

Yazılım Laboratuvarı 2

Proje 2

Ömer ARAN

190202012

Bilgisayar Mühendisliği
190202012@kocaeli.edu.tr

Umut SÜTCÜ

200202038

Bilgisayar Mühendisliği
umutsutcouk@outlook.com

ÖZET

Bu doküman yazılım laboratuvarı 2 dersinin 1.projesi metinlerin uygun bir şekilde birleştirildiği bir websitesi yapma ve bu kelime ve cümlelerin nosql tabanlı bir veritabanında saklanmasıdır. Bu işlemleri yaparken kullandığımız algoritma dili java nosql tabanlı veritabanı için de mongodb kullandık. Bu projede amacımız verilen metinleri en uygun biçimde birleştirmek ve bu birleştirme sırasında geçen süreyi minimum düzeyde tutmaktır. Birleştirilmesi gereken cümleler ve cümlelerin birleştirilmiş son halini de mongodbye kaydederek veri kaybını engelliyoruz.

PROJENİN TANIMI

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. Bu işlemlerin yapılması için gerekli algoritma geliştirilecektir. Bu algoritmada herhangi bir sınırlama yoktur. Bütün cümleleri birleştirebilecek bir algoritma yazılmalıdır.

Projeyi 3 farklı tabana bölebiliriz bunlar algoritma, web arayüzü ve veritabanı bunlar için farklı isterler aşağıda belirtilmiştir. web arayüzünde en az 2 metin girmemiz gerekir bunun içi: En az iki tane metin giriş butonları bulunmalıdır. (Metin Ekle) Girilen metinleri birleştirme butonu bulunmalıdır. (Metinleri Birleştir) Girilen metinleri kaydetme (.json formatında) butonu (Birleşik Metni Kaydet) Veriden elde edilen en uzun dizi listelerinin tümü bir liste halinde kaydedilmeli İki metni birleştirmek işlemi için geçen süre hesaplanmalıdır. Görsel arayüzde iki adet metin giriş etiketi ve bu etiketlere ait “ekle”, “birleştir” ve “kaydet” butonları bulunmalıdır.

Arayüz isterleri olarak 3 farklı isterimiz vardır bunlar aşağıda belirtilmiştir. Metin giriş butonları Burada belirlenen bir metin kaynağından verilerin girilmesi beklenmektedir. Girilen metinler kelime veya cümleler olabilir. Ayrıca

istenilen sayıda metin girişine izin verilmelidir. Yani kullanıcı 2 veya daha fazla metinde birleştirebilmelidir. Girilen metinleri birleştirme butonu Kullanıcı tarafından kaydedilen dökümanlar “birleştir” butonuna tıklanıldığında kalan karakterlerin görelisi sırasını değiştirmeden karakter tabanlı olarak birleştirilmelidir. Burada; Girilen metinler hangi karakterden itibaren birleştirilebileceğine yada girilen metinlerin birleştirilip birleştirilmeyeceğine karar verme işlemi algoritma kullanılarak gerçekleştirilecektir Girilen metinleri kaydetme butonu Kullanıcı sisteme girdiği metinleri kaydedip listeleyebilmelidir. Girilen metinler .JSON formatında döküman olarak NoSQL tabanlı saklanmalıdır.

Algoritma isterleri olarak kullanılan tek koşul yapayzeka kütüphaneleri kullanmamaktır. Bunun dışında algoritma bütün koşulları sağlayacak şekilde geliştirilmeli ve birleştirme süresinde geçen süre hesaplanmalıdır.

Veri tabanında birleştirilen yeni metin JSON formatında kaydedilmeli ve geliştirilen algoritmanın verilen metinler üzerindeki çalışma süresini de kaydetmelidir. Program Tablosu: girilen her bir metin kaydedilmeli (metin 1 , metin2 , metin3 ...), bu metinlerin birleştirilmiş hali (sonuç) ve birleştirme sırasında geçen geçen süreyi de sn cinsinden tutarak kaydedilmeli.

ARAŞTIRMA VE YÖNTEMLER

İlk önceliğimiz projeyi anlayarak bir iş şeması çıkartmak oldu. Projede yapılması gerekenleri listeleyerek ve bu yapılacakları bir düzen içinde aramızda bölüştürmek oldu. Projemizde programlama dili olarak java ve veritabanı olarak nosql tabanlı mongodb kullanılacaktı. Web için dil herhangi bir dil sınırlaması yoktu. Bu koşullardan sonra aramızda iş paylaşımı yaparak projeye başladık.

İlk olarak metinleri birleştirmede nasıl bir algoritma yazmamız gerektiği üzerinde durduk ve yazılması gereken algoritmanın neleri içermesi gerektiğini ve nasıl bir birleştirme yapması gerektiğini kararlaştırdık. Bundan sonra kod yazım aşamasına başladık

```

for (int i = 0; i < n-1; i++) {
    String[] words1 = texts[i].split(regex: " ");
    String[] words2 = texts[i+1].split(regex: " ");

    for (String word1 : words1) {
        for (String word2 : words2) {
            // Eğer iki kelime aynı ise, commonWords'e ekle
            if (word1.equals(word2)) {
                commonWords.add(word1);
            }
        }
    }

    for (String word : commonWords){
        if(ftexts[i+1].contains(word)){
            ftexts[i+1] = ftexts[i+1].replace(word, replacement:"");
        }
    }

    if((words2[0]).contains(words1[words1.length-1])){
        ftexts[i] = ftexts[i].replace(words1[words1.length-1], replacement:"");
    }

    commonWords.clear();
}

```

Girilen metinleri birleştirmek için kullandığımız kodun bir parçası

Metinleri birleştirirken birçok hatayla karşılaştık. Bu hataların en istenen şekilde birleştirmeme geliyordu. Genelde istenen şekilde birleştirmiyor ya 1. metindekileri tamamen siliyor ya da 2. metindekileri tamamen siliyordu. Bunlar dışında birden fazla metin girildiğinde ise tamamen çalışmıyordu. Bu sorunlar için basit bir çözüm bulduk. 1. ve 2. metinleri kontrol etmek. Bu kontrol mekanizmasında aynı anda birden çok yerde geçen kelimeler aynı olarak algılanıp silinmeyecek sadece bir önceki metinde geçen kelimeler kontrol edilecekti. Bu sayede birleştirme sırasında yaşadığımız birçok sorun bu sıralı kontrol sayesinde kalkmış oldu.

```

public static String generateFromMultipleString(List<String> sentences) {
    String result = "";
    for (String sentence : sentences) {
        String[] words = sentence.split(regex: " ");
        for (String word : words) {
            if (!result.contains(word)) {
                result += word + " ";
            }
        }
    }
    return result.trim();
}

```

İki Cümleyi kontrol eden kod parçası

Bulduğumuz çözüm 2 metni aynı anda kontrol edip diğerine geçiyordu. İlk önce 1. metin ve 2. metin'i karşılaştırıyor ve benzer kelimeleri eliyor bundan sonra 2.metin ve 3.metin'e geçiyordu. Bu şekilde girilen metin sayısı fark etmeksizin karşılaştırmayı yapmaya devam ediyor. Bu karşılaştırmamızın bir fark etmediği bir sorun vardı. Bu soru devam eden kelimelerde (çarşı, çarşıya) ikisini farklı kelime olarak alarak düzgün birleştirme yapmıyordu.

```

if((words2[0]).contains(words1[words1.length-1])){
    ftexts[i] = ftexts[i].replace(words1[words1.length-1], replacement:"");
}

commonWords.clear();

```

Kelime kontrolü için kullanılan kod parçası

Bu birleştirmeyi yapabilmesi için bir koşul daha ekledik. Bu koşulda 1.metin'in son kelimesini ve 2. metin'in ilk kelimesini karşılaştırıyor. Bu karşılaştırmada 1.metin'in son kelimesi 2.metin'in ilk kelimesinin içinde geçiyorsa 1.metin'in son kelimesini silerek birleştirme işlemini yapıyor. Örneğin 1.metin'in son kelimesi çarşı ve 2. metnin ilk kelimesi çarşıya burada çarşı kelimesi çarşıya'nın içinde geçtiği için çarşı kelimesini silerek sadece çarşıya kelimesini bırakıyor.

Aynı zamanda bu kod parçası tek kelimeli cümleler için de çalışıyor. Eğer bir cümlede tek kelime varsa o kelime aynı anda cümledeki kelime içinde geçip geçmediğini kontrol ederek, ortak bir birliktelik varsa bunu silerek tek kelime haline getiriyor.

```

public class Content {
    @Id
    private String id;
    private String[] inputTexts;
    private String result;
    private Long elapsedTime;

    public Content(String[] contents, String result, Long elapsedTime) {
        inputTexts = contents;
        this.result = result;
        this.elapsedTime = elapsedTime;
    }
}

```

Verileri veritabanına kaydetmek için kullandığımız değişkenler

Verileri veritabanına kaydederken en çok yaşadığımız sorun girilen metin sayısının belli olmamasıydı. Bunun için farklı çözümler ararken çok basit bir çözümle hallededebildik. Bunun için girilen metinleri bir string listesine atamak oldu. Bu sayede girilen metin sayısı fark etmeksizin veritabanına rahat bir şekilde kaydedebiliyor olduk.

```

function saveDB(){
    contentService.saveContent(content);
    alert("Bazarıyla kaydedildi.");
}

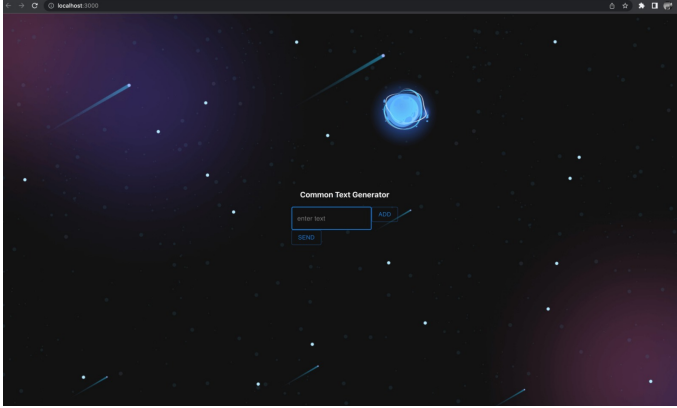
return(
    <Container className="content">
        {
            submitted ?
            <div className="result">
                <div className="header">You submitted successfully</div>
                <div>
                    <div className="header">{content.result}</div>
                    <div className="header">{content.elapsedTime}</div>
                </div>
                <button variant="outlined" onClick={tryAgain}>try again</button>
                <button type="submit" variant="outlined" onClick={saveToDB}>Save to DB</button>
            </div> :
            <div>
                <div className="header">Common Text Generator</div>
                <div>
                    {
                        inputList.map( (x,i) => {
                            return(
                                <div raised={true} children={["node"]}
                                <TextField id="outlined-basic" variant="outlined"
                                    type="text" name="text"
                                    sx={{ input: { color: 'white' } }}
                                    placeholder="enter text"
                                    color="primary" focused
                                    onChange={e=>handleInputChange(e,i)} />
                                {
                                    inputList.length-i==1 &&
                                    <button className="form-control" variant="outlined" onClick={()=> handlerRemove(i)}>Remove</button>
                                }
                                {
                                    inputList.length-i==1 &&
                                    <button variant="outlined" onClick={handleAddClick}>Add</button>
                                }
                            </div>
                        )
                    }
                </div>
            </div>
        )
    </div>
)

```

Web sitesi arayüzü için kullandığımız kodun bir parçası

Bu kısımda aldığımız hataların en başında girilen verileri boş olarak göndermesiydi bu yüzden boş sonuç döndürüyordu.

Bu hatayı çözdükten sonra istenilen sayıda cümle girilebilmesi için bir ekleme butonu ve bu butondan sonra açılan pencere-deki verileri düzgün bir şekilde gönderebilmek oldu.



Verileri girdiğimiz ana ekran Bu sayfada metinlerin girilmesi için bir bölüm ve girilen metin sayısını artırabilmek için bir ekleme tuşu bulunuyor. Metinleri girdikten sonra gönder tuşuyla metinleri birleştirme işlemi yapıyoruz.

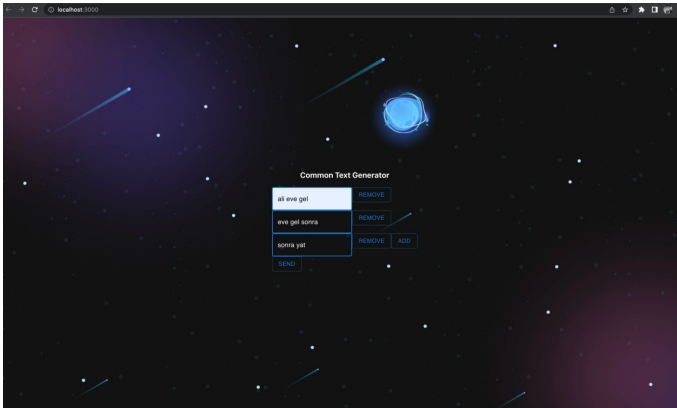
KOD BİLGİSİ(YALANCI KOD)

Yalancı Kodu txt dosyası olarak ek olarak gönderdim.

SONUÇ

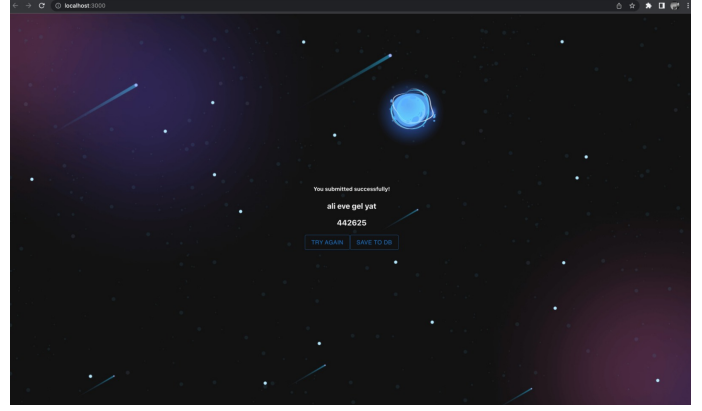
Sonuç olarak java ile yazdığımız bir metin birleştirme algoritmasını kullanıcıların rahat bir şekilde kullanabileceği bir web sitesi yapmış olduk. Kullanıcıların girdiği cümleleri daha sonra görebilmesi için veritabanına kaydederek girilen verilerin kaybolmamasını da sağlamış olduk.

Aşağıda da projemize ait ekran görüntüleri:



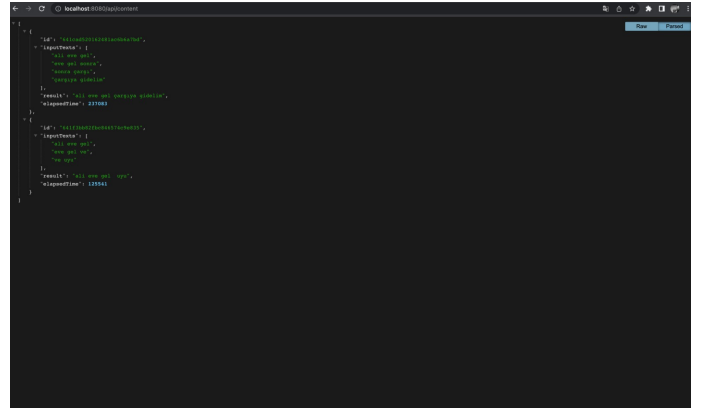
Projenin giriş ekranı

Bu sayfada çoklu metin giriş ekranı bulunuyor bu ekranda ekleme tuşu ile daha fazla metin girişine izin veriyoruz. Gönder tuşu ile de birleştirme işlemi yapıyoruz.



Projenin cümleleri birleştirdikten sonraki çıktı ekranı

Bu ekranda birleşecek cümleleri görüyoruz ve birleştikten sonra nasıl bir çıktı haliyle birleştiklerini görüyoruz. Çıktının altında geçen süreyi nanosaniye olarak gösteriyoruz. Bunların altında da 2 tane buton bulunmakta bu butonlar database'e kaydetmeye ve tekrar denemeye yarıyor.



Verilerin database'e kaydedilme şekli

Database'de her birleştirmeye ait bir id tutuyoruz. Birleştirilecek cümleleri string listesi olarak tutmaya karar verdik çünkü girilecek metin sayısını bilmediğimiz için girilen metin sayısı farketmeksizin database'e kaydetmemizi sağlıyor. Geçen süre için nanosaniye kullanıyoruz bunu database'de saklamak için de bir long int değişkeni kullanıyoruz.

KAYNAKÇA

<https://www.geeksforgeeks.org/stringbuilder-class-in-java-with-examples/?ref=lbp>

<https://www.youtube.com/watch?v=HB0T0hAMk0klist=PLqG356Exox2>

<https://www.javatpoint.com/spring-aop-tutorial>

<https://hyperskill.org/tracks/32?gl> =
12af8f4uh2a9a2aMTc1NjI5MjAzNC4xNjc5ODM5MDY02a9a9.2.54275048.1145886078.1679839065 -
1756292034.1679839064

<https://www.javatpoint.com/spring-modules>

<https://www.baeldung.com/spring-mvc>

https://www.tutorialspoint.com/spring/spring_dependencies_autowiring.htm

<https://docs.google.com/document/d/18CXhDb1ygxg-YXNBJNzfzZsDFosB5e6BfnXLlejd9l0/edit>

<https://www.w3schools.blog/spring-tutorial>

https://www.w3schools.com/REACT/react_memo.asp