

Yazılım Laboratuvarı II - I

Mehmet Ali Akdoğan

200202017

Eren Sezer

200202025

I. ÖZET

Bu rapor Yazılım Laboratuvarı II Dersinin 1. Projesini açıklamak ve sunumunu gerçekleştirmek amacıyla oluşturulmuştur. Bu proje IntelliJ IDEA, MongoDB ve Visual Studio kullanılarak geliştirilmiştir. Raporda projenin tanımı, özet, yöntem, sözde kod, sonuç bölümünden oluşmaktadır. Proje aşamasında yararlanılan kaynaklar raporun son bölümünde bulunmaktadır.

II. GİRİŞ

Bu projede, kitap, dergi vb. gibi metinlerden elde edilen cümleler/kelimeler arasındaki benzerlik, frekans gibi özelliklerin bulunması, metinlerin birleştirip birleştirilmeyeceğine karar verilmesi ve elde edilen verilerden karakter tabanlı cümlelerin/kelimelerin birleştirilmesi beklenmektedir.

Amaç:

Metin birleştirme işlemleri için JAVA dili, veri tabanı MongoDB (NoSQL), Web arayüzü için ise istediğiniz bir programlama dilini kullanarak iki veya daha fazla metnin uygun bir şekilde birleştirilmesi problemi için yeni algoritmalar geliştirmek amaçlanmaktadır. Devamında bunları görsel bir arayüzle sunmamız beklenmektedir. Geliştirdiğimiz algoritmanın performans özelliklerini ve çalışma sürelerini farklı girdilerde karşılaştırtırıyoruz.

III. YÖNTEM

Projeyi iki ana başlık altına böldük. Birisi backend yani algoritma ve veritabanı tarafı diğeri ise arayüz tarafıydı. Bu iki taraf arasındaki bilgi alışverişini api ile sağladık. Backend için Java Spring frontend için ise react kullandık. Veritabanı kısmında ise mongoDB kullandık.

IV. VERİTABANI

Veritabanı için noSQL bir veritabanı olan MongoDB kullanıldı. SQL veritabanlarının aksine ilişkiler olmadığından normalizasyon yapılmadan veritabanı oluşturuldu. Her bir birleştirilmiş metin için eşsiz bir döküman id'si oluşturmak için sırayla artan id tercih edildi.

Kaydedilme şekli ise verilen metinleri array şeklinde birleştirilmiş metinleri ise conclusion adında string bir değişkende tutuldu.

V. ARAYÜZ YAPISI

Öncelikle arayüzü oluşturmak için ReactJS frameworkunu kullandık. Basit bir arayüz tasarımı hedefledik.

Algoritmaya veriyi sistemli bir şekilde gönderebilmek için array yapısı kullanmayı seçmiştik. Bu yapıya göre bir text input arrayi tanımladık. Input ekle tuşuna basınca bu arraya bir eleman ekleyip ekrana fazladan bir input girişi eklemeyi başardık. Input girişlerini ve butonları alt alata ve merkezde paddingli bir şekilde koyarak güzel bir görünüm elde etmeye çalıştık. Butonlara renk vererek de biraz şirinlik katmayı hedefledik :)

Sonuçları göstermek için sadece sonuçlar apiden gelince gözükecek şekilde iki tane textarea kısmı ayarladık. Sadece sonuçlar gelince gözükyor. Ekstardan default olarak gözükecek şekilde veritabanından gelen verileri gösteren bir tablo ayarladık. Bu tablo ilgili tuşa basılınca daha önce yapılp kaydedilen metin birleştirme işlemlerini ve geçen algoritma zamanlarını gösteriyor.

Arayüzün api isteklerini ise basit bir axios kütüphanesi kullanarak hallettik.

VI. ALGORITMA YAPISI

Arayüzden api ile gelen girilen inputları içeren arrayı alıp kendi değişkenimize atadık.

Bu arrayı ilgili fonksiyona atarak işlemleri yapmasını sağladık. Bu fonksiyonda daha iyi bir şekilde string metotlarını kullanabilmek için stringbuilderi kullandık. Aldığımız arrayı birleştirmek için önce kelimemi yoksa cümlemi diye ayırt etmek için boşluk sayısına göre kelimeleri böldük eğer tek kelimeli inputlar ise kelime birleştirmesi değilse cümle birleştirmesi kodlarını çalıştırdık.

Cümle birleştirme yapısına geçilince yapılan işlemler şu şekildedir:

Önce inputlardaki bütün nokatalama işaretlerini birleştirme sırasında sıkıntı çıkarmaması için sildik. Ayrıyeten bütün

harkleri küçüğe çevirdik. Sonra her indisi alacak şekilde for döngüsü oluşturduk .Bu döngüde her indiste dönecek şekilde indisler arası eşleştirme almaya çalıştık. Eğer eşleşme olursa geçiçi değişkenimize atadık. Böylece tekrar eden elemana kadar geçiçi değişkenimize tekrarı alana kadar kelimemizi atadık. Tekrarı alınca sonraki indise geçtik. Bu şekilde indisten indise eşleşme ala ala döğü devam etti.

Bu döngü sonrası eşleşme elde edilen kelimelerin 1den fazla yazılmasıydı. Bu sorunu aşmak için geçiçi değişkenimizi alıp içinde aynı şekilde bulunan veya çalıştırdığımız contains() fonksiyonuna true değeri veren değerleri değişkenden sildik. Bu şekilde düzgün bir output elde ettik. Geçiçi değişkeni alıp kelimeer arası boşluk koyup nihai sonucu elde ettik ve bunu return edip api ile ilgili yerlere gönderdik.

Kelime birleştirme yapısı ise genel manada aynı şekildedir. Farklılıkları ise :

- 1-) Noktalama işaretlerini silmez.
- 2-)Nihai outputa boşluk eklemesi yapmaz.
- 3-)Aynı harfi silerken contains() fonksiyonunu çıkarmaz.

VII. SÖZDE KOD

React(Arayüz) Kısımı

- 1-)Tanımlanan text arraydeki index sayısına göre textarea build et.
- 2-)Altına tuşları build et
- 3-)Add input tuşuna basılınca text arrayine ekleme yap ve build edilen input sayısı artsın
- 4-)Yolla butonuna basılınca api ile text arrayini backende yollasın
- 5-)Veritabanından veri göster tuşuna basılınca api ile veri çekip gelen verileri göstereceğin
- 6-)Eğer apiden sonuç geldi ise metin sonucu textarea kısmını build et.
- 7-)Eğer apiden sonuç geldiyse zaman sonucu textarea kısmını build et

Java Spring(Backend-Algoritma) Kısımı

- 1-)Api ile ilgili data arrayını al ve Text Merge classındaki birleştir fonksiyonuna git.
- 2-)Arrayin indislerine girip ” ”a göre split işlemi yap
- 3-)Split sonucu çıkan arrayın uzunluklarına bakıp ==1 durumuna göre kelimemi değilmi durumuna göre if else durumuna gir
- 4-)Eğer if durumu ise cümle durumudur else durumu ise kelime durumudur.
- 5-)Cümle durumu ise önce geçiçi stringbuilder, hashset, arraylist değişkenlerini oluştur ve datadaki tüm noktalama işaretlerini sil.
- 6-)Alınan data arrayındaki her indisi kelimelerine bölüp for

döngüsüne gir.

- 7-)Eşleşme olana kadar dizinin elemanlarını başka bir diziye aktar.
- 8-)Eşleşme olunca diziye son elemanı yazdır ve diğer indise geç.
- 9-)Bütün indisler bitene kadar karşılaştırma yap.
- 10-)Eşleştirme işi bitince son elde edilen veriyi tekilleştir() fonksiyonuna sok
- 11-)Her indisi gez sonraki indisi ile eşitse veya contains() ise o indisi sil.
- 12-)Sonuç dizisini aralara boşluk charını koyarak Stringe ata ve api ile ilgili arayüze yolla.
- 13-)Arada geçen zamanı hesapla ve ilgili api ile yolla.
- 14-)Kelime durumu ise her harfi dizilere yolla.
- 15-)Bu dizilerin indisinde aynı harf olana kadar aynı şekilde eşleşme olana kadar geçiçi diziye veri aktararak döngüyü devam ettir.
- 16-)Aynı harf olunca diğer indise geç.
- 17-)Bütün döngü bitince aynı olan yan yana ikili harfleri çıkar.
- 18-)Outputu ilgili api ile arayüze yolla.
- 19-)Geçen zamanı ilgili api ile arayüze yolla.
- 20-)Eğer veritabanından veri isteniyorsa ilgili api ile veritabanına git.
- 21-)Gelen verileri ilgili api ile arayüze yolla.

VIII. DENEYSEL GÖRÜNTÜLER

Yazılım Laboratuvarı II - I.Proje

Text 1:

Text 2:

Text Ekle Textleri Birleştir Textleri Kaydet

Databasedeki verileri getir

Databasedeki Veri Listesi

Geçen Süre: ns

Geçen Süre: ns

Default Görünüm

Yazılım Laboratuvarı II - I.Proje

Text 1:

Ali eve gel

Text 2:

eve gelmeden okula git

Text Ekle

Textleri Birleştir

Textleri Kaydet

Databasedeki verileri getir

Düzenlenmiş metin: Ali eve gelmeden okula git

Geçen süre: 1006900 ns

Örnek Input

Yazılım Laboratuvarı II - I.Proje

Text 1:

abcefg

Text 2:

abc

Text 3:

defgh

Text Ekle

Textleri Birleştir

Textleri Kaydet

Databasedeki verileri getir

Düzenlenmiş metin: abcefgd

Geçen süre: 2475300 ns

Örnek Input

Yazılım Laboratuvarı II - I.Proje

Text 1:

Ali eve gel

Text 2:

gelmeden okula git

Text 3:

okula gitmeden arkadaşına gitmeyi unutma

Text Ekle

Textleri Birleştir

Textleri Kaydet

Databasedeki verileri getir

Düzenlenmiş metin: Ali eve gelmeden okula gitmeden arkadaşına gitmeyi unutma

Geçen süre: 184100 ns

Örnek Input

Yazılım Laboratuvarı II - I.Proje

Text 1:

Text 2:

Text 3:

Text 4:

Text 5:

Text 6:

Text Ekle

Textleri Birleştir

Textleri Kaydet

Databasedeki verileri getir

Çoklu Input Girişi

Yazılım Laboratuvarı II - I.Proje

Text 1: [object Object]

Text 2: [object Object]

Text 3: [object Object]

Text 4: [object Object]

Text 5: [object Object]

Text 6: [object Object]

Text 7: [object Object]

Text 8: [object Object]

Text 9: [object Object]

Text 10: [object Object]

Text Ekle

Textleri Birleştir

Textleri Kaydet

Databasedeki verileri getir

musluk

mustafa

Geçen Süre: 27500 ns

musluk

Ali eve gel

gelmeden okula git

okula gitmeden arkadaşına gitmeyi unutma

Geçen Süre: 65700 ns

Ali eve gel gelmeden okula git gitmeden arkadaşına gitmeyi unutma

Veritabanından Veri Çekme

IX. SONUÇ

Input alınabilen ve sonucu gösterebilen arayüzünden alınan metinleri doğru şekilde birleştiren

Birleştirdiği algoritmanın zamanını ve sonucunu web uygulamasında gösteren , sonucu veritabanında gösteren, istenildiği takdirde veritabanındaki verileride getiren bir uygulama yaptık.

Ve bu uygulamayı size sunuyoruz.

X. KAYNAKÇA

noSQL veritabanı için;

- w3schools.com

Java IDE, ReactJs Framework ve genel sorunlar için;

-jetbrains.com

-reactjs.com

-stackoverflow.com

LaTeX Raporu hazırlamak için gerekli ekipman ve bilgiler;

- tex.stackexchange.com,- www.overleaf.com