Yazılım Laboratuvarı-2 Dersinin Üçüncü Projesi
Metin Özetleme

Zeynep Duvarcı (210201113), Mehtap Özbunar (200201095)
Bilgisayar Mühendisliği
Kocaeli Üniversitesi

Özet

Bu rapor Yazılım Laboratuvarı 2 dersinin 3. Projesini açıklamak ve sunumunu gerçekleştirmek için oluşturulmuştur. Raporda projenin içeriği, yöntemi, deneysel, sonuç kısımları bulunmaktadır. Ek olarak kaynakçayı da eklemiş bulunmaktayız. Projenin gereklilikleri, kullanıcının seçtiği dosyadan okunan metni graf kullanarak görselleştirme, bu metni cümle seçme yöntemiyle özetleme ve özetlenen metnin gerçek metin özetine göre ROUGE skorunun hesaplanması işlemlerinin masaüstü uygalaması olarak gerçekleştirilmesidir.

Giriş

PyCharm Community Edition idesiyle Python dilinde bir masaüstü uygulaması geliştirdik. Uygulamayı, graf tabanlı metin özetlemek ve oluşturduğumuz graf ve özeti görselleştirmek amacıyla geliştirdik. Temel olarak, kullanıcının seçtiği doküman ve girdiği parametre değerlerini kullanarak, cümle seçerek metin özetleme yöntemiyle dokümandaki metni özetledik. Ayrıca dokümandaki metni, cümlelere ayırarak her cümleyi bir düğüm ile temsil edecek şekilde, düğümlere ve kenarlara farklı özellikler için hesapladığımız değerleri ekleyerek bir graf oluşturduk. Son olarak, özet metni, grafı ve özet metnin gerçek özetle benzerliği için hesapladığımız ROUGE skorunu ekrana yazdırarak projemizi tamamladık.

Yöntem

Öncelikle, *tkinter* kütüphanesini kullanarak doküman yüklenebilecek, parametreler
seçilebilecek, ekranda graf ve özet gösterilebilecek şekilde masaüstü uygulamamızın arayüzünü tasarladık.

Dokümanın seçilebilmesini sağlamak için tkinter'in filedialog modülünü kullandık.

Diğer parametrelerin girilmesi ve seçilmesini sağlamak için de yine tkinter kütüphanesinin *OptionMenu* ve *Entry* modüllerini kullandık.

Kullanıcıdan aldığımız parametreler şunlardır:
Cümle Benzerliği Thresholdu
Cümle Skoru Thresholdu
Cümle Benzerliği Algoritması

Tüm seçimler yapıldığında, seçimlerin okunması, özetin çıkarılması, grafın oluşturulması gibi işlemlerin başlatılmasını sağlayan bir *Button* (tkinter) ekledik.

Butona basıldığında gerçekleşecek işlemler ise şu şekildedir:
İlk olarak seçilen dokümandan metni okuduk ve cümlelere ayırdık. *nltk* kütüphanesi modüllerini kullanarak bu cümlelere tokenization, stemming, stop-word elimination, punctuation ön işleme adımlarını uyguladık.
Daha sonra Networkx kütüphanesini kullanarak her cümle bir graf düğümünü temsil edecek şekilde bir graf oluşturduk. Grafın kenar özellikleri olarak, seçilen (Word Embedding/BERT) algoritmayı kullanarak, ön işleme adımları uygulanmış cümleler için hesapladığımız cümle benzerliği değerlerini ekledik. Her düğüm için, kullanıcının girdiği cümle benzerliği threshold değerini geçen bağlı düğümlerin sayısını da düğüm özelliği olarak ekledik.
Sonrasında, cümle skoru hesaplamada kullanılacak olan aşağıdaki parametreleri hesapladık.
1.parametre: Cümledeki özel isim sayısı / Cümlenin uzunluğu
2.parametre: Cümledeki numerik veri sayısı / Cümlenin uzunluğu
3.parametre: Tresholdu geçen nodeların bağlantı sayısı / Toplam bağlantı sayısı
4.parametre: Cümledeki başlıkta geçen kelime sayısı / Cümlenin uzunluğu

5.parametre: Cümlenin içinde geçen tema kelime sayısı / Cümlenin uzunluğu

Bu parametreleri kullanarak her cümlenin skorunu aşağıdaki şekilde hesapladık.
Score = (1/2)*p1 + p2 + p3 + 2*p4 + 3*p5

Tüm skorları hesapladıktan sonra, skorları 0-1 arasına normalize ettik.
Grafteki her bir düğüme cümle skorlarını özellik olarak ekledik.
Cümle skorlarını büyükten küçüğe doğru sıralayarak, cümleri sırayla seçtik. Seçtiğimiz cümleleri birleştirerek özeti oluşturduk ve bu özetin gerçek özetle olan benzerliğini ROUGE skor ile hesapladık.

Son olarak oluşturduğumuz özeti, grafı, ve ROUGE skor değerini ekrana yazdırdık.
Ayrıca geliştirdiğimiz uygulamada, seçilen dosyanın ve parametrelerin değiştirilerek özetin ve grafın tekrar oluşturulabilmesini de sağladık.

Deneysel

Projede önce bize verilen örnek metin ve çıktıyı daha sonra paylaşılan veri setini uygulamamızda deneyimledik. Bize verilen metin başlık ve on bir cümle özet ise beş cümleden oluşuyordu.

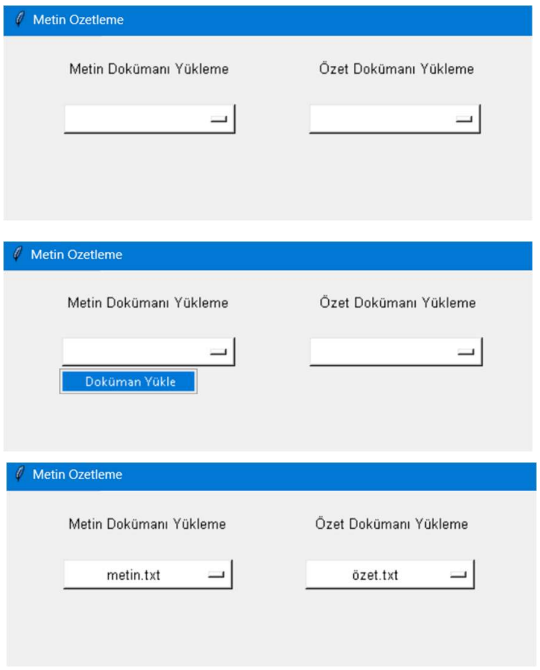
Örnek metnin bir kısmı :

```
Gallery unveils interactive tree

A Christmas tree that can receive text messages has been unveiled at London's Tate Britain art gallery.

The spruce has an antenna which can receive Bluetooth texts sent by visitors to the Tate. The messages will be "unwrapped" by sculptor Richard Wentworth, who is responsible for decorating the tree with broken plates and light bulbs. It is the 17th year that the gallery has invited an artist to dress their Christmas tree. Artists who have decorated the Tate tree in previous years include Tracey Emin in 2002.
```

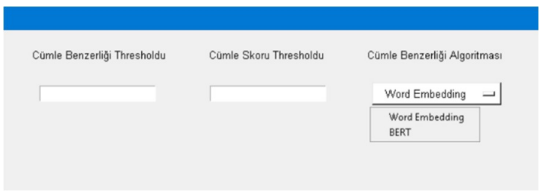
Örnek metnin beklenen özeti :

```
The messages will be "unwrapped" by sculptor Richard Wentworth, who is responsible for decorating the tree with broken plates and light bulbs.A Christmas tree that can receive text messages has been unveiled at London's Tate Britain art gallery.It is the 17th year that the gallery has invited an artist to dress their Christmas tree.The spruce has an antenna which can receive Bluetooth texts sent by visitors to the Tate.His reputation as a sculptor grew in the 1980s, while he has been one of the most influential teachers during the last two decades.
```

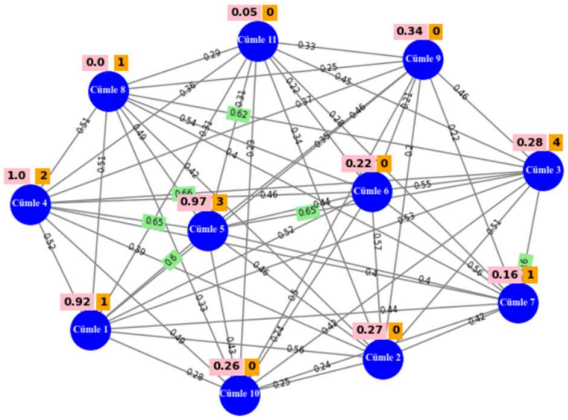
Uygulamamızda öncelikle kullanıcıdan her iki metninde txt formatındaki dosyaları seçmesini sağladık.



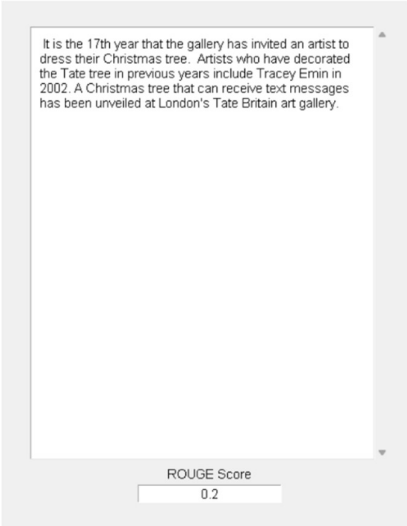
Kullanıcı dilediği txt formatındaki dosyaları bilgisayarından yükledikten sonra üç farklı parametreyi girmesini sağladık. Bunlar cümle benzerliği tresholdu, cümle skoru tresholdu ve cümle benzerliği algoritması. Son parametrede kullanıcıya sadece Word Embedding ve Bert modelini sunduk.



Kullanıcının her parametrenin seçimini yaptığını kontrol ettikten sonra SHOW GRAPH butonu sayesinde girmiş olduğu metni girilen parametreler sayesinde grafa çevirdik. Örneğin bize verilen örnek metni cümle benzerliği tresholdunu 0.6, cümle skoru tresholdunu 0.5 ve cümle benzerliği algoritmasını da Word embedding olarak belirlediğimizde grafı aşağıdaki gibi olur ve kullanıcıya her seferinde arayüzde oluşturduğumuz graf gösterilir.



Metne uygulanan işlem sonuçlandığında arayüzde uygulanan algoritma sonucu olan metin gösterilir ve rouge-1 skoruna da arayüzde gösterilir.



Sonuç

Kullanıcıya seçimleri sonucu girilen metni özetleyebileceği ve beklenen özetle karşılaştırabileceği bir arayüz sunduk. Cümleleri seçerek özet yapan uygulamamızda kullanıcının arka planda oluşan grafı da görmesini sağladık. Oluşan özet ve beklenen özet benzerliğini rouge skoru sayesinde kullanıcıya sunduk.

Kaynakça

Metin Özetleme Algoritması:

Erdağı, E. (2023). *Türkçe metinlerde çıkarım tabanlı otomatik metin özetleme* (Tez No. 779797) [Doktora Tezi, Maltepe Üniversitesi]. YÖK Tez Merkezi. https://tez.yok.gov.tr/UlusalTezMerkezi/

Güran, A. (2013). *Otomatik metin özetleme sistemi* (Tez No. 329658) [Doktora Tezi, Yıldız Teknik Üniversitesi]. YÖK Tez Merkezi. https://tez.yok.gov.tr/UlusalTezMerkezi/

Baydar, E. (2018). *Genetik algoritma kullanarak cümle seçme yaklaşımı ile otomatik metin özetleme* (Tez No. 509691) [Yüksek Lisans Tezi, Van Yüzüncü Yıl Üniversitesi]. YÖK Tez Merkezi. https://tez.yok.gov.tr/UlusalTezMerkezi/

Gümüş, M. (2019). *An evaluation of automatic text summarization techniques* (Tez No. 585926) [Master Tezi, Bahçeşehir Üniversitesi]. YÖK Tez Merkezi. https://tez.yok.gov.tr/UlusalTezMerkezi/

Yararlandığımız video kaynaklar:
Tkinter:
https://www.youtube.com/watch?v=8NLj2qq_qwU
https://www.youtube.com/watch?v=q8WDvrjPt0M
Networkx:
https://www.youtube.com/watch?v=XPaw5EwHQ8U
Dosyadan veri okuma:
https://www.youtube.com/watch?v=3RA0t7cIzfw
Cümle benzerliği hesaplama:
https://towardsdatascience.com/how-to-compute-sentence-similarity-using-bert-and-word2vec-ab0663a5d64
https://towardsdatascience.com/calculating-document-similarities-using-bert-and-other-models-b2c1a29c9630
Tf-idf:
https://stackoverflow.com/questions/34232190/scikit-learn-tfidfvectorizer-how-to-get-top-n-terms-with-highest-tf-idf-score
https://www.kaggle.com/code/rowhitswami/keywords-extraction-using-tf-idf-method

Sözde Kod

main.py

tkinter ile pencere oluştur.
Kullanılacak frame'leri tanımla.

delete_graph_and_summary fonksiyonu:
  Graf ve özet kısmını pencereden sil.

Butona basıldığında çalışacak fonksiyon olan button_submit_click fonksiyonu:
  delete_graph_and_summary fonksiyonunu çağır.
  Boş parametre kalıp kalmadığını ve girilen parametrelerin doğru türde olduğunu kontrol et.
  Seçilen dokümanlaradan örnek metin ve özeti oku.
  Örnek metini cümlelere ayır.
  Cümlelere ön işleme adımlarını uygula.
  Grafı oluştur.
  i 1'den metindeki cümle sayısı+1'e:
    j i'den metindeki cümle sayısı+1'e:
      Grafa Cümle i adında düğüm ekle.
      Cümleyi düğüme özellik olarak ekle.
      Seçilen benzerlik algoritmasına göre Cümle i ve Cümle j'nin benzerliğini hesapla.
      Cümle+i ve Cümleb j'ye kenar bağlantısı ekle. Benzerliği özellik olarak kenara ekle.
      Kullanıcıdan alınan cümle thresholdunu geçen düğüm sayısı düğüme özellik olarak eklenir.
    Cümle i için Score.py dosyasındaki fonksiyonlar kullanılarak tüm parametreleri hesapla.
    Parametrelere katsayı vererek Cümle i'nin skorunu hesapla ve listeye ekle.

  Cümle skorlarının bulunduğu listeyi 0-1 arasına normalize et.
  Normalize edilmiş skorları graftaki cümle düğümlerine ekle.
  Kullanıcıdan alınan cümle thresholdunu geçen cümleleri birleştir ve özeti oluştur.
  Oluşan özet ile beklenen özet arasındaki benzerliği ROUGE skoru ile hesapla.
  Grafın renk ve düzen ayarlamalarını yap.
  Buluna özeti ve ROUGE skorunu ekrana yazdır.
  Grafı ekrana çiz.

load_document ve load_document2 fonksiyonları:
  filedialog ile doküman seçilecek ekranı açtır.
  Seçilen dokümanın ismini dosya yolundan al.
  OptionMenu'ye yaz.

Uygulamadaki Label'ları oluştur.
Parametrelerin girileceği, dokümanın seçileceği giriş alanlarını oluştur.
Threshold değerleri için Entry, doküman ve cümle benzerliği algoritması seçimi için OptionMenu oluştur.
Doküman seçimi için oluşturulan OptionMenuler seçildiğinde load_document ve load_document2 fonksiyonlarını çalıştır.
Parametreler girilip, dokümanlar seçildikten özet çıkarma ve graf çizdirme işlemlerinin başlaması için Button oluştur.
Button'a tıklandığında button_submit_click fonksiyonunu çalıştır.

Score.py

Def p1
Cümlenin ayrılmış halini tagle
Propernouns ve possessives leri listeye ata
Sonucu ikisinin sayısı bölü cümle uzunluğu olarak döndür
Def p2
Cümlenin içinde sayı geçen kelimelerini bul
Sonucu sayı geçen kelime sayısı bölü cümle uzunluğu olarak döndür.
Def nltk_preprocessing
Cümleyi tokenize et
Her bir tokena stemming işlemi uygula
Stopwordleri sil
Cümlenin son halini gönder
Def word_embedding
Modelin kelime listesine ulaş
Karşılaştırılan her iki cümlenin de kelime listesinde olan kadarını listeye at
Her iki cümlenin de cosine similarity ile benzerliğini döndür.
Def get_top_words
TfidfVectorizer tanımlama ve cümle listesini vektörize etme
Kelime listesini al
Cümle listesindeki tf-idf değeri en yüksek olan değerleri %10 oranında ve liste olarak döndür.
Def p5
Get_top_words'ten alınan kelime listesini cümlede kaç tane olduğunu say
Sayıyı cümle uzunluğuna bölerek döndür
Def normalization
Oluşan skor listesini al
Skor listesindeki maksimum değerle minimum değer arasındaki farkı bul
Skor listesindeki elemanı minimum değere böldükten sonra aradaki farka böl ve listeye ekle
tuple listesine normalize hallerini ekle

Def rouge
Rouge skoru için evaluate kütüphanesiyle yükleme yap
Beklenen çıktı alınan çıktının uzunluğuna eşitse rouge skorunu hesapla
Beklenen çıktı alınan çıktının uzunluğundan farklıysa küçük olana göre sınırla ve rouge skorunu döndür.