KOCAELİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLM209 PROGRAMLAMA LAB. 3

Selma Yıldız — *220201091*

Mediha Tuğba Keyik — *220201044*

1.Özet:

Bu proje, Twitter API aracılığıyla kullanıcı Bu proje, Twitter API aracılığıyla kullanıcı verilerini çekerek bu verileri analiz etmeyi ve kullanıcılar arasında benzer ilgi alanlarına göre eşleştirmeler yapmayı hedeflemektedir. Bununla birlikte kullanıcı verilerini, hash tablolarıyla organize ederek ve arama algoritmalarını kullanarak verimli bir şekilde ilgi alanlarına göre eşleştirme yapmayı amaçlamaktadır.kullanıcılar arasında benzer ilgi alanlarına göre eşleştirmeler yapmayı hedeflemektedir. Bununla birlikte kullanıcı verilerini, hash tablolarıyla organize ederek ve arama algoritmalarını kullanarak verimli bir şekilde ilgi alanlarına göre eşleştirme yapmayı amaçlamaktadır.

2.Giriş:

Bu proje Java programlama dili, Zemberek, Json-simple ve Swing kütüphanesi kullanılarak yapılmıştır. Projenin ana metodları şunlardır:

- *Json dosyasından veri çekme,*
- *Girilen kullanıcı adıyla graf oluşturulması,*
- *Kullanıcıların tweet'lerinin analiz edilmesi ve ilgi alanlarının oluşturulması,*
- *Her bölgeye ait trendlerin analiz edilmesi,*
- *Her ilgi alanı için bu ilgi alanına sahip kullanıcıların analiz edilmesi.*

3.Yöntem:

SON Dosyasını Okuma: jsonFilePath değişkeni ile belirtilen twitter_data.json adlı JSON dosyası okunur.

Dosyanın içeriği Files.readAllBytes ve Paths.get metodları kullanılarak bir byte dizisine çevrilir. Bu byte dizisi jsonData adlı bir dize içerisine atanır.

JSON Verisini İşleme: JSON verisini işlemek üzere JSONParser sınıfı kullanılır. jsonArray adlı bir JSONArray nesnesi oluşturularak, JSON verisi bu nesneye parse edilir.

Her bir JSON objesi üzerinde döngü kullanılarak, kullanıcı bilgileri çıkarılır. Bu bilgiler, User sınıfı kullanılarak bir kullanıcı nesnesi oluşturulur.

Oluşturulan kullanıcı nesnesi, CustomHashTable adlı özel bir karma tablosuna eklenir. Kullanıcıdan Giriş Bekleme: Kullanıcıdan bir kullanıcı adı girmesi istenir ve bu bilgi kullaniciGiris değişkenine atanır.

İlişki Grafiği Oluşturma: Girilen kullanıcı adına ait kullanıcının takipçileri ve takip ettikleri belirlenir. Bu bilgiler, UserGraph sınıfı kullanılarak bir grafikte görselleştirilir.

Kullanıcı Bilgilerini Gösterme: Girilen kullanıcı adına ait kullanıcı bilgileri (retrievedUser) bulunursa ekrana yazdırılır.

Kullanıcının adı, tweet'leri gibi bilgiler görüntülenir. Eğer kullanıcı bulunamazsa, "User with username not found" şeklinde bir hata mesajı yazdırılır. Zemberek İşlemleri: ZemberekProcessor sınıfı, dil işleme görevleri için Zemberek kütüphanesini kullanır.

Kullanıcı tweet'leri Zemberek kütüphanesi kullanılarak işlenir. Her Kullanıcının En Çok Kullandığı 10 Kelime ve Ortak Kelimeleri Bulma: CustomHashTable sınıfının ilgili metotları kullanılarak her kullanıcının en çok kullandığı 10 kelime ve ortak kelimeler bulunur. Bu istatistikler ekrana yazdırılır.

Grafiksel Kullanıcı Arayüzü: SwingUtilities kullanılarak grafiksel bir kullanıcı arayüzü oluşturulur. UserGraphVisualization sınıfı, kullanıcıların takipçi-takip edilen ilişkilerini gösteren bir grafiksel arayüz sağlar.

Bu arayüzde, kullanıcıların birbirleriyle olan etkileşimleri görsel olarak takip edilebilir. ZemberekProcessor, Zemberek kütüphanesini kullanarak Türkçe metinleri işlemek için tasarladık. zemberek.morphology.TurkishMorphology ve zemberek.tokenization.TurkishTokenizer sınıfları kullanılıyor. CustomHashTable sınıfına bağlı bir nesne (kelimeFrekansTable) kullanılıyor. IOException durumları ele alınıyor. Zemberek morfolojik analizi ve tokenizasyon için öntanımlı ayarlar kullanılıyor.

Constructor (Yapıcı Metod): kelimeFrekansTable parametresi alarak bir ZemberekProcessor nesnesi oluşturuluyor. Zemberek tokenizer ve morfoloji nesneleri başlatılıyor. processUserTweets Metodu: Bir User nesnesi alıp, kullanıcının tweetlerini işleyen bir metot. Kullanıcının her tweet'i için Zemberek ile ayrıştırma yapılıyor. Ayrıştırılan kelimeler, kelime frekans tablosuna ekleniyor.

engellenenKelimeler Listesi: Zemberek ile ayrıştırılırken filtrelenen kelimeler listesi. Örneğin: bağlaçlar, zamirler, sayılar, sıklıkla kullanılan kelimeler.

zemberekIleAyrıştır Metodu: Zemberek ile tokenizasyon yaparak metni kelimelere ayırıyor. Her kelimenin kök formunu alarak engellenen kelimeler listesinde olup olmadığını kontrol ediyor. Engellenen kelimeler dışındaki kök kelimeleri bir liste olarak döndürüyor.

getRootForm Metodu: Verilen kelimenin morfolojik analizini Zemberek kullanarak alıyor. Analiz sonuçlarından kök kelime formunu buluyor. Eğer analiz sonucu yoksa, kelimenin kendisini kullanıyor. Kodun temel amacı, kullanıcının tweetlerindeki kelimelerin kök formlarını bulmak ve bu kelimelerin frekanslarını bir tabloda tutmaktır. Ayrıca, belirli kelimelerin analiz sırasında filtrelenmesi sağlanmaktadır.

CustomHashTable, özel bir hash tablosu oluşturan ve kullanıcıların tweetlerini işleyen işlevselliği sağlayan bir sınıf. Bu sınıfta MyLinkedList, MyList, User, ZemberekProcessor sınıfları kullanılıyor. UserGraph sınıfına bağlı bir nesne (userGraph) kullanılıyor. Constructor (Yapıcı Metod): Hash tablosu ve ilgili veri yapıları başlatılıyor. UserGraph nesnesi oluşturuluyor.

Hash Fonksiyonları: hash ve hashForKelimeFrekans metotları, kullanıcı adları ve kelimeler için özel hash fonksiyonları sağlıyor. Kullanıcı İşlemleri: addUser metodu, kullanıcı eklemek için kullanılıyor. getUser metodu, kullanıcıyı kullanıcı adına göre getirmek için kullanılıyor. processUserTweetsWithZemberek metodu, ZemberekProcessor kullanarak kullanıcı tweetlerini işlemek için kullanılıyor.

Kelime Frekansları ve İlgili İşlemler: addKelime metodu, kullanıcının belirli bir kelimeyi kullanım frekansını güncelliyor. findKelime metodu, belirli bir kelimenin frekanslar listesinde olup olmadığını kontrol ediyor. getKullaniciKelimeFrekansListesi metodu, belirli bir kullanıcının kelime frekans listesini getiriyor. herKullanıcınınEnCokKullandigi10KelimeyiBul metodu, her kullanıcının en çok kullandığı 10 kelimeyi buluyor ve rapor oluşturuyor.

Ortak Kelimeler ve İlgili İşlemler: CommonWordUsage iç içe sınıfı, ortak kelimeleri ve bu kelimeleri kullanan kullanıcıları takip eden sınıfı temsil eder. ortakKelimeleriBul metodu, tüm kullanıcıların en çok kullandığı ortak kelimeleri bulur ve bu kelimeleri kullanan kullanıcıları takip edenleri bulur.

Dil ve Bölge Trendleri: dilbolgetrend iç içe sınıfı, belirli bir dil bölgesinin trendlerini temsil eder. addTrendToBolgeList metodu, belirli bir bölgeye belirli bir trendi ekler.

Tweet Bilgisi ve İlgili İşlemler:TweetBilgisi iç içe sınıfı, bir tweet'in kullanıcı adını ve metni içerir. getIlgiAlaniniIcerenTweetler metodu, belirli bir kullanıcının belirli bir ilgi alanını içeren tweetleri getirir. Bağlantılı Kullanıcılar ve İlgili İşlemler: findAndPrintConnectedUsers metodu, belirli bir kullanıcı grubundaki bağlantılı kullanıcıları bulur ve yazdırır.

Kelime Frekans Sınıfı: KelimeFrekans iç içe sınıfı, bir kullanıcının bir kelimeyi kaç kez kullandığını temsil eder. Bu kodun temel amacı, kullanıcılar arasındaki ilişkileri, ortak kelime kullanımlarını ve dil bölge trendlerini analiz etmektir. Ayrıca, her kullanıcının en çok kullandığı kelimeleri ve bu kelimelerin geçtiği tweetleri raporlamaktadır.

User Sınıfı:
Twitter verilerini temsil etmek üzere tasarlanmış bir Java sınıfıdır. Her bir kullanıcı için özellikleri ve aktivitelerini içerir. Sınıfın özellikleri şunlardır: kullanıcı adı (username), adı (name), takipçi sayısı (followersCount), takip ettiği kişi sayısı (followingCount), konuştuğu dil (language), bulunduğu bölge (region), kullanıcının attığı tweet'ler (tweets), takip ettiği kişiler (following), ve takipçileri (followers).Sınıf, bu özelliklere erişim

sağlamak için getter metotları içerir. Constructor metodu, bir kullanıcının tüm bilgilerini alarak bir "User" nesnesini oluşturur. Bu nesne, programın diğer kısımlarında kullanılmak üzere Twitter verilerini taşır.

MyLinkedList Sınıfı:
Bu sınıf, bağlı liste işlemlerini gerçekleştirmek üzere kullanılır. size(), get(int index), add(T data), contains(T data) gibi temel bağlı liste işlemlerinin yanı sıra, sıralama işlemi için sort(Comparator<T> comparator) metodunu içerir. Sıralama işlemi, kabarcık sıralama (bubble sort) algoritması kullanılarak gerçekleştirilir. Ayrıca, bağlı listenin üzerinde iterasyon yapmak için iterator() metodu ve bunu destekleyen bir iç içe sınıf (LinkedListIterator) bulunmaktadır.Bağlı liste, her biri bir veri elemanını temsil eden düğümlerden oluşur. Bu düğümler, Node<T> adlı iç içe sınıf aracılığıyla tanımlanır. Her bir düğüm, veri elemanını (data) ve bir sonraki düğümü (next) içerir. Bu sınıf, genel amaçlı bağlı liste işlemleri yapmak isteyen programcılara olanak tanıyan bir arayüz sunar.

MyList Sınıfı:
MyList, genişletilebilir bir kapasiteye sahip nesne dizisini yönetir ve bir dizi liste işlemini destekleyen çeşitli metodlara sahiptir. add, get, subList, ensureCapacity gibi temel işlemlerin yanı sıra isEmpty, size, toString, iterator, contains, stream, addAll, toArray, ve sort gibi işlevlere de sahiptir. Bu metodlar, liste üzerinde öğe ekleme, alma, alt liste oluşturma, kapasite kontrolü gibi temel işlemleri gerçekleştirmek için kullanılır. İçerdiği iç sınıf olan MyListIterator, listenin üzerinde yineleme yapmak için bir iterator sağlar. Bu kod, genel liste operasyonlarını destekleyen dinamik bir dizi uygulamasını temsil eder.

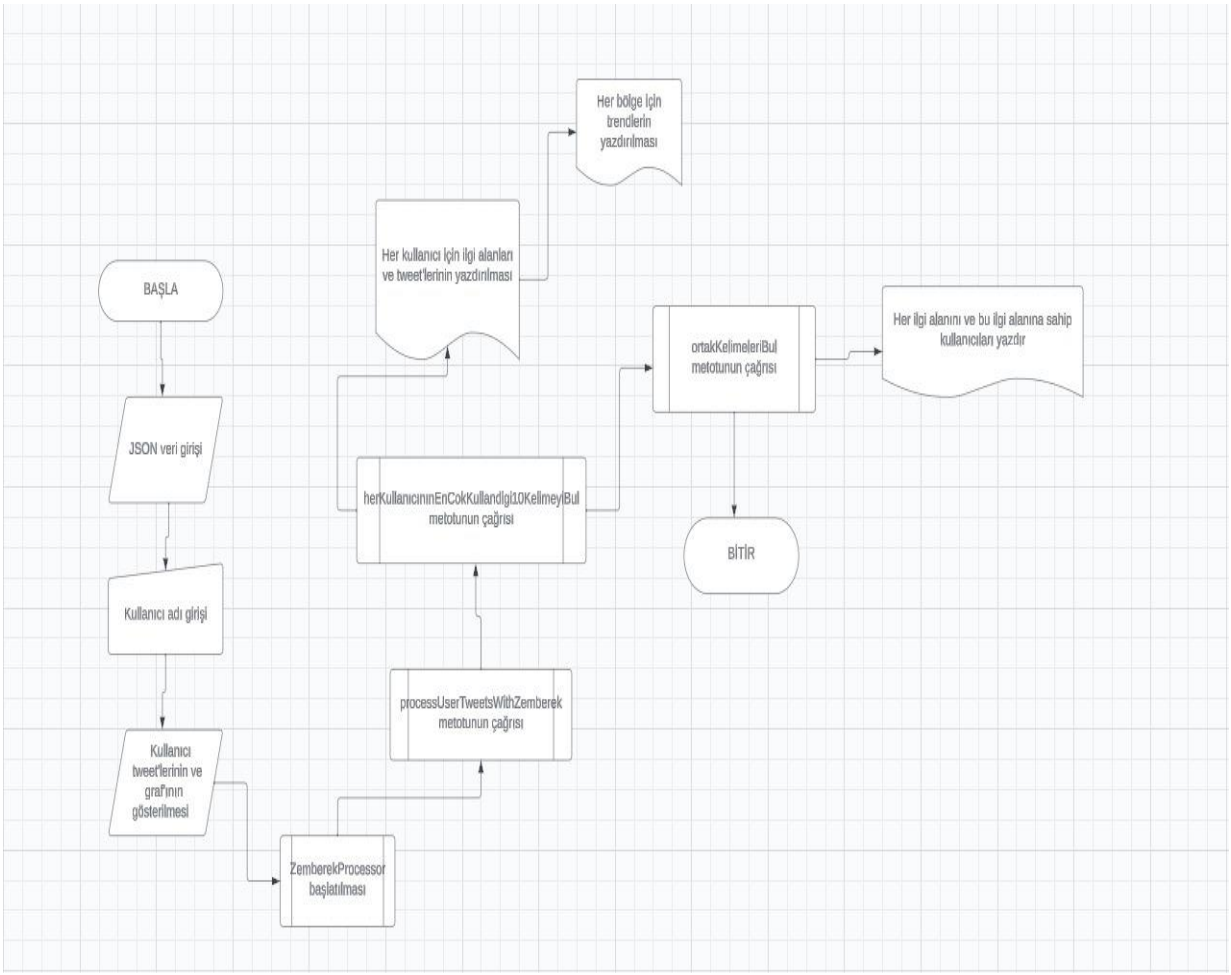
4.Sonuç:
Program veri çekme ve verileri detaylı bir şekilde işleme isterini başarıyla yerine getiriyor. Dil işleme tekniklerini uygulama (ilgi alanı hesaplarken kelime frekans hesaplama), istatistiksel bilgiler üretme (kullanıcıların ilgi alanlarının saptanması, ilgi alanı ortak olan kullanıcıların bir arada gösterilmesi...) ve bu bilgileri grafiksel bir arayüz üzerinden kullanıcıya gösterme isterlerini de başarıyla yerine getiriyor.

Yazar Katkıları:
Ekip çalışmasının çok önemli olduğunun farkında olduğumuz için her bir aşamada fikir alışverişinde ve sürekli etkileşimde bulunarak projeyi birlikte tamamladık.

KAYNAKÇA:

- https://www.youtube.com/watch?v=k2bsAEkB5l4&ab_channel=algoritmauzmani
- https://melikebektas95.medium.com/zemberek-k%C3%BCt%C3%BCphanesi-ile-t%C3%BCrk%C3%A7e-metinlerde-kelime-k%C3%B6klerinin-bulunmas%C4%B1-6ddd3a875d5f
- https://ugurozker.medium.com/zemberek-nlp-7add032881e9
- http://omercetin.com.tr/DERS/VY/Konu-11-Graflar.pdf?i=3
- https://www.freecodecamp.org/news/md5-vs-sha-1-vs-sha-2-which-is-the-most-secure-encryption-hash-and-how-to-check-them/
- https://metatime.com/tr/blog/hash-fonksiyonu-hash-function-nedir
- https://academy.patika.dev/courses/veri-yapilari-ve-algoritmalar/hash-collision
- https://code.google.com/archive/p/json-simple/downloads
- https://github.com/ahmetaa/zemberek-nlp

- ***AKIŞ ŞEMASI***


Her bölge için trendlerin yazdırılması
Her kullanıcı için ilgi alanları ve tweet'lerinin yazdırılması
BAŞLA
ortakKelimeleriBul metotunun çağrısı
Her ilgi alanını ve bu ilgi alanına sahip kullanıcıları yazdır
JSON veri girişi
herKullanıcınınEnCokKullandigi10KelimeyiBul metotunun çağrısı
BİTİR
Kullanıcı adı girişi
processUserTweetsWithZemberek metotunun çağrısı
Kullanıcı tweet'lerinin ve graf'ının gösterilmesi
ZemberekProcessor başlatılması

- ***UML DİYAGRAMI***