I. Kodun İçeriği

Tarih ve tıp ile ilgili kelimeler içeren dokümanları sınıflandırabilmek için yazdığımız kod 5 ana yapıdan oluşmaktadır. Bunları şu şekilde tanımlayabiliriz:

1. Dosyaların girdi olarak alınması,
2. Dosyalardan elde edilen textlerin ön işlemden geçirilmesi,
3. Textlerin benzerlik oranlarını bulan algoritmanın çalıştırılması ve benzerlik matrisinin oluşturulması,
4. Textlerin sınıflandırılması,
5. Sonuçların yazdırılması,

olarak 5 ana başlıkta değerlendirebiliriz. Bu başlıkları sırayla inceleyelim.

1. Dosyaların Girdi Olarak Alınması

- Girdi olarak alınacak dosyaları 3 farklı yapıya ayırabiliriz:
  - **A.** Verilen text dosyaları:
    - datafile-0.txt
    - datafile-1.txt
    - datafile-2.txt
    - datafile-3.txt
    - datafile-4.txt
    - datafile-5.txt
    - datafile-6.txt
    - datafile-7.txt
    - datafile-8.txt
    - datafile-9.txt

  - **B.** Stopwords'leri ("the", "a", "for" gibi kelimeler) içeren dosya:
    - stop_words_english.txt

  - **C.** İlk başta benzerliğe göre sınıflandırma yaparken baz alacağımız dosyalar:
    - history.txt
    - medicine.txt

- **input**(char filePath[], FILE *fp, char *text) fonksiyonu ile bu işlem gerçekleştirilmektedir.

2. Ön İşlemler

- Textlere 5 ön işlem uygulanmıştır. Bunlar:
  - **A.** Newline ve tab'ların kaldırılması.
    - **preprocessing1**(char *text) fonksiyonu kullanılmıştır.
  - **B.** Ekstra whitespace'lerin kaldırılması.
    - **preprocessing2**(char *text) fonksiyonu kullanılmıştır.
  - **C.** Noktalama işaretlerinin kaldırılması.
    - **preprocessing3**(char *text) fonksiyonu kullanılmıştır.
  - **D.** Bütün karakterlerin küçük harfe çalışılması.
    - **preprocessing4**(char *text) fonksiyonu kullanılmıştır.
  - **E.** Stopwords'lerin ("the", "a", "for" gibi kelimeler) kaldırılması.
    - **preprocessing5**(char *text, char *stopWords) fonksiyonu kullanılmıştır.

3. Benzerlik Oranları ve Benzerlik Matrisi

- Bütün textlerin birbiri ile benzerlikleri hesaplanarak benzerlik matrisi oluşturulmuştur.
- **similarity**(char *text0, char *text1, char *text2, char *text3, char *text4, char *text5, char *text6, char *text7, char *text8, char *text9, char *history, char *med, float **similarityMatrix) fonksiyonu ile benzerlik matrisi oluşturulmaktadır.
- **similarity**() fonksiyonu içerisinde çağrılan **wordComparison**(char *text, char *comparisonText) fonksiyonu ile 2 text arasındaki benzerlik oranı çıkarılmaktadır.

4. Sınıflandırma

- Bütün textler benzerlik oranlarına göre "tarih" ya da "tip" olarak sınıflandırılır.
- **design**(float **matris, int *sinif) fonksiyonu kullanılır.
- **design**() fonksiyonu içerisinde çağrılan **findMax**(float **matris, int i) fonksiyonu matriste belirtilen indise sahip texte en benzer textin indisini bulmaktadır.

5. Sonuçların Yazdırılması

- Belirtilen dosyaların hangi sınıfa ait olduğu yazdırıldı.
- **resultText**(int *sinif, char filePath[], int i) fonksiyonu kullanıldı.
- 1 ile etiketlenen metinler "tarih", 2 ile etiketlenen metinler "tip" olarak ekrana yazdırmaktadır.

```
datafile-0.txt = Tip

datafile-1.txt = Tarih

datafile-2.txt = Tarih

datafile-3.txt = Tip

datafile-4.txt = Tip

datafile-5.txt = Tip

datafile-6.txt = Tip

datafile-7.txt = Tarih

datafile-8.txt = Tarih

datafile-9.txt = Tarih

--------------------------------
Process exited after 0.7567 seconds with return value 3221225477
Press any key to continue . . .
```

-

II. Tasarlanan Algoritmalar

- Bu programda 2 farklı algoritma geliştirilmiştir. Bunlar:
  1. Benzerlik,
  2. Sınıflandırma

- Benzerlik için kullandığımız algoritma sınıflandırma için kullanacağımız algoritmaya input değeri sağlamaktadır

1. Benzerlik Algoritması

- Bir texteki her bir kelimenin diğer texteki her bir kelime ile karşılaştırılarak 2 kelimenin aynı olma sayısının toplam karşılaştırma sayısına oranıdır. 0 ile 1 arası değerler alır.
- Her bir textin benzerlik oranlarını gösteren matris aşağıdaki gibidir:

	0	1	2	3	4	5	6	7	8	9	10	11
0	0.000000	0.144501	0.183504	0.167839	0.144821	0.187020	0.152813	0.140985	0.133312	0.155371	0.001598	0.000320
1	0.144501	0.000000	0.396189	0.247743	0.202106	0.275326	0.229689	0.341525	0.296389	0.339017	0.003511	0.000000
2	0.183504	0.396189	0.000000	0.252702	0.204073	0.285952	0.235661	0.352452	0.304239	0.351621	0.009144	0.000000
3	0.167839	0.247743	0.252702	0.000000	0.198638	0.259932	0.225501	0.203935	0.175558	0.163829	0.001892	0.003784
4	0.144821	0.202106	0.204073	0.198638	0.000000	0.207003	0.181600	0.133883	0.123584	0.155166	0.000687	0.001030
5	0.187020	0.275326	0.285952	0.259932	0.207003	0.000000	0.204715	0.168424	0.156017	0.177419	0.000620	0.002792
6	0.152813	0.229689	0.235661	0.225501	0.181600	0.204715	0.000000	0.163636	0.150545	0.173818	0.001455	0.002182
7	0.140985	0.341525	0.352452	0.203935	0.133883	0.168424	0.163636	0.000000	0.245275	0.274628	0.006836	0.000000
8	0.133312	0.296389	0.304239	0.175558	0.123584	0.156017	0.150545	0.245275	0.000000	0.296154	0.001923	0.000000
9	0.155371	0.339017	0.351621	0.163829	0.155166	0.177419	0.173818	0.274628	0.296154	0.000000	0.007399	0.000000
10	0.001598	0.003511	0.009144	0.001892	0.000687	0.000620	0.001455	0.006836	0.001923	0.007399	0.000000	0.000000
11	0.000320	0.000000	0.000000	0.003784	0.001030	0.002792	0.002182	0.000000	0.000000	0.000000	0.000000	0.000000

(Her bir satir ve sütun numaraları her bit texti temsil etmektedir)

2. Sınıflandırma Algoritması

- Benzerlik matrisi çıkartıldıktan sonra bütün textlerin sınıf bilgisini içeren bir dizi oluşturup bütün değerlerini 0 yaptık. Kendimizin dışarıdan eklediği tarih ile ilgili kelimeleri barındıran “history.txt” ait textin bulunduğu sınıfı 1 yaptık. Yine kendimizin dışarıdan eklediği tıp ile ilgili kelimeleri barındıran “medicene.txt” ait textin bulunduğu sınıfı 2 yaptık. Sonrasında algoritmamızı çalıştırdık. Algoritmamız bütün sınıf değerleri 0’dan başka bir değer olana kadar çalışacaktır. Benzerlik oranlarına göre dokümanlara ait textlerin en benzer olduğu textler eğer sınıf değeri 0 ise kendi sınıf değeri atanmaktadır. Eğer algoritmamız bu şekilde 14 iterasyonda hiçbir değişiklik yapamazsa sınıf değeri 0 olan en yüksek benzerlik oranını bularak benzer olduğu textin sınıf değerini atamaktadır. Sınıf atama işlemleri aşağıda olduğu gibi gerçekleşmiştir.

0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	0	0	0	0	1	2
2	0	0	0	0	0	0	0	0	0	1	2
2	1	0	0	0	0	0	0	0	0	1	2
2	1	1	0	0	0	0	0	0	0	1	2
2	1	1	2	0	0	0	0	0	0	1	2
2	1	1	2	2	0	0	0	0	0	1	2
2	1	1	2	2	2	0	0	0	0	1	2
2	1	1	2	2	2	2	0	0	0	1	2
2	1	1	2	2	2	2	1	0	0	1	2
2	1	1	2	2	2	2	1	1	0	1	2
2	1	1	2	2	2	2	1	1	1	1	2

(Her sütun en üstteki numaralandırılmış texti temsil etmektedir. İlk satır hariç her satır sıralı iterasyondur)