

BLM2642 ÖDEV



HAZIRLAYAN

ÖMER DİNER 20011017

omer.diner@std.yildiz.edu.tr

1-)

Ödeve başlamak için ilk önce 2 adet veri setine ihtiyaç duydum. Bunun için arkadaşımın kullandığı veri setini aldım ben de. İki veri seti pozitif ve negatif düşünce ve bakış açısına sahip cümleler içermesi yönüyle birbirinden ayrılıyor. Her iki sette de 100'er cümle var. Setlerden bir örnek;

Today is a beautiful day filled with endless possibilities.
I am surrounded by love and positivity.
Success and happiness are my constant companions.
I radiate confidence and attract success.
Every challenge I face is an opportunity to grow and improve.
I am grateful for the abundance in my life.
Positivity flows through me like a refreshing breeze.
I am capable of achieving anything I set my mind to.
My life is a reflection of the positive energy I put into it.
Good things are constantly manifesting in my life.
I am a magnet for joy and good fortune.
I am in control of my own happiness.
The universe is conspiring to bring me success.
I am surrounded by supportive and uplifting people.
I am a beacon of positivity, inspiring those around me.
Challenges are opportunities for growth and learning.
I am deserving of all the good things life has to offer.
My mindset is a powerful tool for creating positive outcomes.
I choose to focus on the present and create a positive future.
Each step I take is a step toward a brighter and happier life.
I attract positive experiences with my optimistic attitude.
I am filled with gratitude for the simple joys in life.
I am a source of inspiration for others.
I believe in my abilities and potential.
My heart is open to love, kindness, and compassion.
I am at peace with my past and excited for the future.

St 1, Süt 1

Pozitif set

Today feels like a never-ending struggle.
Nothing seems to go right no matter how hard I try.
I am constantly surrounded by disappointment and frustration.
The world seems bleak and devoid of joy.
Every step forward is met with two steps backward.
My efforts are always met with failure.
I feel overwhelmed by the challenges in my life.
Negativity and doubt consume my thoughts.
It seems like happiness is always just out of reach.
I can't shake off this feeling of sadness and despair.
Everything I touch turns to dust.
The future looks grim and uncertain.
My dreams seem unattainable and distant.
I am trapped in a cycle of negativity.
The world appears to be full of hostility and conflict.
I'm constantly battling inner demons and self-doubt.
Each day is filled with disappointment and regret.
I can't escape the darkness that surrounds me.
Failure follows me like a shadow.
I'm drowning in a sea of problems with no lifeline in sight.
The weight of my problems is too much to bear.
It feels like I'm losing the battle against life.
I'm stuck in a never-ending loop of negativity.
Hopelessness clouds every aspect of my existence.
I'm haunted by past mistakes that I can't undo.
The world seems to be against me at every turn.

St 1, Süt 1

Negatif set

Bu işlemden sonra metinleri okuma fonksiyonunu c'de yazmam gerekti.

Fonksiyonun pozitif dosya için çalıştırılmış halinin deneme amaçlı print yapılması:

```
12 int main()
13 {
14     char class1[] = "positive.txt";
15     char class2[] = "negative.txt";
16     char *cumleler_1[MAX_CUMLE_SAYISI];
17     char *cumleler_2[MAX_CUMLE_SAYISI];
18     int cumleSayisi_1 = 0, cumleSayisi_2 = 0;
19     int i = 0;
20     int j = 0;
21     int uniqueKelimeSayisi = 0;
22
23     cumleSayisi_1 = dosyadanCumleleriOku(class1, cumleler_1, MAX_CUMLE_SAYISI);
24     cumleSayisi_2 = dosyadanCumleleriOku(class2, cumleler_2, MAX_CUMLE_SAYISI);
25
26     for (i = 0; i < cumleSayisi_1; i++)
27     {
28         printf("%s\n", cumleler_1[i]);
29     }
30 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
> cd "c:\Users\OMER\Desktop\dif\" ; if ($?) { gcc corsa.c -o corsa.c }
Today is a beautiful day filled with endless possibilities.
I am surrounded by love and positivity.
Success and happiness are my constant companions.
I radiate confidence and attract success.
Every challenge I face is an opportunity to grow and improve.
I am grateful for the abundance in my life.
Positivity flows through me like a refreshing breeze.
I am capable of achieving anything I set my mind to.
My life is a reflection of the positive energy I put into it.
Good things are constantly manifesting in my life.
I am a magnet for joy and good fortune.
I am in control of my own happiness.
The universe is conspiring to bring me success.
I am surrounded by supportive and uplifting people.
I am a beacon of positivity, inspiring those around me.
```

Bundan sonraki aşamada iki dosyadan da alınan metnin kelimelerine ayrılıp, unique elemanlar içerecek şekilde bir dizide saklanması geliyordu. Bu işlemi yapan fonksiyonun döndürdüğü dizinin ilk elemanlarını yorumlamak amaçlı ekliyorum.

```
25
26
27 //
28 char **uniqueKelimeler = uniqueKelimeleriBul(cumleler_1, cumleSayisi_1, cumleler_2, cumleSayisi_2, &uniqueKelimeSayisi);
29
29 printf("Unique kelimeler:\n");
30 for (i = 0; i < uniqueKelimeSayisi; i++)
31 {
32     printf("%d. kelime: %s\n", i + 1, uniqueKelimeler[i]);
33 }
34
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
> cd "c:\Users\OMER\Desktop\dif\" ; if ($?) { gcc corsa.c -o corsa } ; if ($?) { .\corsa }

Unique kelimeler:
1. kelime: Today
2. kelime: is
3. kelime: a
4. kelime: beautiful
5. kelime: day
6. kelime: filled
7. kelime: with
8. kelime: endless
9. kelime: possibilities
10. kelime: I
11. kelime: am
12. kelime: surrounded
13. kelime: by
14. kelime: love
15. kelime: and
16. kelime: positivity
17. kelime: Success
18. kelime: happiness
19. kelime: are
20. kelime: my
21. kelime: constant
```

Çıktıda da görüldüğü üzere ilk pozitifleri eklediğim için ilk cümlenin tüm kelimeleri en başta yer almış. Çünkü düşününce en başta birinci cümleden başlayarak diziyi inşa edeceği için 1. cümlenin sırayla diziden bakıldığında inşaa edilmesi gerekli zaten. Onun dışında çıktıda her kelimenin unique olduğunu görüyoruz, ayrıca kelimelerin sonunda noktalama varsa onları da kaldırıp diziyi öyle ekledim.

[illegible]

Yani `int onehot [toplam_veri_adeti][unique_kelime_sayısı]` gibi temsil edebileceğimiz bir matrise sahibiz.

Bu aşamadan sonra eğitim ve test verisi olarak ikiye bölmek gerekiyordu. Bu işlem aşağıdaki işe yarar:

Eğitim seti, modelin parametrelerini öğrenmesi için kullanılırken, test seti, modelin eğitim seti dışındaki verilere ne kadar iyi uyarlandığını ölçmek için kullanılır. Bu şekilde, modelin aşırı uyuma (overfitting) eğilimini değerlendirip önlemek ve gerçek dünya performansını daha doğru bir şekilde tahmin etmek mümkün olur.

2-) KODLAR

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#define MAX_CUMLE_UZUNLUK 300
```

```
#define MAX_CUMLE_SAYISI 200
```

```
int dosyadanCumleleriOku(char *dosyaAdi, char *cumleler[], int maxCumleSayisi);
```

```
char **uniqueKelimeleriBul(char *cumleler1[], int cumleSayisi1, char *cumleler2[], int  
cumleSayisi2, int *uniqueKelimeSayisi);
```

```
int **oneHotVectorOlustur(char *cumleler1[], char *cumleler2[], int cumleSayisi1, int cumleSayisi2,  
char **uniqueKelimeler, int uniqueKelimeSayisi);
```

```
int main()
```

```
{
```

```
    char class1[] = "positive.txt";
```

```
    char class2[] = "negative.txt";
```

```
    char *cumleler_1[MAX_CUMLE_SAYISI];
```

```
    char *cumleler_2[MAX_CUMLE_SAYISI];
```

```
    int cumleSayisi_1 = 0, cumleSayisi_2 = 0;
```

```
    int i = 0;
```

```
    int j = 0;
```

```
    int uniqueKelimeSayisi = 0;
```

```
    cumleSayisi_1 = dosyadanCumleleriOku(class1, cumleler_1, MAX_CUMLE_SAYISI);
```

```
    cumleSayisi_2 = dosyadanCumleleriOku(class2, cumleler_2, MAX_CUMLE_SAYISI);
```

```
    char **uniqueKelimeler = uniqueKelimeleriBul(cumleler_1, cumleSayisi_1, cumleler_2,  
cumleSayisi_2, &uniqueKelimeSayisi);
```

```

/*printf("Unique kelimeler:\n");
for (i = 0; i < uniqueKelimeSayisi; i++)
{
    printf("%s\n", uniqueKelimeler[i]);
}*/

/* printf("unique kelime sayisi: %d", uniqueKelimeSayisi);
for (i = 0; i < cumleSayisi_1; i++)
{
    printf("%s\n", cumleler_1[i]);

}*/

//

int **oneHotVektorler = oneHotVectorOlustur(cumleler_1, cumleler_2, cumleSayisi_1,
cumleSayisi_2, uniqueKelimeler, uniqueKelimeSayisi);

printf("One hot vektorler:\n");
for (i = 0; i < cumleSayisi_1 + cumleSayisi_2; i++)
{
    printf("%d. cumle icin one hot vektoru:\n ", i + 1);

    for (j = 0; j < uniqueKelimeSayisi; j++)
    {

        printf("%d ", oneHotVektorler[i][j]);
    }
    printf("\n\n");
}

// Bellekte ayrılana yerleri temizleyen kdolar

```



```

    for (i = 0; i < cumleSayisi_1; i++)
    {
        free(cumleler_1[i]);
    }
    for (i = 0; i < cumleSayisi_2; i++)
    {
        free(cumleler_2[i]);
    }

    for (i = 0; i < uniqueKelimeSayisi; i++)
    {
        free(uniqueKelimeler[i]);
    }
    free(uniqueKelimeler);

    for (i = 0; i < cumleSayisi_1 + cumleSayisi_2; i++)
    {
        free(oneHotVektorler[i]);
    }

    free(oneHotVektorler);

    return 0;
}

int dosyadanCumleleriOku(char *dosyaAdi, char *cumleler[], int maxCumleSayisi)
{
    FILE *dosya;
    char cumle[MAX_CUMLE_UZUNLUK];
    int cumleSayisi = 0;

```

```

dosya = fopen(dosyaAdi, "r");
if (dosya == NULL)
{
    perror("hata,dosya acilamadi");
    exit(EXIT_FAILURE);
}

// Dosyadan cumleleri satir satir okur ve diziye atar
while (fgets(cumle, MAX_CUMLE_UZUNLUK, dosya) != NULL && cumleSayisi < maxCumleSayisi)
{

    cumle[strcspn(cumle, "\n")] = '\0';
    cumleler[cumleSayisi] = strdup(cumle);
    cumleSayisi++;
}

fclose(dosya);

return cumleSayisi;
}

char **uniqueKelimeleriBul(char *cumleler1[], int cumleSayisi1, char *cumleler2[], int
cumleSayisi2, int *uniqueKelimeSayisi)
{
    char **uniqueKelimeler = NULL;
    int toplamUniqueKelimeSayisi = 0;
    int i, j;

    for (i = 0; i < cumleSayisi1; i++)
    {
        char *cumleKopya1 = strdup(cumleler1[i]);
        char *kelime = strtok(cumleKopya1, " ");
    }

```

```
while (kelime != NULL)
{

    // kelimenin sonunda noktalama varsa kaldırır
    char *sonKarakter = strchr(kelime, '\0') - 1;
    if (ispunct(*sonKarakter))
    {
        *sonKarakter = '\0';
    }

    // kelime unique dizinde var mı kontrol eder
    int unique = 1;
    for (j = 0; j < toplamUniqueKelimeSayisi; j++)
    {
        if (strcmp(kelime, uniqueKelimeler[j]) == 0)
        {
            unique = 0;
            break;
        }
    }

    // eger kelime unique diziside yoksa yani unique=1 kaldıysa, diziye ekler
    if (unique)
    {
        toplamUniqueKelimeSayisi++;
        uniqueKelimeler = realloc(uniqueKelimeler, toplamUniqueKelimeSayisi * sizeof(char *));
        uniqueKelimeler[toplamUniqueKelimeSayisi - 1] = strdup(kelime);
    }

    // obur kelimeyi al
    kelime = strtok(NULL, " ");
}
```

```
}  
}
```

```
// aynı işlemlerin 2. dosya için yapılması
```

```
for (i = 0; i < cumleSayisi2; i++)
```

```
{
```

```
    char *cumleKopya2 = strdup(cumleler2[i]);
```

```
    char *kelime = strtok(cumleKopya2, " ");
```

```
    while (kelime != NULL)
```

```
{
```

```
    char *sonKarakter = strchr(kelime, '\\0') - 1;
```

```
    if (ispunct(*sonKarakter))
```

```
{
```

```
        *sonKarakter = '\\0';
```

```
}
```

```
    int unique = 1;
```

```
    for (j = 0; j < toplamUniqueKelimeSayisi; j++)
```

```
{
```

```
        if (strcmp(kelime, uniqueKelimeler[j]) == 0)
```

```
{
```

```
            unique = 0;
```

```
            break;
```

```
}
```

```
}
```

```
    if (unique)
```

```
{
```

```
        toplamUniqueKelimeSayisi++;
```

```
        uniqueKelimeler = realloc(uniqueKelimeler, toplamUniqueKelimeSayisi * sizeof(char *));
```

```

        uniqueKelimeler[toplamUniqueKelimeSayisi - 1] = strdup(kelime);
    }

    kelime = strtok(NULL, " ");
}
}

// yollanan pointera unique kelime sayısını yazıyor
*uniqueKelimeSayisi = toplamUniqueKelimeSayisi;

return uniqueKelimeler;
}

// iki dosya için one hot vektor olusturur
int **oneHotVectorOlustur(char *cumleler1[], char *cumleler2[], int cumleSayisi1, int cumleSayisi2,
char **uniqueKelimeler, int uniqueKelimeSayisi)
{
    int **oneHotVektorler = NULL;

    int i, j;

    // Her cümle için one hot vektör
    for (i = 0; i < cumleSayisi1; i++)
    {
        // dinamik bellek alanı ayır ve sıfır doldurur
        oneHotVektorler = realloc(oneHotVektorler, (i + 1) * sizeof(int *));
        oneHotVektorler[i] = calloc(uniqueKelimeSayisi, sizeof(int));

        // kelimelere ayıran kısım
        char *cumleKopya = strdup(cumleler1[i]);
        char *kelime = strtok(cumleKopya, " ");
        while (kelime != NULL)

```

```

{
    // printf("%d %s ", i, kelime);

    char *sonKarakter = strchr(kelime, '\0') - 1;
    if (ispunct(*sonKarakter))
    {
        *sonKarakter = '\0';
    }

    // Kelimeninin unique listesindeki gectigi indexi bul
    int kelimeIndeksi = -1;
    for (j = 0; j < uniqueKelimeSayisi; j++)
    {
        if (strcmp(kelime, uniqueKelimeler[j]) == 0)
        {
            // printf("%d :%s %s\n", i, kelime, uniqueKelimeler[j]);
            kelimeIndeksi = j;
            break;
        }
    }

    // kelimeyi one hot vektorunde isaretle
    if (kelimeIndeksi != -1)
    {
        oneHotVektorler[i][kelimeIndeksi] = 1;
    }

    // obur kelimeye gec
    kelime = strtok(NULL, " ");
}
}

```

```
// 2. dosya icin aynı işlemler ,
// x degiskeni kullandim cunku aynı one hot vektor dizisine eklemek için
// i degiskeninin devamına eklemek gerekiyor
int x = 0;

for (x = 0; x < cumleSayisi2; x++)
{
    oneHotVektorler = realloc(oneHotVektorler, (x + i + 1) * sizeof(int *));
    oneHotVektorler[x + i] = calloc(uniqueKelimeSayisi, sizeof(int));

    char *cumleKopya = strdup(cumleler2[x]);
    char *kelime = strtok(cumleKopya, " ");
    while (kelime != NULL)
    {

        char *sonKarakter = strchr(kelime, '\\0') - 1;
        if (ispunct(*sonKarakter))
        {
            *sonKarakter = '\\0';
        }

        int kelimeIndeksi = -1;
        for (j = 0; j < uniqueKelimeSayisi; j++)
        {
            if (strcmp(kelime, uniqueKelimeler[j]) == 0)
            {
                kelimeIndeksi = j;
                break;
            }
        }
    }
}
```

```
    if (kelimeIndeksi != -1)
    {
        oneHotVektorler[x + i][kelimeIndeksi] = 1;
    }

    kelime = strtok(NULL, " ");
}

return oneHotVektorler;
}
```

3-) Video Adresi

<https://youtu.be/i17CnL8gk9c>