.na FALSE = 1153

quoted %>%
  count(is.na(screen_name), name = "frequency")
#is.na FALSE = 1153
```

uniques veri çerçevesi doğrudan tweet ve yanıt tweetleri (*reply to*) kapsamaktadır. Üstteki veri çerçeveleri için takip edilen adımlara benzer şekilde, yanıt olarak yazılan tweetlerde yanıt verilen kullanıcı adlarını getirmek için bu sefer veri çerçevesindeki *in_reply_to_user_id* sütunundan yararlanılmıştır. Bu sütunda değeri NA olmayan satırlar yanıt tweetlerdir (*reply to*).

Tweetin biçimi hakkında bilginin bulunduğu *sourcetweet_type* sütunu "quoted", "retweeted" ve "NA" değerlerinden oluşmaktadır. Ancak yanıt tweetler (*reply to*) de NA olarak görünmektedir. Bu nedenle aşağıdaki kod parçacığının son iki satırında görüldüğü gibi yanıt tweetler için "replyto" değeri atanmıştır. Bu, özellikle *Gephi* yazılımı aracılığıyla sosyal ağ görselleştirilirken kullanışlı olacaktır. Bu örnekte, *uniques* veri çerçevesi *in_reply_to_user_id* sütununda 1483 satırın değerinin NA olmadığı (yani bu sütunda yanıt verilen kullanıcının Twitter kimlik numarasının mevcut olduğu) görülmektedir. Bu da 1483 adet tweetin yanıt tweet (*reply to*) olduğu anlamına gelmektedir. Ancak, yine aynı kod parçacığında görülebileceği gibi yeni oluşturulan *screen_name* sütununda değeri NA olmayan sütun sayısı 1341 adete düşmüştür. Diğer bir deyişle, 142 kullanıcı kimlik numarasından kullanıcı adının tespit edilememiştir. Bunun nedeni süreç içerisinde bazı Twitter kullanıcılarının hesaplarını kapatması ya da Twitter'ın bu hesapları kapatması veya askıya almasıdır. *screen_name* sütununa kullanıcı adı atanamayan kullanıcı kimlik numaraları *www.twitterid.com* ve benzeri Twitter kimlik numarasının sorgulanabildiği web siteleri aracılığıyla kontrol edilirse bu hesapların kapanmış olduğu görülecektir.

```
uniques %>%
  count(is.na(in_reply_to_user_id), name = "frequency") # is NA
TRUE:12156, FALSE: 1483


unique_replyto_ids_uniques <- distinct(uniques, in_reply_to_user_id,
.keep_all=TRUE)


in_reply_to_usernames <-
lookup_users(as_userid(unique_replyto_ids_uniques$in_reply_to_user_id))

uniques <- uniques %>%
  left_join(
    select(in_reply_to_usernames, in_reply_to_user_id = user_id,
screen_name)
  )

uniques %>%
  count(is.na(screen_name), name = "frequency") #is.na TRUE: 12298 FALSE:
1341

uniques <- uniques %>%
  mutate(sourcetweet_type = if_else(!is.na(screen_name), "replyto", ""))
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

Veri artık bundan sonraki aşamalarda gerçekleştirilecek analizler için hazır durumdadır.

Twitter Verisiyle Sosyal Ağ Analizi

Sosyal ağ analizi sosyal aktörler arasındaki etkileşimin incelenmesine dayanan bir yaklaşımdır ve özellikle sosyal medya platformları gibi geniş ve büyük ağlardaki ilişkileri haritalandırmayı ve ilişkisel yapıları ortaya çıkarmayı kolaylaştırır (Anbarlı ve Çınar, 2019: 37).

Öncelikle veri çerçevelerimizde *created_at* sütununda yer alan tarih/saat bilgilerini *lubridate* paketiyle, daha önce de bahsettiğimiz POSIXct sınıfına dönüştürülmüş ve ardından sadece gün bilgisini içeren daha basit bir tarih sütunu (*created_at_round*) daha eklenmiştir.

```
retweeted$created_at = ymd_hms(retweeted$created_at)
quoted$created_at = ymd_hms(quoted$created_at)
uniques$created_at = ymd_hms(uniques$created_at)

library(magrittr)

retweeted %<>%
  mutate(created_at_round = created_at %>% round_date(unit = "day"))
quoted %<>%
  mutate(created_at_round = created_at %>% round_date(unit = "day"))
uniques %<>%
  mutate(created_at_round = created_at %>% round_date(unit = "day"))
```

Sosyal ağ analizi, sosyal aktörler arasındaki bağlantılara odaklanırken çizge teorisinden (*graph theory*) yararlanır. Matematiğin bir dalı olan çizge teorisi, nesneler ve onlar arasındaki ilişkilerin matematiksel temsili olan çizgeleri (*graph*) inceler (Maheswaran, Craigs, Read, Bath ve Willet, 2009: 2). Çizge teorisine göre sosyal ağ, sosyal aktörleri temsil eden düğümler (*node, unit, vertex*) ve onlar arasındaki bir ya da daha çok sayıdaki sosyal ilişkiyi temsil eden ayrıtlardan (*edge, line, tie,link*) oluşur (de Nooy, 2009).

Analiz için sosyal aktörler arasındaki bağlantıyı temsil eden ağ çizgesinin (*network graph*) oluşturulması gerekir. Çizgeyi oluşturmak için kullanılan veri çerçevesi en basit haliyle iki sütundan oluşur. Diğer bir deyişle, her bir satırda iki düğümün adının olduğu bir ayrıt listesidir (*edge list*). Aşağıdaki örnekte *retweeted* veri çerçevesindeki *user_username* ve *screen_name* sütunları kullanılarak bir ayrıt listesi oluşturulmuştur. *user_username* sütunundaki kullanıcı adı retweet yapan kullanıcının adı, *screen_name* sütunundaki kullanıcı adı ise retweet yapılan tweetin sahibi kullanıcının adıdır.

```
retweet_df <- retweeted[, c("user_username","screen_name")]
```

Bu veri çerçevesi ağ çizgesi oluşturmak ve sosyal ağ analizi yapmak için yeterlidir, ancak çoğu zaman ağ çizgeden daha fazlasıdır. Düğümler ve ayrıtlar hakkında çeşitli ek bilgiler içerir. Örnek veri setimiz kullanıcılar ve aralarındaki ilişkiler hakkında çeşitli bilgileri içermektedir. Tweetin hangi cihazdan gönderildiği bilgisini içeren *source* (`retweeted$source`) sütunu, tweetin biçiminin hakkında bilgi içeren *sourcetweet_type* sütunu, ya da tweetin hangi tarihte gönderildiği bilgisini içeren *created_at_round* sütunu bu ek bilgilerden bazılarıdır. Bu bilgiler, sosyal aktörlerin özellikleri, aralarındaki ilişkilerin yapısı hakkında daha derinlemesine analizler yapmayı sağlamaktadır. Dolayısıyla, sosyal ağ analizi için oluşturulan veri çerçevelerine (*retweet_df, quoted_df, replyto_df, mentions_df*) ayrıtların nitelikleriyle (*edge attributes*) ilgili bazı bilgiler eklenmiştir. Eğer veri setinde mevcutsa, araştırmacı amaçları doğrultusunda, düğüm hakkında demografik bilgiler ya da düğümün lokasyon bilgisi gibi daha fazla düğüm niteliği (*node attribute*) ya da ayrıt niteliği (*edge attribute*) ekleyebilir.

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
# İlk iki sütun ayrıt listesini (edge list), diğer ek sütunlar ise ayrıt
niteliklerini (edge attributes) oluşturmaktadır.

retweet_df <- retweeted %>%
  select(user_username, screen_name, device = source, relationship_type =
sourcetweet_type, created_at_round)

quoted_df <- quoted %>%
  select(user_username, screen_name, device = source, relationship_type =
sourcetweet_type, created_at_round)


# replyto_df'nin oluşturma

uniques_df <- uniques %>%
  select(user_username, screen_name, device = source, relationship_type =
sourcetweet_type, created_at_round)

replyto_df <- filter(uniques_df, !is.na(screen_name))

# mentions_df'yi oluşturma

mentions <- rbind(quoted, uniques)
mentions_df <- mentions %>%
  select(user_username, mention_username, device = source,
relationship_type = sourcetweet_type, created_at_round) %>%
  separate_rows(mention_username, sep = " ") %>%
  filter(mention_username != "") %>%
  mutate(relationship_type ="mention")
```

Yukarıdaki kod parçacığında görüldüğü farklı ilişki biçimlerini içeren dört veri çerçevesi (*retweet_df, quoted_df, replyto_df, mentions_df*) oluşturulmuştur. *replyto_df* veri çerçevesi, *uniques_df* veri çerçevesinin içinden sadece reply to tweetlerin filtrelenmesiyle elde edilmiştir. Retweet, quote ve reply to ilişkilerinin yanında son olarak mention ilişkilerini içeren *mentions_df* veri çerçevesi oluşturulmuştur. Mention (bahsetme) özetle, bir tweetin gövde metninin bir ya da daha çok kullanıcının adını içermesidir. Bu veri çerçevesini oluşturmak için *quoted* ve *uniques* veri çerçeveleri birleştirilmiş, *retweeted* veri çerçevesi ise dahil edilmemiştir. Bunun nedeni retweetin yeniden paylaşılan tweetin içeriği dışında ayrıca bir metin içermemesidir. Bir tweet metninde birden fazla kullanıcı adının mention yapılması mümkün olduğu için *mention_username* sütunu kontrol edilirse bazı satırlarda birden fazla kullanıcı adının listelendiği görülecektir. Bu nedenle separate_row fonksiyonu kullanılarak birden fazla olan kullanıcı adları dışarı aktarılmıştır. Son olarak filter fonksiyonu ile *mention_username* sütunu boş olan (NA) satırlar kaldırılarak *mentions_df* oluşturulmuştur.

Ardından, veri çerçevelerinde ayrıt listesini (*edge list*) oluşturan ilk iki sütundaki kullanıcı adları daha iyi bir görselleştirme yapmak adına küçük harfe dönüştürülmüştür.

```
retweet_df <- retweet_df %>%
  mutate(user_username = user_username %>% str_to_lower) %>%
  mutate(screen_name = screen_name %>% str_to_lower)

quoted_df <- quoted_df %>%
  mutate(user_username = user_username %>% str_to_lower) %>%
  mutate(screen_name = screen_name %>% str_to_lower)

replyto_df<- replyto_df %>%
  mutate(user_username = user_username %>% str_to_lower) %>%
  mutate(screen_name = screen_name %>% str_to_lower)
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
mentions_df<- mentions_df %>%
  mutate(user_username = user_username %>% str_to_lower) %>%
  mutate(mention_username = mention_username %>% str_to_lower) %>%
  rename(screen_name = mention_username)
```

Ağ çizgesi yapmak için oluşturulan veri çerçevelerini head ve count fonksiyonlarıyla incelemek mümkündür.

```
head(retweet_df)
count(retweet_df, user_username, screen_name)
```

Ağ analizi ve görselleştirme için kullanılan *igraph* paketinin graph_from_data_frame fonksiyonuyla hazırlanan veri çerçeveleri çizgeye (*graph*) dönüştürülebilir. Bir tweeti retweet/quote yapmak, tweete yanıt vermek ya da tweette başka bir kullanıcıdan bahsetmek (*mention*) yönlü (*directed*) ilişki olduğu için, kod parçacığında ilişkinin yönlü olduğu belirtilmiştir.

```
library(igraph)

retweet_graph <- graph_from_data_frame(d=retweet_df, directed = T)
quoted_graph <- graph_from_data_frame(d=quoted_df, directed = T)
replyto_graph <- graph_from_data_frame(d=replyto_df, directed = T)
mentions_graph <- graph_from_data_frame(d=mentions_df, directed = T)

# Dört veri çerçevesi birleştirilerek, verinin bütünü için de bir çizge
oluşturulmuştur.

whole_df <- rbind(retweet_df, quoted_df, replyto_df, mentions_df)
whole_graph <- graph_from_data_frame(d=whole_df, directed = T)
```

R'nin temel fonksiyonlarından summary ile ya da *igraph* paketinin vcount ve ecount fonksiyonlarıyla, oluşturulan çizgenin kaç adet düğüm (*node*) ve ayrıttan (*edge*) oluştuğu tespit edilebilir.

```
summary(whole_graph) #DN-- 34207 59423 -

#ya da

vcount(whole_graph) #32407
ecount(whole_graph) #59423
```

Düğüm (*node/vertex*) ve ayrıtlardan (*edge*) oluşan bir veri yapısı olan çizge (*graph*) kullanılarak, düğüm düzeyinde ölçümler (*node-level measures*) yapılabilir. Sosyal ağdaki kullanıcılar (düğümler) ağda farklı yapısal pozisyonlar alabilirler ve ağdaki bilgi akışına farklı biçim ve düzeylerde etkileri vardır. Kullanıcıların konumları ve bilgi akışına etki dereceleri çeşitli bağlantı ölçümleriyle saptanır. En bilinen ölçümlerden birisi merkezilik ölçümüdür (*centrality measures*). Merkezilik, bir aktörün ağda ne kadar belirgin bir şekilde bağlı olduğuyla ilgilidir. Ağda daha önemli olan aktörler merkezdedir (Himelboim, 2017: 4). Sosyal ağ analizinde en sık kullanılan merkezilik ölçümleri derece merkeziliği (*degree centrality*), arasındalık merkeziliği (*betweenness centrality*), yakınlık merkeziliği (*closeness centrality*) ve özvektör merkeziliğidir (*eigenvector centrality*). Tablo-1'de görüldüğü gibi bu merkezilik ölçümleri dereceye dayalı merkezilik ölçümleri (*degree-based centrality measures*) ve en kısa yola dayalı merkezilik ölçümleri (*shortest path-based centrality measures*) olarak iki başlık altında ele alınmaktadır.

Tablo 1: Sıklıkla kullanılan merkezilik ölçümleri

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

Dereceye dayalı merkezilik ölçümleri	En kısa yola dayalı merkezilik ölçümleri
Derece merkeziliği	Arasındalık merkeziliği
Özvektör merkeziliği	Yakınlık merkeziliği

Derece merkeziliği, bir düğüme komşu olan düğümlerin sayısının ölçümüdür. Twitter gibi yönlü ağlarda girdi-derecesi merkeziliği (*in-degree centrality*) ve çıktı-derecesi merkeziliği (*out-degree centrality*) olmak üzere iki tür merkezilik çeşidi vardır. Girdi-derecesi diğer kullanıcıların bir kullanıcıyla başlattıkları ilişkilere, çıktı-derecesi ise bir kullanıcının diğer kullanıcılarla başlattığı ilişkilere dayanır. Derece merkeziliği ağdaki tüm komşuların eşit olduğunu varsayar. Önemli olan komşuların sayısıdır. Ancak birçok durumda ağdaki düğüm eğer güçlü düğümlerle bağlantılıysa onun önemi artar. İşte özvektör merkeziliği bu kriter dikkate alınarak hesaplanır. Arasındalık merkeziliği, ağın akışında hangi düğümlerin önemli olduğunu ortaya koyar. Bunu ağdaki en kısa yolları kullanarak belirler. Dolayısıyla, arasındalık merkeziliği her bir düğümün kaç tane en kısa yol üzerinde olduğunu sayar. Düğümün arasındalığı ne kadar yüksekse o ağdaki akış için o kadar önemlidir. Yakınlık merkeziliği de benzer şekilde düğümler arasındaki en kısa yolları dikkate alır ve bir düğümün diğer tüm düğümlere ortalama mesafesini hesaplar (Pelechrinis, 2015).

igraph paketi ile merkezilik dereceleri hesaplanabilir. Aşağıdaki örnekte retweet çizgesi (*retweet_graph*) kullanılarak bu ağdaki düğümlerin merkezlik dereceleri hesaplanmış ve en yüksek puan sahip ilk 20 düğüm sıralanmıştır. Sadece yakınlık merkeziliğinde sıralama küçükten büyüğe doğru olarak seçilmiştir (`decreasing = FALSE`). Bunun nedeni ise bir düğümün yakınlık merkeziliği derecesi ne kadar düşükse o kadar merkeze yakın olmasıdır.

```
in_degree.c <- degree(retweet_graph, mode = c("in"))
in_degree.c_sort <- sort(in_degree.c, decreasing = TRUE)
In_degree.c_sort[1:20]

out_degree.c <- degree(retweet_graph, mode = c("out"))
out_degree.c_sort <- sort(out_degree.c, decreasing = TRUE)
out_degree.c_sort[1:20]

eig.c <- evcent(retweet_graph)$vector
eig.c_sort <- sort(eig.c, decreasing = TRUE)
eig.c_sort[1:20]

closeness.c <- closeness(retweet_graph)
closeness.c_sort <- sort(closeness.c , decreasing = FALSE)
closeness.c_sort[1:20]

betweenness.c <- betweenness(retweet_graph)
betweenness.c_sort <- sort(betweenness.c , decreasing = TRUE)
betweenness.c_sort[1:20]
```

Alternatif bir yol olarak tüm merkezilik dereceleri ölçümlerini içeren bir veri çerçevesi oluşturabilir. Böylelikle, merkezilik derecesi ölçümleri arasında karşılaştırmalar da yapılabilir. Aşağıdaki kod parçacığında tüm veriyi kapsayan çizge (*whole_graph)* kullanılarak bu ağdaki düğümlerin çeşitli merkezilik dereceleri hesaplanmıştır.

```
whole_graph_centralities <- tibble(
  word = V(whole_graph)$name,
  in_degree = degree(graph = whole_graph, mode = c("in")),
  out_degree = degree(graph = whole_graph, mode = c("out")),
  betweenness = betweenness(graph = whole_graph),
  eigen = evcent(whole_graph)$vector)
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

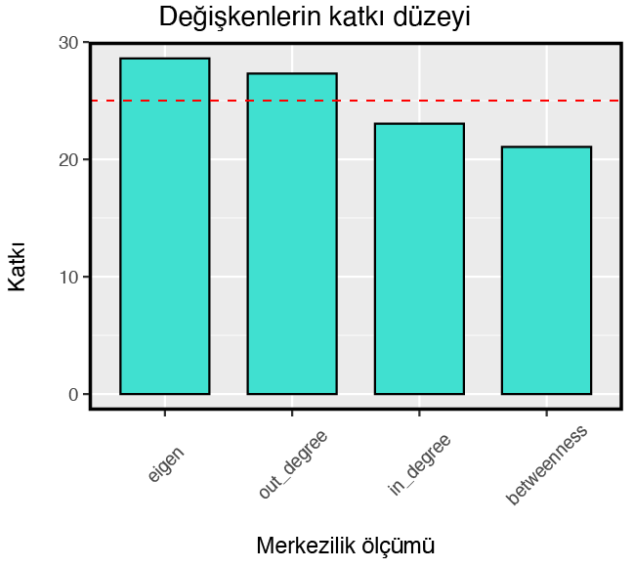
Research Article - This article was checked by iThenticate

Ağ hakkında en bilgilendirici merkezilik ölçüsünü belirlemek için temel bileşenler analizi (*principal component analysis – PCA*) algoritmasından yararlanılır. Temel bileşenler analizi, doğrusal analizde kullanılan bir boyut indirgeme tekniğidir. Bu teknik sayesinde hangi merkezilik derecelerinin, merkezi düğümler hakkında daha fazla bilgi verdiği tespit edilebilir. Böylelikle ağdaki etkili aktörlerin kimler olduğu daha doğru şekilde tanımlanabilir (Ashtiani, Mirzaie ve Jafari, 2019). *CINNA* paketinin pca_centralities fonksiyonu kullanılarak merkezilik dereceleri bilgi verme düzeylerine göre sıralanabilir.

```
library(CINNA)

pca_centralities(
  whole_graph_centralities[2:5],
  scale.unit = TRUE,
  cut.off = 80,
  ncp = 4,
  graph = FALSE,
  axes = c(1, 2)
)
```

Bu kod parçacığının uygulanması sonucunda elde edilen Şekil-3'de görüldüğü gibi analiz edilen sosyal ağdaki merkezi düğümler hakkında en fazla bilgi verici olan özvektör merkeziliğidir.



Şekil 3: Temel Bileşenler Analizi (PCA) algoritmasına göre en bilgi verici merkezilik ölçümü

Merkezilik ölçümleri arasındaki korelasyonları hesaplamak ve görselleştirmek için *CINNA* paketinin visualize_correlations fonksiyonundan yararlanılabilir. Eğer iki ölçüm arasında yüksek korelasyon varsa analizde ölçümlerin ikisini de kullanmaya gerek yoktur çünkü bu farklı ölçümler istatistiksel analizlerde benzer sonuçlar veriyor demektir. Diğer taraftan, iki ölçüm arasında yüksek korelasyon yoksa, farklı sonuçlarla ilişkilendirilebilecek birbirinden ayrı ölçümler olduklarına işaret etmektedir (Valente, Coronges, Lakon ve Costenbader, 2008: 16).

```
visualize_correlations(whole_graph_centralities[,2:5], scale = TRUE,
method = "pearson")
```

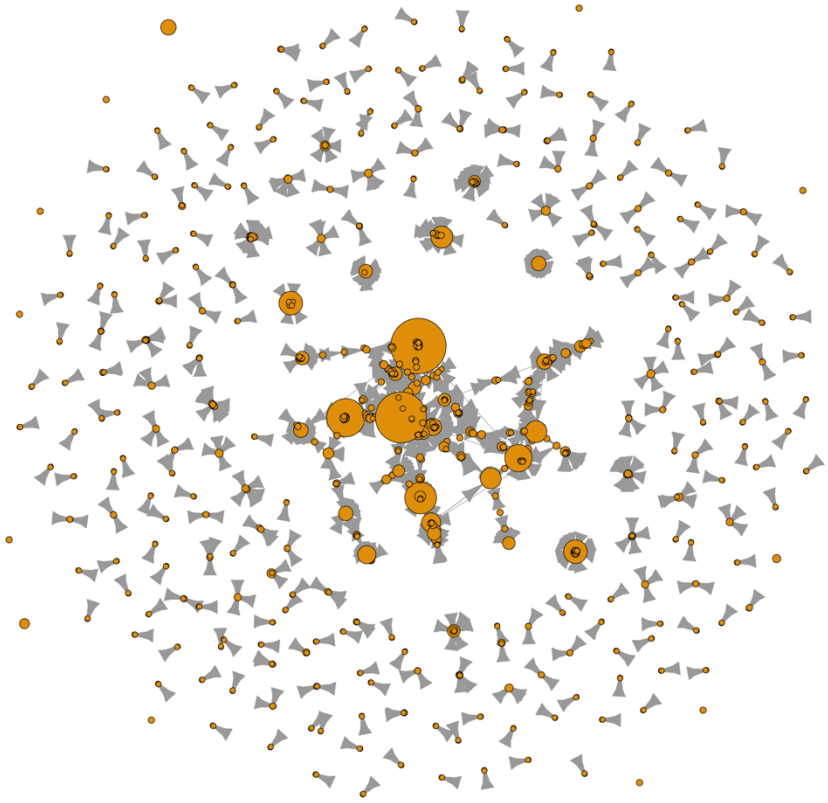
R ile ağ görselleştirmesi yapmak için çeşitli seçenekler mevcuttur. Örneğin, aşağıdaki kod parçacığında *igraph* paketinden yararlanılarak daha önce oluşturulan *quoted_graph* çizgesi görselleştirilmiştir.

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
quoted_graph_layout <- layout.auto(quoted_graph)

plot(
  delete.vertices(simplify(quoted_graph), degree(quoted_graph)==0),
  layout = quoted_graph_layout,
  vertex.size = degree(quoted_graph)/5 +1,
  vertex.label = NA,
  edge.arrow.size = .25
)
```



Şekil 4: *quoted_graph* çizgesi için örnek ağ görselleştirmesi

Bu çalışmada belirlenen konuyla ilgili bir haftalık bir zaman dilimini kapsayan 60245 adet tweetten oluşan örnek bir veri seti üzerinde çalışılmıştır ve oluşturulan ağda 32407 adet düğüm, 59423 adet ayrıt bulunmaktadır. Daha önce count_all_tweets fonksiyonu ile elde edilen bulgular hatırlanacak olursa, 2020 ve 2022 tarihleri arasında İstanbul Sözleşmesi ile ilgili 2 milyonun üzerinde tweet atılmıştır. Uzun bir zaman dilimini kapsayan bu büyük veri setinden elde edilen ağda çok daha fazla sayıda düğüm ve ayrıt olacaktır. Bu büyüklükteki bir ağdan, daha hızlı bir şekilde görsel olarak çekici görselleştirmeler yapmak için açık kaynak bir ağ analizi ve görselleştirme yazılımı olan Gephi'nin kullanılması tercih edilebilir. *igraph* paketinin write_graph fonksiyonu çizgeyi GML (*Graph Modeling Language*) formatına, *rgexf* paketinin igraph.to.gexf fonksiyonu ise çizgeyi GEXF (*Graph Exchange XML Format*) formatına dönüştürmektedir. Gephi bu iki veri formatını da desteklemektedir. Oluşturulan dosya Gephi ile açılarak ağ görselleştirilebilir.

```
#igraph paketi ile GML dosyasının oluşturulması
write_graph(whole_graph, "whole_graph.gml", format = "gml")

#rgexf paketi ile GEXF dosyasının oluşturulması
library(rgexf)
igraph.to.gexf(whole_graph, output = "whole_graph.gexf")
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

Twitter Verisiyle Metin Analizi

Sosyal ağı oluşturan aktörler arasındaki ilişkiler kadar iletişimin içeriği de önemlidir. Twitter görsel, işitsel ve metinsel birçok formatı destekliyor olmakla birlikte, kelimeler Twitter içeriğinin ana yapı taşlarıdır. Bu platformda metin baskın iletişim aracı olduğu için iletişimi oluşturan kelimelerin seçimi, bağlamları ve aralarındaki ilişkileri araştırmak, iletişimin içeriğini çözümlemeye ve gizli anlamları ortaya çıkarmaya yardımcı olmaktadır (Segev, 2021). Bu bölümde veri setindeki tweetlerin oluşturduğu oldukça büyük miktardaki metnin R ile analiz edilmesinin adımları paylaşılmıştır.

Öncelikle veri setinde benzersiz tweetlerin olduğu satırlar filtrelenmelidir. Diğer bir deyişle veri setindeki retweetler orijinal metin içermediği için metin analizinin dışarısında tutulmalıdır. Daha önce oluşturulan *uniques* veri çerçevesi doğrudan tweet ve yanıt tweetleri (*reply to*) kapsamaktadır. Yine benzer şekilde *quoted* veri çerçevesindeki tweetler, quote (alıntı) yapılan tweet yanında kullanıcının kendi yazdığı metni de içermektedir. Bu iki veri çerçevesinin "text" sütunundaki metinler veri çerçevemizdeki toplam benzersiz tweet metinlerini oluşturmaktadır. *tweets.text* veri çerçevesi metin analizi yapılacak tüm tweetleri içermektedir.

```
tweets.text <- rbind(uniques, quoted)
tweets.text = subset(tweets.text, select = c( id, created_at_round, text,
hashtag))
```

Analiz ve görselleştirme aşamasında sorun yaşamamak için, öncelikle "text" sütunundaki metinler temizlenerek analize hazır hale getirilmelidir. Bu çalışmadaki metin temizliği aşamaları sırasıyla; (1) bazı özel karakterler ve Türkçe büyük harflerin, Latin harflerine dönüştürülmesi, (2) büyük harflerin küçük harfe dönüştürülmesi, (3) Türkçe etkisiz kelimelerin (*stopwords*) silinmesi, (4) bazı özel karakterlerin silinmesi ve (5) Türkçe harflerin Latin harflerine dönüştürülmesi olarak özetlenebilir.

İlk olarak özel karakterler ve Türkçe büyük harfleri dönüştürmek için bir liste oluşturulmuş ve veri çerçevesindeki "text" sütununa uygulanmıştır.

```
replacement.list <- list('α' = 'a', 'á' = 'a', 'é' = 'e', 'í' = 'i', 'ó'
= 'o', 'ú' = 'u', 'İ' = 'i', 'Ş' = 's', 'Ü' = 'u', 'Ö' = 'o', 'Ğ' = 'g',
'Ç' = 'c')

tweets.text <- dplyr::mutate(tweets.text, text = chartr(old =
names(replacement.list) %>% str_c(collapse = ''),
new = replacement.list %>% str_c(collapse = ''), x = text))
```

Ardından "text" sütunundaki tüm büyük harfler küçük harfe dönüştürülmüştür.

```
tweets.text$text <- str_to_lower(tweets.text$text)
```

Türkçe etkisiz kelimelerin (stopwords) silinmesi için *stopwords* paketinin kütüphanesinden yararlanılmıştır. Listede küçük harfle yazılmış 471 adet Türkçe etkisiz kelime bulunmaktadır. Araştırma ihtiyaçları doğrultusunda bu listeye yeni kelimeler eklenebilir ya da var olanların bir kısmı silinebilir. Liste oluşturulduktan sonra metin madenciliği (*text mining*) için geliştirilen *tm* paketi kullanılarak "text" sütunundaki metinden derlem (*corpus*) oluşturulmuş ve etkisiz kelimeler kaldırılmıştır. Bunun yanında *tm* paketindeki fonksiyonlar kullanılarak tüm noktalama işaretleri ve rakamlar kaldırılmıştır. Temizleme aşamasından geçirilen metin için veri çerçevesinde "clean.text" adında yeni bir sütun oluşturulmuştur.

```
library(stopwords)

turkish.stopwords <- stopwords::stopwords("tr", source = "stopwords-iso")
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
library(tm)

corpus.text <-  Corpus(x = VectorSource(x = tweets.text$text))

corpus.text <- corpus.text %>%
  tm_map(removeWords, turkish.stopwords) %>%
  tm_map(removePunctuation) %>%
  tm_map(removeNumbers)

tweets.text <- tweets.text %>% mutate(clean.text = corpus.text$content)
```

Daha sonra *tidyverse* içinde mevcut olan *stringr* paketi kullanılarak metinde daha fazla temizlik yapmak için texcleaning adlı bir fonksiyon oluşturulmuş ve veri çerçevesindeki "clean.text" sütununa uygulanmıştır. Bu fonksiyon Stackoverflow.com platformundaki RDRR adlı kullanıcının yazmış olduğu kod parçacığından uyarlanmıştır (Stackoverflow, 2020).

```
#Metni temizlemek için oluşturulan textcleaning fonksiyonu
textcleaning <- function(x) {
  x %>%
    # URL'leri kaldır
    str_remove_all(" ?(f|ht)(tp)(s?)(://)(.*)[.|/](.*)") %>%
    # Evet yerine kullanılan & işaretini boşlukla değiştir
    str_replace_all("&amp;", " ") %>%
    # Noktalama işaretlerini kaldır
    str_remove_all("[[:punct:]]") %>%
    # Mention yapılan kullanıcı adının başındaki @ işaretini kaldır
    str_remove_all("@[[:alnum:]]+") %>%
    # Hashtag (#) işaretini kaldır
    str_remove_all("#[[:alnum:]]+") %>%
    # Herhangi bir yeni satır karakterini boşlukla değiştir
    str_replace_all("\\\n", " ") %>%
    # Metin etrafındaki boşlukları kaldır
    str_trim("both")
}

tweets.text <- tweets.text %>%
  mutate(clean.text = textcleaning(clean.text))
```

Ardından *tidytext* paketindeki unnest_tokens fonksiyonuyla, "clean.text" sütunundaki her satır, kelimelerine bölünerek, her bir kelimenin bir satırda olduğu *words.df* adında yeni bir veri çerçevesi oluşturulmuştur. Son olarak bu veri çerçevesinde yer alan kelimelerdeki Türkçe harfler Latin harflerine dönüştürülmüştür.

```
library(tidytext)

words.df <- tweets.text %>%
  unnest_tokens(input = clean.text, output = word)

# Latin harfe dönüştürülecek Türkçe harfler listesi
replacement.list.turkish <- list('ç' = 'c', 'ğ' = 'g', 'ş' = 's', 'ı' =
'i', 'ü' = 'u', 'ö' = 'o')

words.df <- dplyr::mutate(words.df, word = chartr(old =
names(replacement.list.turkish) %>% str_c(collapse = ''),
new = replacement.list.turkish %>% str_c(collapse = ''), x = word)
```

Artık veri metin analizi için hazır durumdadır. *words.df* veri çerçevesinin "word" sütunundaki (words.df$word) kelimeleri sayarak en sık kullanılan kelimelere hızlıca göz atılabilir. Aşağıda

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

oluşturulan *word.count* veri çerçevesinde kelimeler kullanım sıklıklarına göre sıralanmıştır ve "n" sütununda her bir kelimenin kaç defa kullanıldığı bilgisi bulunmaktadır.

```
word.count <- words.df %>% count(word, sort = TRUE)

word.count %>% head(10)
```

En sık kullanılan 10 kelime ve kaç defa kullanıldığı bilgisi şu şekildedir:

```
> word.count %>% head(10)
# A tibble: 10 × 2
   word                          n
   <chr>                     <int>
 1 istanbulsozlesmesiyasatir  9718
 2 istanbul                   9086
 3 sozlesmesi                 7461
 4 kadinlari                  4563
 5 pinargultekin              4436
 6 uygulansin                 4342
 7 koruyan                    4270
 8 istanbulsozlemesiyasatir   4219
 9 adaletbakanlik             4168
10 tbmmresmi                  4157
```

Sıralanan ilk 10 kelimeye bakıldığında hashtagler ve mention yapılan kullanıcı adlarını da içerdiği görülebilir. Araştırmacı temizlik aşamasında hashtag ve mention yapılan kullanıcı adlarını silmeyi tercih edebilir ancak bu çalışmada metin analizine anlamlı bir katkı sunduğu düşünüldüğü için saklanmıştır. Yine sıralamaya bakıldığında 1. ve 8. satırda aslında aynı hashtag olduğu ancak 8. satırdakinin yanlış yazıldığı görülebilir. *words.df* incelendiğinde buna benzer hatalar tespit edilebilir. Bu durumda silme ya da birleştirme yolu tercih edilerek *word.count* veri çerçevesinde yeniden bir temizlik işlemi gerçekleştirilebilir. Örneğin, 8. sıradaki yanlış yazılmış kelime grubu düzeltilerek 1. sıradaki "istanbulsozlesmesiyasatir" ile birleştirilmiş ve *word.count* veri çerçevesi yeniden oluşturulmuştur. Kod parçacığının altındaki çıktıda iki kelime grubunun birleşmiş olduğu görülmektedir.

```
words.df$word <- words.df$word %>%
  str_replace_all(c("istanbulsozlemesiyasatir" =
"istanbulsozlesmesiyasatir"))

word.count <- words.df %>% count(word, sort = TRUE)

word.count %>% head(1)
```

```
> word.count %>% head(1)
# A tibble: 1 × 2
  word                          n
  <chr>                     <int>
1 istanbulsozlesmesiyasatir 13937
```

ggplot2 ve *plotly* paketleri kullanılarak en sık tekrarlanan kelimeler interaktif bir tablo ile görselleştirilebilir.

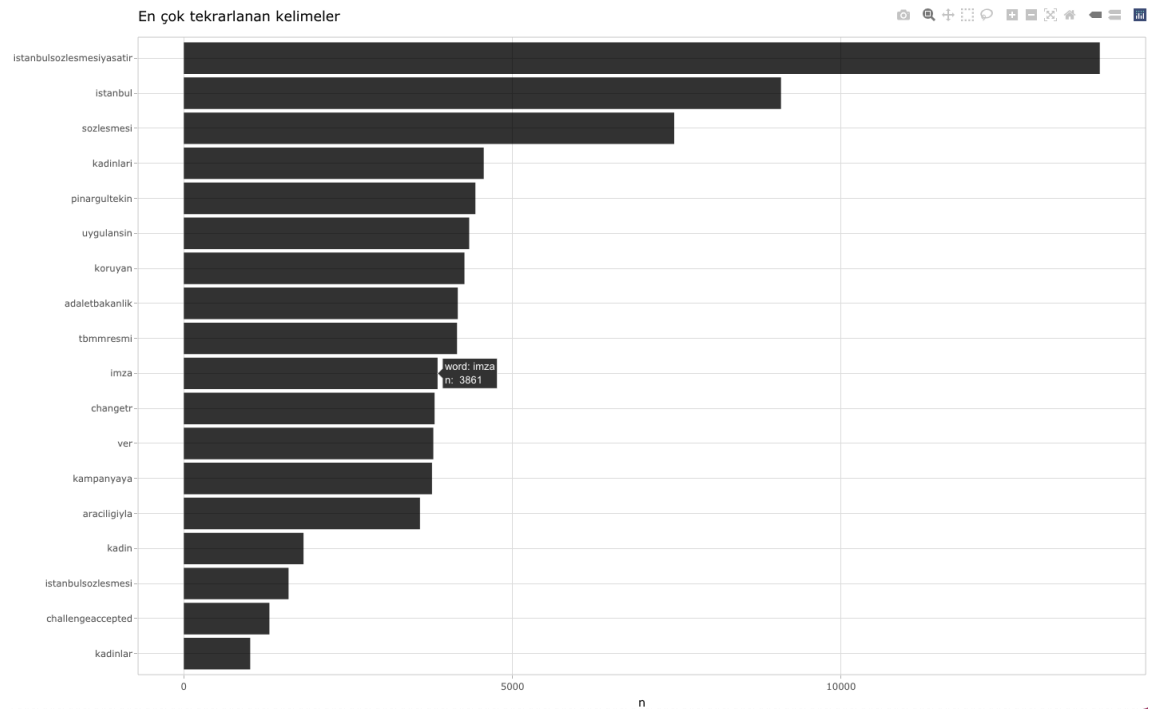
```
plt <- word.count %>%
  # Set count threshold.
  filter(n > 1000) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n)) +
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
  theme_light() +
  geom_col(fill = 'black', alpha = 0.8) +
  xlab(NULL) +
  coord_flip() +
  ggtitle(label = 'En çok tekrarlanan kelimeler')

plt %>% ggplotly()
```



Şekil 5: En çok tekrarlanan kelimeler için oluşturulmuş interaktif tablonun ekran görüntüsü

İnteraktif tablonun yanında, bir kelime bulutu oluşturularak en çok tekrarlanan kelimeler görselleştirilebilir. Bu örnekte kelime bulutu oluşturmak için *wordcloud2* paketi kullanılmıştır.

```
library(wordcloud2)

set.seed(1234)
wordcloud2(data = word.count, size = 4, minSize = 0, shape = 'circle')
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

Şekil 6: En çok tekrarlanan kelimeler için oluşturulmuş kelime bulutu

Tweetlerde emoji adı verilen duygu belirteçleri sıklıkla kullanmaktadır. Emojiler kullanıcıların duygularını dilden bağımsız olarak ifade etmelerine yardımcı olan ideogramlardır. Bu nedenle sosyal medya metinlerindeki duyguların anlaşılmasında emoji ve emoticonların analizi önemli bir katkı sağlamaktadır. *emoji* paketi ile analiz edilen metindeki emojiler tespit edilip, kullanım sıklıklarına göre sıralanabilir. Kod parçacığının altındaki çıktıda en çok tekrarlanan 10 emoji ve ne sıklıkla kullanıldıkları görülmektedir.

```
library(emoji)

emoji.count <- tweets.text %>%
  mutate(emoji = emoji_extract_all(clean.text)) %>%
  unnest(cols = c(emoji)) %>%
  count(emoji, sort = TRUE) %>%
  top_n(30)

emoji.count %>% head(10)
```

```
> emoji.count %>% head(10)
# A tibble: 10 × 2
   emoji     n
   <chr> <int>
 1 🖤      174
 2 💜      142
 3 🙏      101
 4 💪       92
 5 👇       87
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

6	📌	80
7	🌼	65
8	👏	65
9	💪🏻	61
10	🌸	57

Kelimeler Arası İlişkiler

Çalışmanın son bölümünde skip-gram modeli kullanılarak otomatik metin analizi yapılması ve kelimeler arasındaki ilişkilerin ağının görselleştirmesine yer verilmiştir. Fernandez, Gomez ve Martínez-Barco (2014), büyük miktardaki Twitter metnin analizinde n-gram ve skip-gram modellerinin kullanılmasının iyi sonuçlar verdiğini belirtmiştir. N-gram, örnek bir metin içerisindeki bitişik bir *n* sayıda öğe (bu çalışmada öğe ile kastedilen her bir kelimedir) dizisidir. *n* ifadesi metni kaçar kaçar böleceğimizi açıklar. *n* = 1 ise "unigram" , *n* = 2 ise "bigram" (art arda gelen iki kelime), *n* = 3 olduğundaysa "trigram" olarak ifade edilir. *n*-gram modelleri doğal dil öğrenme (*natural language processing -NLP*) uygulamalarında bir sonraki gelecek metni ya da sözü tahmin etme yolu olarak sıklıkla kullanılır (van Gompel ve van den Bosch, 2016). Özetlenecek olursa, *k*-skip-*n*-gram için *n* öğe (kelime) sayısını, *k* ise kaç kez atlama (*skip*) yapılmasına izin verildiğini ifade eder. Dolayısıyla bir *n*-gram aslında hiç atlama olmadığı için 0-skip-*n*-gram ile aynı anlama gelmektedir (Fernandez, Gómez ve Martínez-Barco, 2014: 2). Skip-gram modeli, metin içerisindeki bir kelime için, yine metin içindeki onunla alakalı çevresindeki diğer kelimeleri tespit etmek için kullanılan güdümsüz bir öğrenme tekniğidir. *n*-gram ve skip-gram ile metnin küçük parçalara ayrılması korelasyonların ve kelimelerin etrafındaki bağlamın incelenmesine olanak verir (Woods, 2021). Bu çalışmada skip-gram modeliyle metin küçük parçalara bölünerek (*tokenisation*) kelimeler arası ilişkiler incelenmiştir. Skip-gram modelinin uygulanması ve metinler arası ilişkilerin görselleştirmesi için Orduz (2018)'un R ile veri madenciliği konulu makalesinde paylaşmış olduğu kodlar referans alınmıştır.

tidytext paketindeki `unnest_tokens` fonksiyonu kullanılarak metin skip-gram modeliyle parçalarına ayrılmıştır (*k* = 1, *n* = 2). Ardından, "skipgram" sütunundan (`skip.gram.words$skipgram`) *ngram* paketi kullanılarak iki kelime içeren satırlar belirlenmiş ve filtrelenmiştir. Kelimeler oluşturulan "word-1" ve "word-2" sütunlarına atanarak *skip.gram.words.extracted* veri çerçevesi oluşturulmuştur. Son olarak, *skip.gram.count* veri çerçevesinde skipgram modeline göre kelimeler yan yana gelme sıklıklarına göre sıralanmıştır. "weight" sütununda (`skip.gram.count$weight`) kelimelerin kaç defa yan yana kullanıldıkları bilgisi bulunmaktadır.

```
skip.gram.words <- tweets.text %>%
  unnest_tokens(
    input = clean.text,
    output = skipgram,
    token = 'skip_ngrams',
    n = 2, # metni ikişer ikişer böl
    k = 1 # skip(atlama) sayısı
  ) %>%
  filter(! is.na(skipgram))

library(ngram)

skip.gram.words$num_words <- skip.gram.words$skipgram %>%
  map_int(.f =  ~ ngram::wordcount(.x))

skip.gram.words %<>% filter(num_words == 2) %>% select(- num_words)

skip.gram.words %<>%
  separate(col = skipgram, into = c('word1', 'word2'), sep = ' ') %>%
  filter(! is.na(word1)) %>%
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

```
  filter(! is.na(word2))

skip.gram.words.extracted <- subset(skip.gram.words, select = c(word1,
word2))

skip.gram.count <- skip.gram.words.extracted %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n)

skip.gram.count %>% head(5)
```

```
> skip.gram.count %>% head(5)
# A tibble: 5 × 3
  word1      word2                       weight
  <chr>      <chr>                        <int>
1 istanbul   sözleşmesi                    7303
2 sözleşmesi uygulansın                    4263
3 istanbul   uygulansın                    4262
4 sözleşmesi istanbulsözleşmesiyaşatır     4260
5 uygulansın istanbulsözleşmesiyaşatır     4233
```

skip.gram.count veri çerçevesi incelendiğinde, yine silinmesi ya da değiştirilmesi gereken kelimeler olduğu tespit edilebilir. Bu noktada eğer gerekliyse bir kez daha temizlik işlemi yapılmalıdır. Bu çalışmadaki veri çerçevesinin incelenmesi sonucunda aşağıdaki kelime silme ve değiştirme işlemleri yapılmıştır.

```
#Silinecek kelimeler
wordlist <-
c('the','via','ni','si','nin','la','nden','in','a','been','has','account'
,'witheld','to','la','sı')

skip.gram.words.extracted <- skip.gram.words.extracted %>%
  filter(!word1 %in% wordlist, !word2 %in% wordlist)

#Değiştirilecek kelimeler
skip.gram.words.extracted<- skip.gram.words.extracted %>%
  mutate(word1 = str_replace_all(word1, pattern =
c("istanbulsozlemesiyasatir" = "istanbulsözleşmesiyaşatır",
"istanbulsozlemesiyas" = "istanbulsözleşmesiyaşatır"))) %>%
  mutate(word2 = str_replace_all(word2, pattern =
c("istanbulsozlemesiyasatir" = "istanbulsözleşmesiyaşatır",
"istanbulsozlemesiyas" = "istanbulsözleşmesiyaşatır")))
```

skip.gram.words.extracted veri çerçevesinde kelime temizleme işlemi yapılmışsa, *skip.gram.count* veri çerçevesi yeniden oluşturularak, güncellenmelidir.

```
skip.gram.count <- skip.gram.words.extracted %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n)

skip.gram.count %>% head(5)
```

```
> skip.gram.count %>% head(5)
# A tibble: 5 × 3
  word1                    word2                     weight
  <chr>                    <chr>                      <int>
1 uygulansın               istanbulsözleşmesiyaşatır   8429
2 istanbulsözleşmesiyaşatır pinargultekin              8301
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

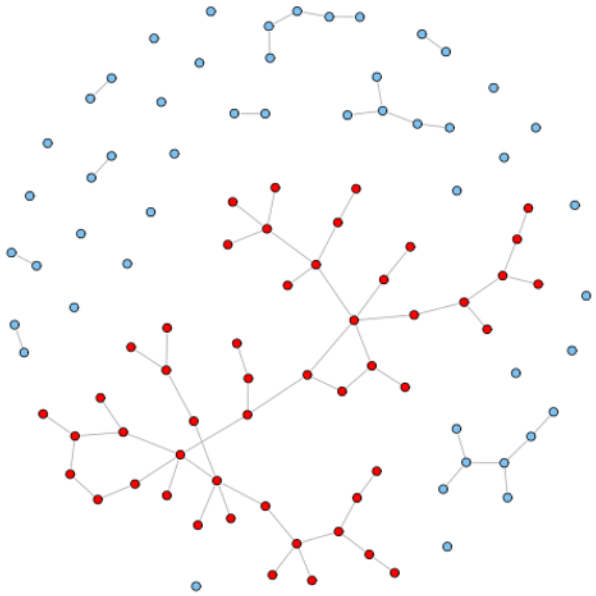
```
3 istanbul                  sözleşmesi                  7303
4 sözleşmesi                istanbulsözleşmesiyaşatır   4264
5 sözleşmesi                uygulansın                  4263
```

skip.gram.count veri çerçevesi, kelimeler arasındaki ilişkilerin ağının görselleştirmesi için hazırdır. Öncelikle, veri çerçevesi çizgeye (*graph*) dönüştürülmüştür. Kelimeler arasındaki tüm bağlantılılar karşılıklı (*reciprocated*) ilişkilerden oluştuğu için ağın yönsüz (*undirected*) olduğu belirtilmiştir. Ağ görselleştirmesinin daha anlaşılır olması için, yan yana gelme sıklıkları (*weight*) 50 ve üzeri olan ilişkiler filtrelenmiştir.

```
threshold <- 50

graph.skipgram <- skip.gram.count %>%
  filter(weight > threshold) %>%
  graph_from_data_frame(directed = FALSE)
```

Yönsüz bir çizge, birbirlerine ayrıtla bağlı düğüm kümeleri olan birden çok altçizgeden (*subgraph*) oluşabilir. Bir diğer deyişle, yönsüz bir çizgede birbirleriyle bağlantılı olmayan bileşenler mevcut olabilir. Aşağıdaki örnekte çeşitli bağlantılı bileşenlerden (*connected component*) oluşan bir yönsüz ağ görünmektedir. Yönsüz ağlarda, genellikle ağın çoğunu oluşturan büyük bir bağlantılı bileşen (*biggest connected component / giant component*) olduğu görülür. Ağın büyük bölümünü bu bağlantılı bileşen oluştururken, geri kalanı genellikle çok sayıda küçük bağlantılı bileşenler oluşturur (Franceschet, 2022). Görseldeki kırmızı renkle işaretli düğümlerden oluşan yapı büyük bağlantılı bileşendir.



Şekil 7: Bir yönsüz ağı oluşturan temel bileşenler
Kaynak: Franceschet (2022).

Bu çalışmadaki ağ görselleştirmesinde birbiriyle bağlantılı olmayan bileşenler dışarıda tutularak, yalnızca en çok düğüme sahip olan büyük bağlantılı bileşen seçilip görselleştirilmiştir.

```
# büyük bağlantılı bileşeni seç
V(graph.skipgram)$cluster <- clusters(graph = graph.skipgram)$membership

network.skipgram <- induced_subgraph(
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
  graph = graph.skipgram,
  vids = which(V(graph.skipgram)$cluster == which.max(clusters(graph =
graph.skipgram)$csize)))
```

Artık, büyük bağlantılı bileşendeki düğümlerin merkezilik dereceleri (derece merkeziliği, yakınlık merkeziliği, arasındalık merkeziliği) hesaplanabilir. Aşağıdaki çıktıda merkezilik derecesi en yüksek 5 kelime sıralanmıştır.

```
node.centralities <- tibble(
  word = V(network.skipgram)$name,
  degree = strength(graph = network.skipgram),
  closeness = closeness(graph = network.skipgram),
  betweenness = betweenness(graph = network.skipgram)
)

# Düğümleri merkezilik derecesine (Her düğümün kaç ayrıtı olduğunu
gösterir) göre sırala

node.centralities %>%
  arrange(- degree) %>%
  head(5)
```

```
# A tibble: 5 × 4
  word                      degree  closeness betweenness
  <chr>                      <dbl>      <dbl>       <dbl>
1 istanbulsözleşmesiyaşatır  37202 0.00000888      2596
2 istanbul                   22942 0.00000885      2358
3 sözleşmesi                 20769 0.00000858        55.5
4 uygulansın                 16954 0.00000212         0
5 pinargultekin              16554 0.00000137         0
```

Ağı oluşturan kelimelerin merkezilik derecelerine göz atıldıktan sonra, ağ görselleştirme aşamasına geçilmiştir. Öncelikle, düğümler ve ayrıtların boyutlarını belirlemek üzere düğümlerin (V) derece merkeziliği (*degree centrality*) ve ayrıtların (E) ağırlık payları (*weight*) hesaplanmıştır.

```
V(network.skipgram)$degree <- strength(graph = network.skipgram)
E(network.skipgram)$width <-
E(network.skipgram)$weight/max(E(network.skipgram)$weight)
```

İnteraktif bir ağ görselleştirmesi elde etmek için, *networkD3* paketi kullanılarak D3 javascript ağ çizgesi oluşturulmuştur.

```
library(networkD3)
networkD3.skipgram <- igraph_to_networkD3(g = network.skipgram)
```

Düğümler için hesaplanan derece merkeziliği ve ayrıtlar için hesaplanan ağırlık payına oranla ağdaki düğüm ve ayrıtların boyutları belirlenmiştir.

```
#Düğüm boyutunu belirle
networkD3.skipgram$nodes %<>% mutate(Degree = (1E-
2)*V(network.skipgram)$degree)

# Ayrıt boyutunu belirle
networkD3.skipgram$links$Width <- 10*E(network.skipgram)$width
```

Görselleştirmedeki bir sonraki aşama ağdaki topluluk yapılarının (*community structure*) belirlenmesidir. Böylelikle ağdaki kelime gruplanmaları (*cluster*) anlaşılır bir şekilde

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

görselleştirilebilir. Modülerlik ölçümü ve hiyerarşik bir yaklaşım üzerine kurulu olan *igraph* paketinin cluster_louvain fonksiyonu kullanılarak düğümler topluluklara atanmıştır.

```
comm.detection <- cluster_louvain(
  graph = network.skipgram,
  weights = E(network.skipgram)$weight)

comm.detection
```

Bu işlem sonucunda ağda 8 adet topluluk oluştuğu görülmektedir.

```
> comm.detection
IGRAPH clustering multi level, groups: 8, mod: 0.57
```

Her bir düğümün hangi topluluğa ait olduğu belirlendikten sonra bu veri düğüm niteliği (*node attribute*) olarak çizgeye eklenmiştir. Ayrıca düğümleri renklendirmek için topluluk aidiyetleri kullanılmıştır.

```
# Düğümün grup aidiyeti, düğüm niteliği olarak eklenmiştir
V(network.skipgram)$membership <- membership(comm.detection)

# Düğümlerin ait olduğu grup etiketi, düğümleri renklendirmede
kullanılmıştır
networkD3.skipgram$nodes$Group <- V(network.skipgram)$membership
```

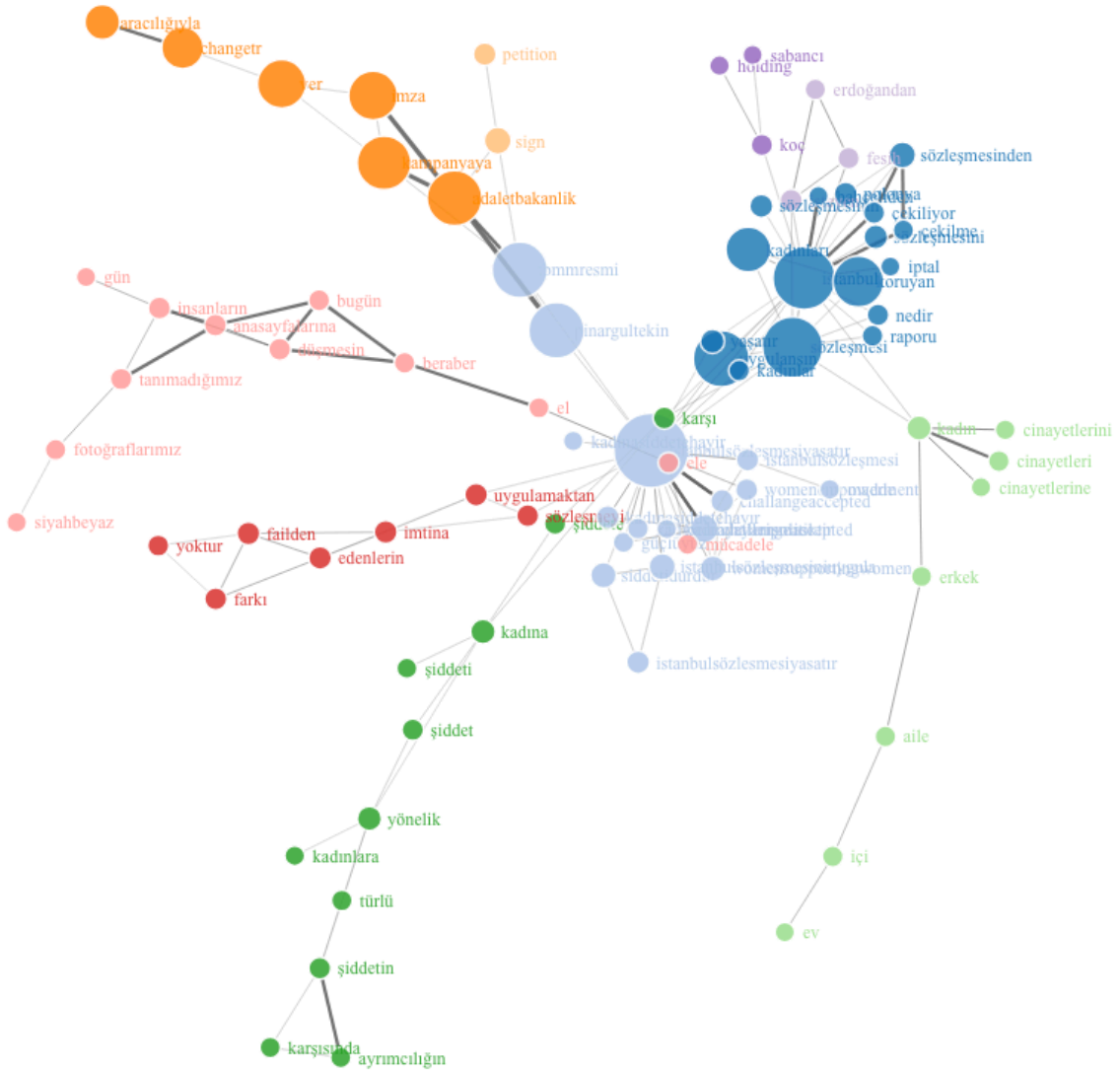
networkD3.skipgram adlı çizge artık interaktif bir ağ görselleştirmesi için hazırdır.

```
forceNetwork(
  Links = networkD3.skipgram$links,
  Nodes = networkD3.skipgram$nodes,
  Source = 'source',
  Target = 'target',
  NodeID = 'name',
  Group = 'Group',
  opacity = 0.9,
  Value = 'Width',
  Nodesize = 'Degree',
  linkWidth = JS("function(d) { return Math.sqrt(d.value); }"),
  fontSize = 12,
  zoom = TRUE,
  opacityNoHover = 1
)
```

Ağ görselleştirmesinde kelimeler arasındaki ilişkiler ve kelimelerin oluşturdukları topluluk yapıları interaktif bir şekilde incelenebilir. Twitter verisi bağlamında düşünülecek olursa, bu analiz Twitter platformunda belirli bir konu üzerine kullanıcıların yaptıkları tartışmalar, kurdukları iletişimden oluşan çok büyük miktardaki metinlerde hangi temaların ön plana çıktığını tespit etmekte oldukça etkilidir. Bunun yanında, otomatik metin analizi ve ağ görselleştirmeleri araştırmacının niteliksel yaklaşımlara başvurarak metindeki örüntüleri ortaya çıkarma sürecine de önemli katkılar sağlayabilir.

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate



Şekil 8: Kelimeler arasındaki ilişkileri gösteren ağ görselleştirmesi

Ağ görselleştirmesine ek olarak, kelime gruplanmalarının oluşturan kelimelerin listesi çıkarılabilir. Aşağıdaki çıktıda her topluluktaki en sık tekrarlanan 5 kelime sıralanmıştır.

```
membership.df <- tibble(
  word = V(network.skipgram) %>% names(),
  cluster = V(network.skipgram)$membership
)

V(network.skipgram)$membership %>%
  unique %>%
  sort %>%
  map_chr(.f = function(cluster.id) {

    membership.df %>%
      filter(cluster == cluster.id) %>%
      slice(1:5) %>%
      pull(word) %>%
      str_c(collapse = ', ')
    })
```

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

```
 [1] "uygulansın, istanbulsözleşmesiyaşatır, pinargultekin,
challengeaccepted, siddetidurdur"
 [2] "istanbul, sözleşmesi, kadınları, koruyan, polonya"
 [3] "tbmmresmi, imza, kampanyaya, adaletbakanlik, changetr"
 [4] "kadına, şiddete, kadınlara, yönelik, şiddetin"
 [5] "kadın, aile, ev, cinayetleri, içi"
 [6] "edenlerin, failden, imtina, sözleşmeyi, uygulamaktan"
 [7] "tanımadığımız, insanların, fotoğraflarımız, anasayfalarına,
düşmesin"
 [8] "koç, holding, sabancı"
```

SONUÇ

İnsanlar sürekli olarak dijital platformlarda birbirileriyle etkileşim kurmakta ve sosyal etkileşimlerine büyük oranda enformasyon teknolojileri aracılık etmektedir. Bunun sonucunda dijital medya ortamında insan tarafından üretilen veri devasa boyutlara ulaşmıştır. Bu büyük veri yığınlarından zengin iç görüler çıkarma fırsatı sosyal bilimler araştırmalarını büyük ölçüde etkilemiştir. Ishikawa (2015)'nın fiziksel dünyadaki veri ve sosyal medya verisi arasındaki ilişkilerin analizi olarak tanımladığı büyük sosyal veri konsepti, blog, mikroblog, sosyal ağ siteleri, video paylaşım platformları, dijital oyunlar, kitle kaynak platformları gibi geniş bir mecra çeşitliliğinde üretilen verilerin tümünü kapsamaktadır (Olshannikova, Olsson, Huhtamäki ve Kärkkäinen, 2017). En bilinen sosyal büyük veri kaynaklarından birisi olan Twitter, ölçeği, uluslararası kapsayıcılığının yüksekliği, kullanıcılarının kamuya açık paylaşımlarında güncel tartışmaların yansımalarının çok hızlı bir şekilde gözlemlenebilmesi ve sunduğu gelişmiş API ile veriye erişimin hızlı, ücretsiz ve kısmen daha kolay olması gibi nedenlerden dolayı akademik çalışmalar için en çok tercih edilen platformlardan birisi olmuştur. R, Twitter verisinin toplanması, temizlenmesi ve analiz edilmesi süreçleri için ideal bir programlama dilidir ve bu süreçleri kolaylaştıran açık kaynak R paketleri mevcuttur.

Bu çalışmada sırasıyla, Twitter verisinin toplanması, analize hazır hale getirilmesi, sosyal ağı oluşturan aktörler arasındaki ilişkilerin analizi ve aktörler tarafından yazılan tweetlerden oluşan metin yığınlarının analizinde R'nin nasıl kullanılabileceği adım adım örneklerle açıklanmıştır. Türkiye gündemindeki tartışmalı konulardan bir tanesi olan İstanbul Sözleşmesi'nin Twitter platformundaki yansımalarını içeren örnek bir veri seti üzerinde çalışma gerçekleştirilmiştir. Akademik çalışmalarda Twitter verisine daha kesin ve eksiksiz olarak ulaşmak için, tüm Twitter API v2 uç noktalarına erişimi olan academic research erişim seviyesinin kullanılması önemli bir avantaj sağlamaktadır. *academictwitteR* sadece academic research seviyesi için sorgulama yapılabilen bir pakettir ve bu çalışmada veri toplama aşamasında kullanılmıştır. Twitter verisi toplamak için kullanılan ve daha bilinen bir paket olan *rtweet* henüz academic research erişim seviyesini desteklememektedir. *rtweet* paketi kullanılarak veri toplaması durumunda, devamındaki veri temizleme ve analiz aşamaları için bu makalede anlatılan aynı adımların izlenmesi mümkündür.

API ile erişilen Twitter verisi sosyal ağı oluşturan düğümler ve ayrıtlar hakkında çok detaylı bilgiler içerdiği için kapsamlı ve derinlemesine bir sosyal ağ analizi gerçekleştirmek mümkündür. Zengin Twitter verisi, oluşturulan ağ çizgesinde düğüm ve ayrıt niteliği olarak atanabilecek çeşitli bilgiler içermektedir ve araştırmanın amaçları doğrultusunda bu nitelikler çizgeye eklenip ağ analiz edilebilir. Örneğin, veri çerçevesindeki *source* sütunu tweetin Android tabanlı telefon, IOS tabanlı telefon, bilgisayar ya da tablet gibi hangi tip bir cihazdan gönderildiği bilgisini içermektedir. Bu bilgi kullanıcıların sosyo-ekonomik yapısı hakkında ipucu verebilir ya da bot hesapların tespitine yardımcı olabilir. Benzer şekilde, *user_verified* sütunu kullanıcının mavi onay rozeti olan tanınmış bir hesap olup olmadığı bilgisini içermektedir. Tanınmış kullanıcıların ağ üzerindeki etkileri incelenmek istenirse bu bilgi ağ çizgesine düğüm niteliği olarak eklenebilir. Özetle, detaylı bir sosyal ağ analizi gerçekleştirmek için gerekli birçok veriye Twitter API aracılığıyla erişmek mümkündür.

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate

Kelimeler Twitter içeriğinin ana yapı taşlarıdır ve bu metin yığınlarından zengin iç görüler çıkarmak mümkündür. Bu çalışmada metin analizi ile büyük metin yığınlarını oluşturan kelimelerin bir ağ yapısı içerisinde bağlamsal olarak nasıl konumlandıklarını ve birbirleriyle ilişkilerin nasıl incelenebileceği gösterilmiştir. Yapılan görselleştirmeler ile metin kaynaklarının görsel olarak nasıl keşfedilebileceği açıklanmıştır. Bununla birlikte Twitter'daki metin içeriklerinin informal yapıda olması, Türkçe'nin yapısal özellikleri ve çalışmaların genellikle İngilizce üzerinde yoğunlaşmış olması metin analizi sürecinde, duygu analizi, ironi tespiti, kelime kökü bulma gibi bazı aşamaları zorlaştırmaktadır.

KAYNAKÇA

Anber, H., Salah, A., ve Abd El-Aziz, A. A. (2016). A literature review on Twitter data analysis. *International Journal of Computer and Electrical Engineering*, 8(3), 241. https://doi.org/10.17706/ijcee.2016.8.3.241-249

Ashtiani, M., Mirzaie, M. ve Jafari, M. (2019). CINNA: an R/CRAN package to decipher central informative nodes in network analysis. *Bioinformatics*, 35(8), 1436-1437. https://doi.org/10.1093/bioinformatics/bty819

Barrie C, Ho J (2021). academictwitteR: an R package to access the Twitter Academic Research Product Track v2 API endpoint. *Journal of Open Source Software*, 6(62), 3272. https://github.com/cjbarrie/academictwitteR.

Bruns, A. (2020). Big social data approaches in Internet studies: The case of Twitter. Second international handbook of Internet research, 65-81. https://doi.org/10.1007/978-94-024-1555-1_3

Comeforo, K. ve Görgülü, B. (2022). Democratic possibilities of digital feminism. *Democratic Frontiers: Algorithms and Society*, (63-82), Routledge Focus. https://doi.org/10.4324/9781003173427-4

Chambers, J. M. (2020). S, R, and data science. *Proceedings of the ACM on Programming Languages,* 4(HOPL), 1-17. https://doi.org/10.1145/3386334

Çamurcu, M. H. (2022). İstanbul Sözleşmesi: Türkiye'de iç hukuka etkisi ve toplumun tepkisi. *Ankara Barosu Dergisi*. 79(4), 63-106. https://doi.org/10.30915/abd.1090725

de Nooy, W. (2009). Social network analysis, graph theoretical approaches to. *Encyclopedia of complexity and system science,* 8231-8245. https://doi.org/10.1007/978-0-387-30440-3_488

Fernández, J., Gómez, J. M. ve Martínez-Barco, P. (2014, October). A supervised approach for sentiment analysis using skipgrams. In *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working* Days (JISIC), 30-36.

Francechet, M. (2022). *Network science.* Universita Degli Studi di Udine. http://users.dimi.uniud.it/~massimo.franceschet/teaching/datascience/network

Grandjean, M. (2016). A social network analysis of Twitter: Mapping the digital humanities community. *Cogent Arts & Humanities*, 3(1), 1171458. https://doi.org/10.1080/23311983.2016.1171458

Hoffman, M. (2021). Methods for network analysis. Stanford University. https://bookdown.org/markhoff/social_network_analysis/

İstanbul sözleşmesi kadınları şiddetten koruyor. (2021, 22 Mart). İstanbul Barosu. https://www.istanbulbarosu.org.tr/HaberDetay.aspx?ID=16265

Maheswaran, R., Craigs, C., Read, S., Bath, P.A. ve Willett, P. (2009). A graph-theory method for pattern identification in geographical epidemiology-a preliminary application to deprivation and mortality. *International Journal of Health Geographics,* 8(1), 1-8. http://www.ij-healthgeographics.com/content/8/1/28

Nature. (t.y.). *Computational social science.* https://www.nature.com/collections/cadaddgige/

Olshannikova, E., Olsson, T., Huhtamäki, J. ve Kärkkäinen, H. (2017). Conceptualizing big social data. *Journal of Big Data*, 4(1), 1-19. https://doi.org/10.1186/s40537-017-0063-x

Orduz, J. C. (2018, 20 Aralık). *Text mining, networks and visualization: Plebiscito tweets.* Juanitorduz. https://juanitorduz.github.io/text-mining-networks-and-visualization-plebiscito-tweets/

Özbaş Anbarlı Z. ve Çınar, N. (2019). Sosyal ağlarda neler oluyor? Sosyal ağ analizi ile pazarlama iletişimi araştırmaları. *İletişim Araştırmalarında Farklı Bakış Açıları*, (37-55), Detay Yayıncılık.

Pelechrinis, K. (2015). TELCOM2125: Network science and analysis. [Powerpoint sunumu]. School of Information Sciences, University of Pittsburgh. https://sites.pitt.edu/~kpele/Materials15/module2.pdf

Phua, J., Jin, S. V., ve Kim, J. J. (2017). Uses and gratifications of social networking sites for bridging and bonding social capital: A comparison of Facebook, Twitter, Instagram, and Snapchat. *Computers in Human Behavior*, 72, 115-122. https://doi.org/10.1016/j.chb.2017.02.041

Qiu, D., Li, B., ve Leung, H. (2016). Understanding the API usage in Java. *Information and Software Technology*, 73, 81-100. https://doi.org/10.1016/j.infsof.2016.01.011

Sagepub. (t.y.). *Methods map: Computational Social Science.* https://methods.sagepub.com/methods-map/computational-social-science

Segev, E. (Ed.). (2021). *Semantic Network Analysis in Social Sciences*. Routledge. https://doi.org/10.4324/9781003120100

Stackoverflow. (2020, Ocak 29). How do I clean Twitter data in R. https://stackoverflow.com/questions/31348453/

Steinert-Threlkeld ve Zachary C. (2018). Twitter as data. Cambridge University Press. https://doi.org/10.1017/9781108529327

Stylos, J., & Myers, B. (2007, September). Mapping the space of API design decisions. In *IEEE Symposium on Visual Languages and Human-Centric Computing* (VL/HCC 2007) (pp. 50-60). IEEE.

Twitter. (t.y.). *Twitter API.* https://developer.twitter.com/en/docs/twitter-api

Valente, T. W., Coronges, K., Lakon, C., ve Costenbader, E. (2008). How correlated are network centrality measures? *Connect(Tor).* 28(1), 16-26. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2875682/

van Gompel, M. ve van den Bosch, A. (2016). Efficient n-gram, skipgram and flexgram modelling with colibri core. *Journal of Open Research Software*, 4(1), e30. https://doi.org/10.5334/jors.105

Wasim, A. (2021, 18 Mayıs). Using Twitter as a data source an overview of social media research tools. *The London School of Economics and Political Science (LSE), Impact of Social Sciences Blog.* http://eprints.lse.ac.uk/111332/

Woods, C. (2021). *Text and sentiment analysis in R*. Chryswoods.com. https://chryswoods.com/text_analysis_r

Yu, J., 2021. Discovering Twitter through computational social science methods. Yayınlanmamış Doktora Tezi. Universitat Autonoma de Barcelona.

Submit Date: 21.11.2022, Acceptance Date: 26.12.2022, DOI NO: 10.7456/11301100/014

Research Article - This article was checked by iThenticate