



**KOCAELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ**

**YAZILIM LABORATUVARI-II
PROJE -1**

**Engin Tosun
200202028**

**Eray Karataş
200202079**

METİN BİRLEŞTİRME WEB PROJESİ

1) Özet:

Bu rapor Yazılım Laboratuvarı II Dersinin 3.Projesini açıklamak ve sunumunu gerçekleştirmek amacıyla oluşturulmuştur. Bu proje Java diliyle kelime birleştirme işlemleri yazılmıştır. Web tarafı için ise Java Spring MVC yapısı kullanılmıştır. Raporda, özet, giriş, yöntem, karşılaşılan sorunlar ve çözümler, sözde kod, deneysel sonuç bölümünden oluşmaktadır. Proje aşamasında yararlanılan kaynaklar raporun son bölümünde bulunmaktadır.

2) Giriş:

Bu projede, kısaca yapılması gereken konu açıklanmak istenirse; 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.

Projenin genel amacı, iki veya daha fazla kullanıcıdan alınan kelimeyi gerekli controller sonrasında birleştirmeyi devamında görsel bir şekilde web arayüzü kullanılarak sunulması istenmiştir. Kelimeleri birleştirme süresi çeşitli örnekler ile test edilmesi istenmiştir.

Metin birleştirme işlemleri için JAVA dili, veri tabanı MongoDB (NoSQL), Web arayüzü için ise herhangi bir dil kullanılması koşulu verilmiştir.

Projede istenen isterler, web arayüzünde metin giriş kısımları, metin girişlerine girilen metinleri birleştirmek için birleştir butonu, son olarak birleştirilen metni .json formatında kaydetme butonu (kaydet) olarak arayüzde istenmiştir. Görsel arayüzde iki adet metin giriş etiketi ve bu etiketlere ait “ekle” , “birleştir” ve “kaydet” butonları bulunmalıdır.

Arayüz isterleri:

Metin giriş butonları, iki veya daha fazla giriş butonu istenmektedir. O yüzden arayüzde giriş alanları manuel artırılması gerekir.

Girilen metinleri birleştirme butonu, birleştir butonuna tıklandığında karakterlerin göreceli sırasını değiştirmeden karakter tabanlı olarak birleştirilmesi istenmektedir.

Girilen metinler hangi karakterden itibaren birleştirilebileceğine yada girilen metinlerin birleştirilip birleştirilmeyeceğine karar verme işlemi geliştirdiğimiz algoritma aracılığıyla gerçekleştirilecektir.

Girilen metinleri kaydetme butonu, veritabanına kaydetme işlevli bir buton istenmiştir. Girilen metinler .json formatında Mongo DB’de saklanması gerekmektedir.

Örnek: "abcdefgh" ve "abcefg" kelimelerinin en uzun ortak ön eki "ABC" dir. Sonra, bu ön ek birleştirilecek karakterleri birleştirerek veritabanına kaydedilmelidir.

Veya başka bir örnekle tanımlamak gerekirse;

Text 1: Ali eve gel

Text 2: eve gel sonra

Text 3: eve gel sonra çarşı

Text 4: çarşıya git

Çıktı: Ali eve gel sonra çarşıya git

Girdileri verilen metinleri şeklinde birleştirmesi gerekir.

Veritabanı istekleri:

Veri tabanında birleştirilen yeni metin JSON formatında kaydedilmeli ve geliştirilen algoritmanın verilen metinler üzerindeki çalışma süresini de kaydetmelidir.

3.YÖNTEM

Projede birleştirme kısmında Java dilini ve web arayüzünde Spring MVC’yi veritabanı olarak Mongo DB’yi kullandık. İlk başta IntelliJ’de Spring projesi oluşturup model, controller , repository ve service katmanlarını oluşturup içlerine gerekli yapıları tanımladık.

Modelin içinde veritabanında tutulacak olan metinler , bu metinlerin birleştirilmiş halini ve metin birleştirmek için geçen süreyi karşılayacak olan değişkenleri tanımladık. Templates kısmında anasayfa, metin birleştirme sayfası, ve veritabanındaki kayıtlı verilerin gösterildiği sayfaları oluşturup controllerda bu sayfalar için mappinger tanımladık. Bu 3 sayfayı butonlarla birbirlerine bağladıktan sonra metin birleştirme sayfasında kullanıcıdan metin almak için input alanları oluşturduk.

Kullanıcının istediği kadar metin girebilmesi için static klasörü içerisinde JavaScript dosyası oluşturup içine yeni input alanı oluşturan kodları yazıp birleştirme sayfasındaki ekle butonuna basarak JavaScript tarafını çalıştırıp kullanıcının istediği sayıda metin girebilmesi için yeni input alanları ekledik.

Birleştir butonuna basıldığı zaman controllerdaki SaveText methodunu çağırdık. Bu methodun içinde modelda olan verileri ArrayList’e atıp bu ArrayListin içinde kullanıcının girdiği metinleri ayıklayıp başka bir ArrayListe attık. Bu ArrayListi kullanıp ArrayListin içindeki bir metni diğer metinlerle karşılaştırdık. Karşılaştırırken diğer metnin ilk kelimesini kıyaslanacak metnin içinde olup olmadığını kontrol ettik. Eğer içinde var ise ondan sonra gelen kelimelerin de kıyaslanacak metinde sıralı bir şekilde devam edip etmediğini bulan, benzer kelime sayısı ve ilk benzer kelimenin bulunduğu indisi hesaplayan bir algoritma geliştirdik.

Daha sonra kıyaslanacak metnin uzunluğunun ilk benzer kelimenin bulunduğu indisle arasındaki farkın benzer kelime sayısına eşit olup olmadığını kontrol edip eğer eşitse benzer kelime sayısına göre kıyaslanacak metne diğer metni aynı kelimeyi birden fazla kez yazmıyacak şekilde birleştirme işlemini yaptık. Eğer kıyaslanacak metin ile diğer tüm metinler arasında benzer kelime bulunamamışsa kıyaslanacak metnin değerini ArrayListdeki bir sonraki metin değeri ile güncelleyip o metin üzerinde karşılaştırma işlemi yaptık. Eğer metinlerin hiçbirisi birbiriyle benzer kelimeye sahip değilse birleştirilmiş metnin değerine kullanıcının en son girdiği kelimeyi atarak birleştirme kısmını tamamlamış olduk.

Veri tabanındaki kayıtlı verilerin gösterildiği sayfada ise table yapısı oluşturup içine gerekli sütunları tanımladık. Bu sayfaya verileri çekmek için controllerda service katmanındaki getShowTexts methodunun içinde olan findAll methodunu çağırarak veritabanındaki tüm verileri verilerin gösterildiği sayfada gösterme işlemi yaptık.

Yalancı Kod:

1-BAŞLA
2-ANASAYFAYA YÖNLENDİR
3-KULLANICI EĞER "Metin Birleştir" BUTONUNA BASARSA
4-METİN BİRLEŞTİRME SAYFASINI AÇ
5-KULLANICI "Ekle" BUTONUNA BASARSA YENİ INPUT ALANLARI OLUŞTUR
6-KULLANICIDAN METİNLER AL
7-KULLANICI "Birleştir" BUTONUNA BASARSA
8-INPUTLARDAN GİRİLEN METİNLERİ "ArrayList" LERE AT
9-"ArrayList" DEKİ METİNLERİN BİRİNCİSİNİ KIYASLANACAK METİN YAP VE DİĞER METİNLER İLE KARŞILAŞTIR
10-KARŞILAŞTIRMADA DİĞER METİNLER İLE KIYASLANACAK METİN ARASINDA BENZER KELİME VAR MI YOK MU KONTROLÜNÜ YAP
11-BENZER KELİMELERİN SIRALI OLUP OLMADIĞININ VE BENZER KELİME SAYISININ, KIYASLANACAK METİNİN UZUNLUĞUNUN İLK BENZER KELİMENİN BULUNDUĞU İNDİSTEN FARKINA EŞİT OLUP OLMADIĞININ KONTROLÜNÜ YAP

12-EĞER EŞİTSE KIYASLANACAK METİNE DİĞER METNİ KELİME TEKRARI OLMAYACAK ŞEKİLDE EKLE

13-KIYASLANACAK METİN İLE DİĞER METİNLER ARASINDA BENZERLİK BULUNAMAMIŞSA KIYASLANACAK METNİN DEĞERİNİ "ArrayList" DEKİ BİR SONRAKİ METİN YAP VE DİĞER METİNLERLE TEKRAR KARŞILAŞTIRMA İŞLEMİ YAP

14-TÜM METİNLERİN DİĞER METİNLERLE BENZERLİĞİ BULUNAMAMIŞSA GİRİLEN EN SON METNİ BİRLEŞTİRİLMİŞ METİN OLARAK GÜNCELLE

15-METİN BİRLEŞTİRME İŞLEMİ İÇİN GEÇEN SÜREYİ HESAPLA

16-"Model" TARAFINDAKİ VERİLERE SÜRE VE BİRLEŞTİRİLMİŞ METİNİ EKLE

17-GİRİLEN METİNLERİ, BİRLEŞTİRİLMİŞ METİNİ VE GEÇEN SÜREYİ VERİTABANINA KAYDET

18-KULLANICI "Kaydet" BUTONUNA BASARSA

19-ANASAYFAYA YÖNLENDİR

20-KULLANICI "Verileri Gör" BUTONUNA BASARSA

21-VERİLERİN GÖSTERİLDİĞİ SAYFAYI AÇ

22-VERİTABANINDA KAYITLI OLAN TÜM VERİLERİ SAYFADA GÖSTER

23-ÇIKIŞ

4. DENEYSEL SONUÇLAR:

WEB SİTESİ ARAYÜZÜ:

GİRİŞ EKRANI:

METİN GİRİŞ EKRANI:

VERİ GÖSTERME EKRANI:

Tüm Veriler										
Ana Sayfa										
1.Mein	2.Mein	3.Mein	4.Mein	5.Mein	6.Mein	7.Mein	8.Mein	9.Mein	10.Mein	Birleştirmiş Mein
Ali eve gel	eve gel sonra	eve gel sonra parş	parşıya git							Ali eve gel sonra parşıya git
										0.001

MONGO DB:

Text.text										
0 1										
DOCUMENTS INDEX										
Documents Aggregations Schema Explain Plan Indexes Validation										
Filter 0 Type a query: { field: 'value' } Reset Find More Options										
ADD DATA EXPORT COLLECTION 1-1 of 1										
{ "_id": 1, "text1": "Ali eve gel", "text2": "eve gel sonra", "text3": "eve gel sonra parşı", "text4": "parşıya git", "unified_text": "Ali eve gel sonra parşıya git", "time": 0.001, "_class": "com.example.Project.model.Text" }										

SONUÇ:

Birleştirme algoritması oluşturarak algoritmik düşünme becerimizi geliştirdik. Web arayüzü oluşturma ve Spring yapısı hakkında yeni bilgiler öğrendik.

6.KAYNAKÇA:

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#web>

https://www.tutorialspoint.com/spring_boot/index.htm

<https://www.yusufsezer.com.tr/spring-boot/>

<https://www.youtube.com/watch?v=qcfQbJr6Bi0&list=PLHHxlyNPK3PEiySM3TdqIrvSIVGGfPS43>

<https://start.spring.io/>

<https://www.mongodb.com/docs/manual/core/document/>

<https://medium.com/@mustafakoroglu/spring-boot-ile-web-sayfalar%C4%B1-thymeleaf-9613c8a6e888>

