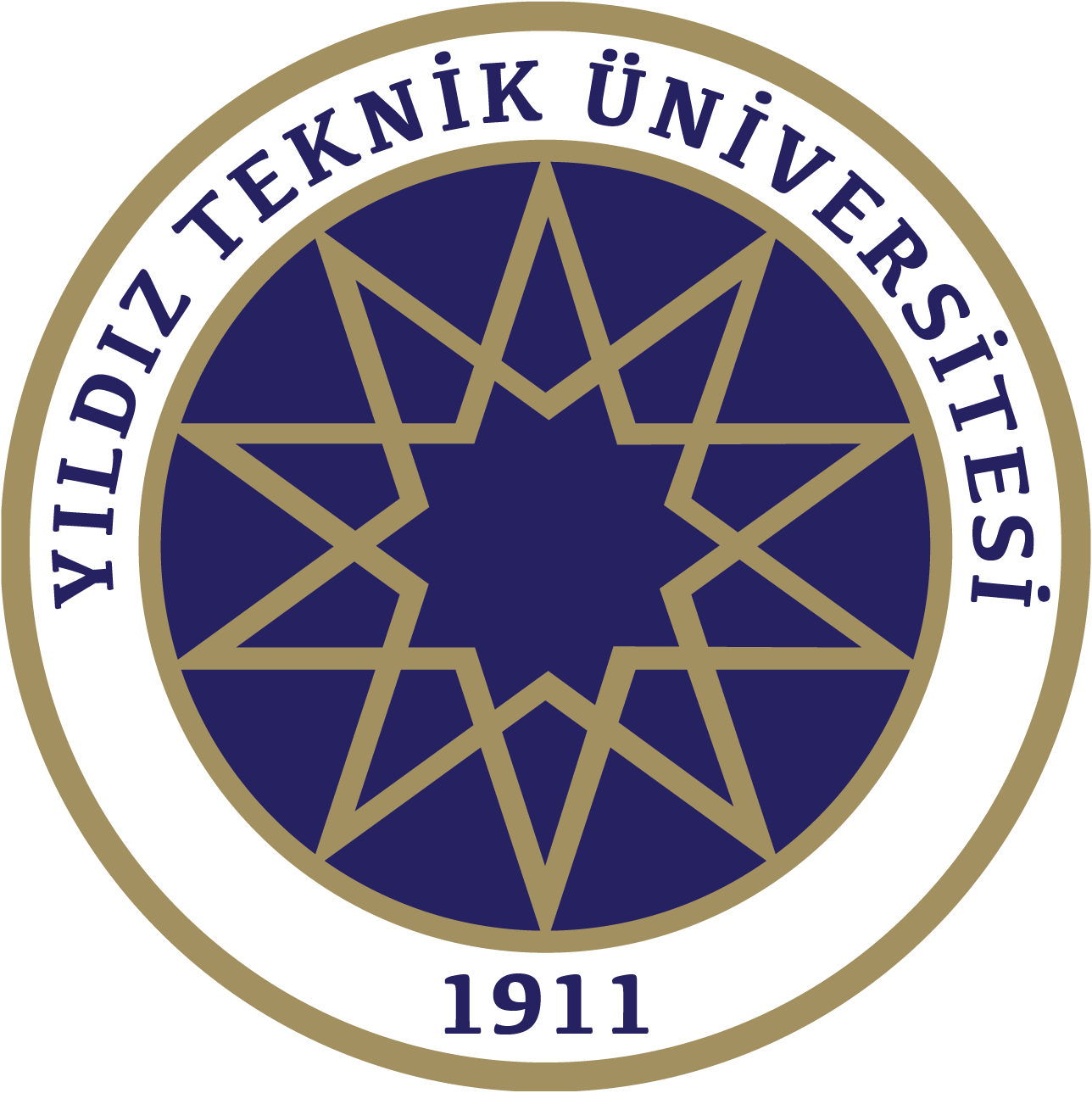
**BLM2642 ÖDEV**



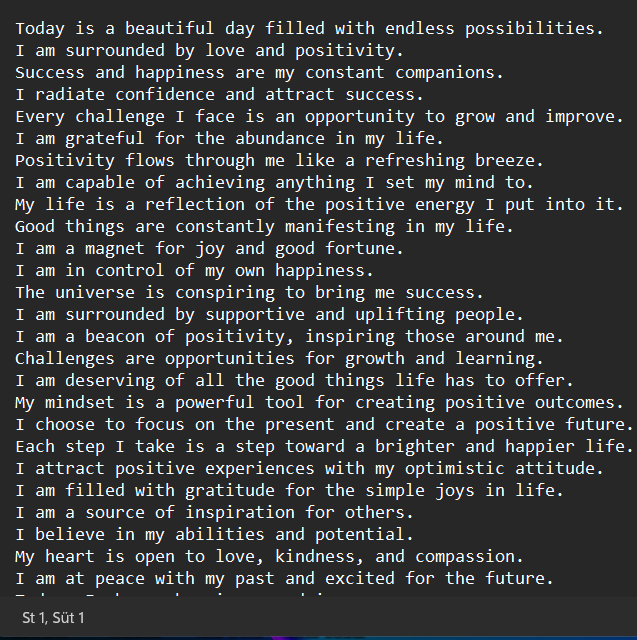
**HAZIRLAYAN**

ÖMER DİNER 20011017

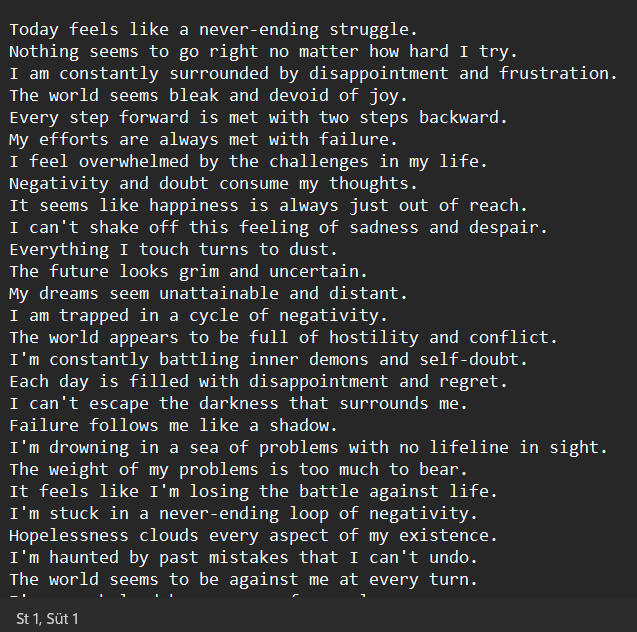
[omer.diner@std.yildiz.edu.tr](mailto:omer.diner@std.yildiz.edu.tr)

**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;



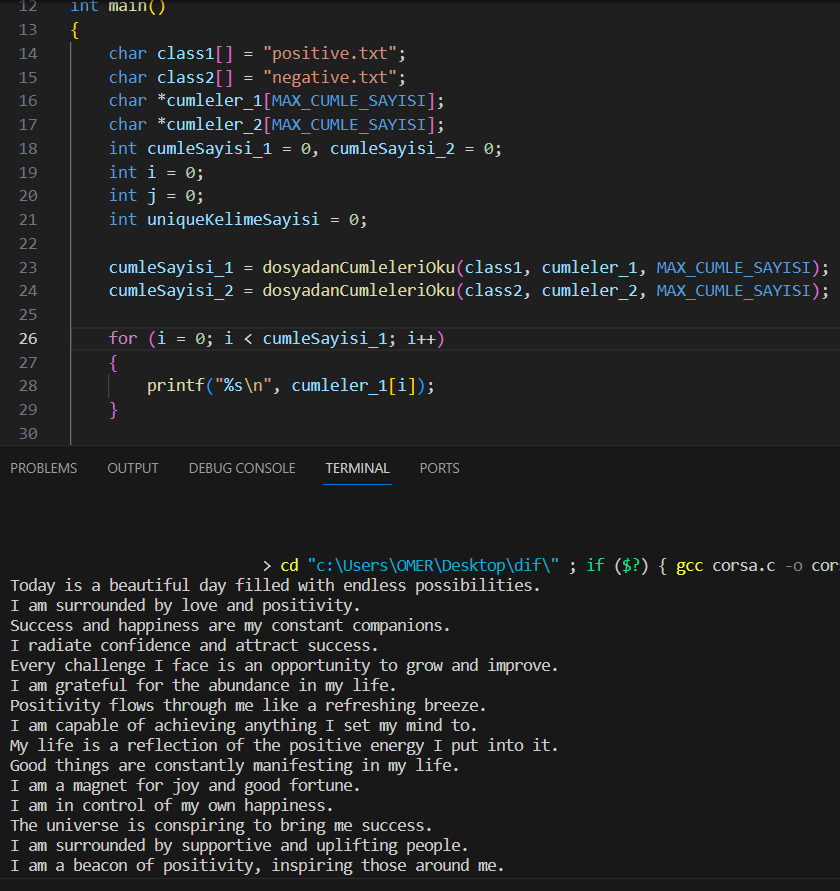
Pozitif set



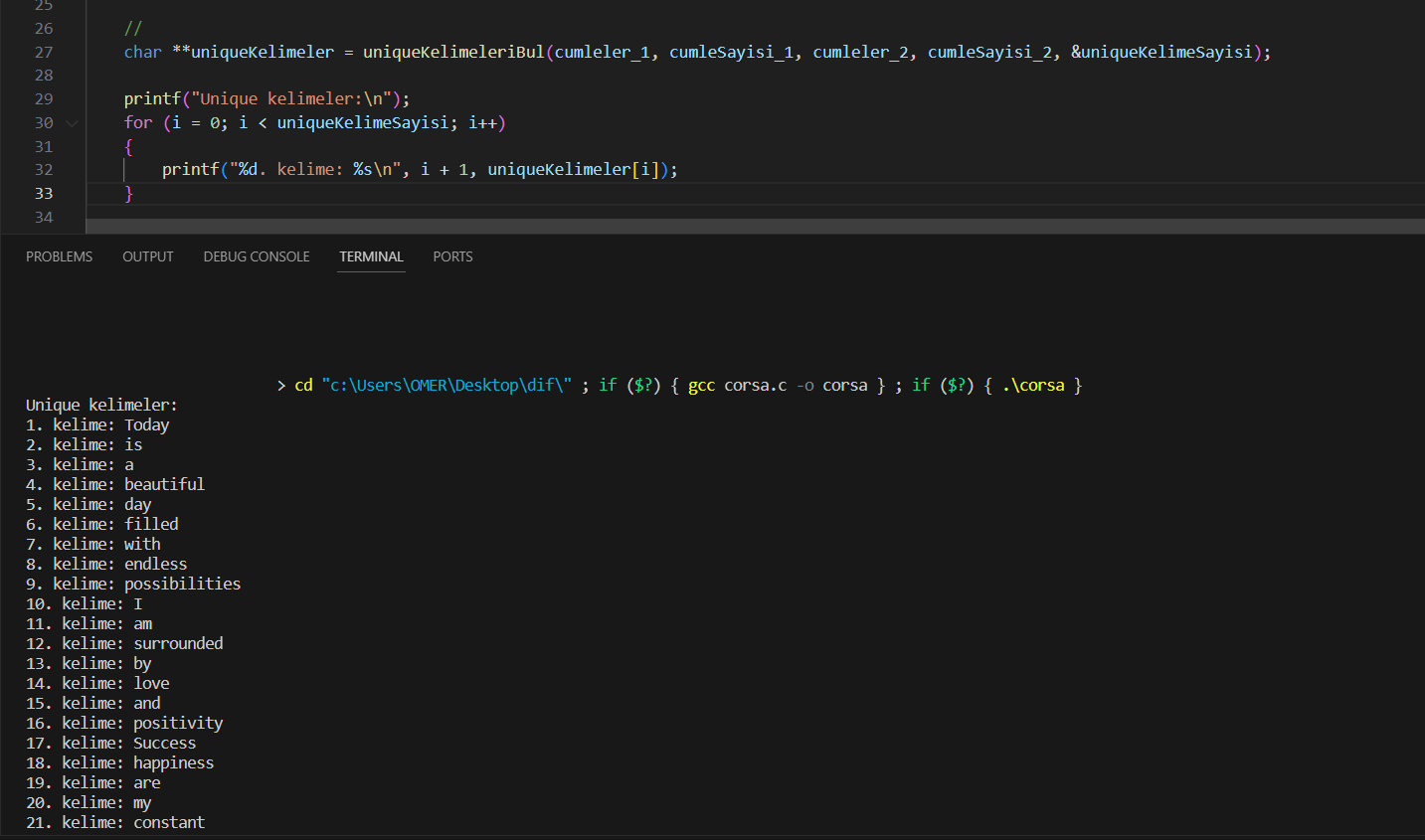
Negatif set

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

Fonksiyonun pozitif dosya için çalıştırılmış halinin deneme amaçlı print yapılması:



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.



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 diziye öyle ekledim.

Sırada bu unique kelimeler dizisini kullanarak her veriyi ,hot vektör dizisi haline getirmek var.Bu fonksiyon sonrasında iki boyutlu bir int dizisi elimizde olacak. Ve vektörde bulunan 1’ler bulunduğu indisin unique dizisinde karşılık geldiği kelimenin , hot vektorunu oluşturduğumuz verinin de içinde bulunduğunu temsil ediyor. Örnek vermek gerekirse;



Toplam veri sayısı kadar one hot vector var , ve her birinin uzunluğu da unique kelime sayısı kadar.

Yani int onehot [toplam\_veri\_adeti][unique\_kelime\_sayısı] gibi temsil edebileceğimiz bir matrise sahibiz.

Burada ilk cümle için çıkan sonuçta ilk indislerin 1 olması tesadüf değil çünkü 9 adet 1 var ve ilk cümlemiz de 9 kelimeden oluşuyordu zaten. Yani her kelime için unique dizisindeki karşılık geldiği yerde 1 işaretlenmiş. Bu unique dizisini oluşturmaya pozitiflerden başlanıldığı için 1.pozitif cümle en baştan yerleşmiş oldu. Sonrasında veriler arttıkça içindeki kelimeler hangi indislerde varsa o indislerde 1 olarak gözükecek. En baştaki gibi sıralı bir 1 görüntüsü elde etmek düşük bir ihtimal olacak onlar için.

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 islemlerin 2. dosya icin 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 icin 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 ksıım**

**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 islemler ,**

**// x degiskeni kullandim cunku ayni one hot vektor dizisine eklemek icin**

**// 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**