**GRAF TABANLI METİN ÖZETLEME PROJESİ**

Berk Sunduri

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

berksunduri@gmail.com

**1-)Projenin Tanımı**

Bu kısım sadece projenin açıklamasını okuyup edindiğim ön bilgiye göre yazılmıştır.

Bize verilen pdf dosyasında projenin amacını öğrendim. Verilen isterleri dikkatlice okudum

Bu proje, grafik tabanlı bir sıralama algoritması kullanan bir Otomatik Metin Özetleme Uygulaması geliştirmeyi amaçlamaktadır.

**1.1-)Problem Tanımı**

Bu kısımda bizden yapmamız istenilenler içermektedir.

Birinci adımda bizden bir masaüstü arayüzü geliştirmemiz ve graf yapısının oluşturulması istendi.

İkinci ister olarak cümleler arası anlamsal ilişki kurmamız istendi.

Üçüncü aşamada ise bizden cümle skoru hesaplamaya yarayan bir algoritma geliştirmemiz istendi.

Dördüncü ister olarak bizden skorlara göre metin özetleme algoritması geliştirmemiz istendi

Beşinci ister olarak ise özetleme başarısının ROUGE skoru ile hesaplamamız istendi.

**2-)Yapılan Araştırmalar ve Karşılaşılan Sıkıntılar**

Bu kısım proje öncesi ve sonrası araştırmaları ve de projenin yapım aşamasındaki sıkıntıları ve çözümlerini içermektedir.

İlk karşılaştığım sorun hangi programlama dilini kullanacağım olduğuydu. IDE olarak PyCharm, programlama dili olarakta Python dilini seçtim.

Daha sonrasında pythonda ilk defa bir text summarizer uygulaması yazıcağım için internette uzun süre araştırma yapmak zorunda kaldım ki bu beni baya zorladı. Bunun için stackoverflow’dan yardım aldım.  
  
Diğer bir zorluksa graf oluşturmak oldu. Tam olarak örnek bir graf verilmediği için grafımın neye benzeyeceğini bilmek zor oldu. Bunun için internette uzun süre araştırma yaptım.

**2.1-Proje Sırasında Yararlanılan Teknolojiler**

Projeyi Python dili kullanarak PyCharm IDE’sinde yazdım.

Programı yaparken Python’un birkaç kütüphanesinden yararlandım.

**3-)Tasarım**

**3.1-Akış Diyagramı**

Kısım ektedir.(1)

**4-)Genel Yapı**

**4.1-Kullanıcı Kısmı**

Program başladığında karşımıza text yükleyebileceğimiz bir ekran çıkmaktadır.

Kullanıcılar, yazı içeren herhangi bir .txt dosyasını yükleyebilirler. Program bundan sonra otomatik olarak kullanıcının önüne girilen .txt dosyasının özetini çıkarır.

Kullanıcı eğer isterse bu verilen özeti farklı bir özetle karşılaştırabilir, girilen tekstin grafını oluşturabilir ve iki tekst arasında ki ROUGE skorunu hesaplayabilir.

**4.2-Kod Kısmı**

Kod bir Tkinter sınıfı içerisinde çalışmaktadır. Bu sınıf ilk başlatıldığında direkt program için gerekli butonlar ve textviewları oluşturur.

Bunun sonrasında dosya yükleme işlemleri yapılır. Dosya yükleme işlemi sırasında dosya açılıp okunduktan hemen sonra tokenlenir ve cümle yerleştirmeleri oluşturulur.

Hemen sonra bir graph oluşturulur ve her cümle için bir node eklenir. Sonrasında her node için kosinüs benzerliğine bakılarak birer bağlantı eklenir.

Bunun sonrasında cümle skorları hesaplanır ve nodelar ilgili puanlarıyla güncellenir.

Ancak ve ancak bu işlemler bittikten sonra cümlenin özeti çıkarılabilir. İki cümle arasında ki anlamsal ilişkiyi kırmak için BERT modeli kullanılmıştır. Cümle skoru hesaplama algoritması geliştirilirken verilen isterler dikkatlice okunup koda dökülmüştür. TF-IDF değerinin hesaplanması Pythonun bir kütüphanesi olan sklearn kütüphanesi yardımıyla gerçekleştirilmiştir.

**5-)Kütüphaneler**

Bu kısımda projeye include ettiğim kütüphaneler bulunmakta:

1. tkinter
2. nltk
3. matplotlib
4. transformers
5. sklearn
6. networkx
7. re
8. numpy
9. rouge

**6-)Referanslar**

1-)SciKit Learn

”https://scikit-learn.org/stable/”

2-) NLTK

”https://www.nltk.org/ “

3-) Stack Over Flow ”[https://www.stackoverflow.com](https://www.stackoverflow.com/)”

4-) BertModel and Transformers

“https://huggingface.co/docs/transformers/index”