

YAZILIM LABORATUVARI 2

PROJE 3

TARIK GÖREN
Bilgisayar Mühendisliği
200202022@kocaeli.edu.tr

FURKAN ENES DAĞLI
Bilgisayar Mühendisliği
200202012@kocaeli.edu.tr

MUSTAFA KAHVECİ
Bilgisayar Mühendisliği
200202036@kocaeli.edu.tr

ÖZET

Bu doküman Yazılım Laboratuvarı 2 dersi 3. Projesi için hazırlanmıştır. Dokümanda projenin tanımı, çözüm yöntemi, kullanılan kütüphaneler gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Doküman sonunda projemizi hazırlarken yararlandığımız kaynaklar bulunmaktadır.

I. GİRİŞ

Bu projede verilen bir dokümandaki cümlelerin graf yapısına dönüştürülmesi ve bu graf modelinin görselleştirilmesi istenmektedir. Ardından graf üzerindeki düğümler ile özet oluşturan bir algoritma oluşturulması beklenmektedir.

Projede temel amaç cümleleri graf yapısına çevirip Cümle Seçerek Özetleme (Extractive Summarization) gerçekleştirmektir. Graf yapısına çevirerek cümlelerin metindeki anlamsal ilişkilerini görselleştirmek ve bu ilişkileri kullanarak önemli cümleleri belirlemek amaçlanmaktadır.

Bu proje kapsamında bizden aşağıdaki isterlerin gerçekleştirilmesi istenmiştir:

- 1) Projede masaüstü uygulama geliştirmeniz gerekmektedir
- 2) Masaüstü uygulamada ilk olarak doküman yükleme işlemi gerçekleştirilecektir.
- 3) Ardından yüklenen dokümandaki cümleleri graf yapısı haline getirmeniz ve bu graf yapısını görselleştirmeniz beklenmektedir. sağlanmalıdır. Bu grafta her bir cümle bir düğümü temsil edecektir.

4) Cümleler arasındaki anlamsal ilişki kurulmalı, cümleler skorlanmalıdır. Belirli parametreleri kullanarak cümle skorunun hesaplama algoritmasını ve cümle skorlarına göre metin özeti çıkarma algoritmalarını sizin geliştirmeniz istenmektedir.

5) Özet metni arayüzde sunmanız beklenmektedir. Sonuç olarak size verilen bir metnin özetini bu yöntem ile çıkarmanız ve gerçek özet ile benzerliğini “ROUGE” skorlaması ile ölçmeniz istenmektedir.

II. YÖNTEM

A. Algoritma

Öncelikle projeyi hangi programlama dili, veritabanı ve frameworkleri kullanarak gerçekleştireceğimizin değerlendirmesini yaptık. Bu değerlendirme sonucunda masaüstü uygulama geliştirme kısmında ElectronJs kütüphanesini ve Angular frameworkünü, graph veritabanı olarak Neo4j, backend tarafında Spring Boot (JAVA) ve Flask (Python) frameworklerini kullanmaya karar verdik.

Bizden istenen ilk isterden - Projede masaüstü uygulama geliştirmeniz gerekmektedir - başladık. Bir Angular projesi oluşturup bu projeye ElectronJs implementasyonlarını gerçekleştirip projemizi bir masaüstü uygulama olarak ayağa kaldırdık.

Sonra ikinci isterde belirtilen kullanıcıdan dokümanın alınması ve okunması işlemini gerçekleştirdik ve girilen doküman inputundan başlık ve metin kısımlarını ayırıp bir json objesinde tuttuk.

Daha sonra üçüncü isteri gerçekleştirmeye başladık. Kullanıcıdan aldığımız metni cümlelere ayırıp Neo4j veritabanına her bir cümle bir node'u temsil edecek şekilde kayıt ettik. Aynı zamanda

masaüstü uygulamamızda da Vis.js kütüphanesini entegre ederek metnin graph görüntüsünü kullanıcı arayüzüne de sunduk.

Daha sonra dördüncü isteri gerçekleştirmeye başladık. Cümleler arasında anlamsal ilişki kurma ve cümlelerin skorlanması kısmında bize sunulan 2 algoritmayı - Word Embedding ve BERT - araştırıp değerlendirdik. Değerlendirmemiz sonucunda Word Embedding algoritmasını seçerek GloVe modelini kullanmaya karar verdik.

GloVe, kelime vektörlerini oluşturmak için istatistiksel bilgileri kullanır. Bu algoritma, büyük metin veri setleri üzerinde eğitilir ve kelime öbekleri arasındaki istatistiksel ilişkileri yakalar. GloVe, kelime çiftlerinin birbirleriyle orantılı olduğu bir matris faktörizasyonu yöntemi kullanır. Bu sayede, bir kelimenin diğer kelimelerle olan ilişkilerini ve anlamsal benzerliklerini temsil edebilir. Örneğin, "kral" kelimesi ile "kraliçe" kelimesi arasındaki ilişkiyi GloVe kullanarak bulabilir ve bu ilişkiyi matematiksel olarak ifade edebiliriz.

GloVe modeli ile bulduğumuz benzerlik sonuçlarını Neo4j veritabanımıza kaydettik. Sonrasında bu benzerlikleri cümle skoru hesaplama algoritmasında bulunan diğer 4 parametrenin (Cümle özel isim kontrolü, Cümlede numerik veri olup olmadığının kontrolü, Cümle benzerliği threshold'unu geçen node'ların bulunması, Cümlede başlıktaki kelimelerin olup olmadığının kontrolü ve Her kelimenin TF-IDF değerinin hesaplanması) bulunması ile cümle skoru hesaplama algoritmamızı da tamamladık. Bulmuş olduğumuz tüm skorları Neo4j veritabanımıza kaydettik.

İsterimizin son aşaması olan metnin özetinin çıkarılmasında daha önce bulmuş olduğumuz cümle skorlarını kullanıcıdan aldığımız cümle skoru thresholdundan büyük olan cümleleri seçerek kendi özet metnimize ekledik ve özeti kaydettik.

Daha sonra beşinci isterde bahsedilen ROUGE skoru hesaplama algoritmasını projemize ekledik.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) skoru, metin özetleme veya metin yüzünden çeviri gibi doğal dil işleme görevlerinde kullanılan bir değerlendirme metriğidir. ROUGE, otomatik olarak üretilen bir özetin, referans olarak kabul edilen bir özete ne kadar benzediğini ölçmek için kullanılır.

Böylelikle bu aşamanın da tamamlanmasıyla birlikte tüm isterleri gerçekleştirmiş olduk. Bulmuş olduğumuz tüm sonuçları kullanıcı arayüzünde açık bir şekilde gösterdik.

B. Geliştirme Ortamı

Projenin backend tarafı Java'nın Spring Boot ve Python'ın Flask frameworkü ile geliştirilirken frontend tarafı Angular ve ElectronJs frameworkü ile geliştirilmiştir. Veritabanı olarak Neo4j (NoSQL) kullanılmıştır.

Windows sistemlerde geliştirilip IDE olarak IntelliJ Idea Ultimate Version ve WebStorm kullanılmıştır. Projenin versiyon kontrolü tamamen github üzerinden yapılmıştır. IDE'lerimizde yazdığımız kodları veritabanı bağlantısı sağladıktan sonra localimizdeki portlarda çalıştırıp her türlü olumsuzluğa karşı önlem açısından testler yapılmıştır.

C. Kullandığımız Teknolojiler

• Backend Kısmı

- **Spring Boot:** Spring boot, Java programlama dilinde geliştirilen bir framework'tür. Java dilinde microservice yazılmasına olanak sağlar.
- **Flask:** Flask, Python ile yazılmış bir mikro web çerçevesidir. Belirli araçlar veya kitaplıklar gerektirmediğinden mikro çerçeve olarak sınıflandırılır.
- **Apache Kafka:** Apache Kafka, Apache Yazılım Vakfı tarafından geliştirilen, Scala ve Java ile yazılmış açık kaynaklı bir akış işleme yazılım platformu. Proje, gerçek zamanlı veri akışlarının işlenmesi için birleşik, yüksek verimli, düşük gecikme süreli bir platform sağlamayı amaçlamaktadır.
- **WebSocket:** WebSocket, tek bir TCP bağlantısı üzerinden tam çift yönlü iletişim kanalı sağlayan bir bilgisayar iletişim protokolüdür.

- **Swagger :** Web API geliřtirmede en önemli ihtiyalardan biri dokümantasyon ihtiyacıdır. Çünkü API methodlarının ne işe yaradığı ve nasıl kullanıldığı dokümantasyon içerisinde anlaşılır olması gerekir ve swagger bu dokümantasyonu online ve sürekli güncel olarak kullanmamıza olanak sağlayan araçtır.
- **Lombok :** Spring framework içerisinde class üretirken bu class'lara erişim sağlayıp kullanımı için getter, setter ve constructorlara ihtiyaç duyarız işte bu noktada basmakalıp kodları tekrar etmek yerine basit anotasyonlarla hızlandırmak ve daha az maliyetli hale getirmek amacıyla kullanılan kod üretme amacıyla üretilen bir kütüphanedir.

• Frontend Kısmı

- **Angular :** Angular, Google tarafından geliştirilen ve desteklenen SPA(Single Page Application) uygulama yapılmasına olanak sağlayan javascript frameworküdür. Projemizin ön yüzünü bu framework ile dizayn ettik.
- **Electron :** Electron, GitHub tarafından geliştirilen ve bakımı yapılan özgür ve açık kaynaklı bir yazılım iskeletidir. Çereve, Chromium tarayıcı motorunun bir çeşidi kullanılarak oluşturulan web teknolojilerini ve Node.js çalışma ortamını kullanarak masaüstü uygulamaları oluşturmak için tasarlanmıştır.

• Veritabanı Kısmı

- **Neo4j :** Neo4J Technology isimli firma tarafından açık kaynak olarak dağıtılan ve geliştirilen graph veritabanıdır. Verileri ve verilerin arasındaki ilişkileri saklamak amacıyla graf yapısı kullanan, açık kaynak kodlu bir NoSql veritabanı altyapısıdır.

D. Yalancı Kod

```

Start
Function
DesktopApplicationDevelopment
Start
Print "You are required to develop a desktop
application in the project."
Print "Create an Angular project."
Print "Integrate the project with ElectronJs."
Print "Launch the desktop application."
End
End

Function
ReadAndSeparateDocument
Start
Print "Read and process the document input from
the user."
Print "Separate the document into title and text
sections."
Print "Store the separated information in a JSON
object."
End
End

Function
StoreInNeo4j
Start
Print "Split the text into sentences."
Print "Save each sentence as a node in the Neo4j
database."
Print "Integrate the graph visualization of the text
using the Vis.js library in the user interface."
End
End

Function
EstablishSemanticRelationships
Start
Print "Establish semantic relationships between
sentences."
Print "Evaluate Word Embedding and BERT
algorithms for scoring the sentences."
Print "Use the Glove model to obtain similarity
results."
Print "Store the similarity results in the Neo4j
database."
End

```

End

Function
CalculateSentenceScores
Start

Print "Evaluate other parameters (for example, checking for proper nouns in a sentence, checking for numeric data, setting sentence similarity threshold, checking for keywords in the title, calculating TF-IDF value for each word) to calculate the sentence scores."

Print "Store all scores in the Neo4j database."

End

End

Function GenerateTextSummary
Start

Print "Create the summary of the text using the sentence scores."

Print "Select the sentences with scores above a user-defined threshold."

Print "Save the generated summary."

End

End

Function CalculateRougeScore

Start

Print "Add the ROUGE score calculation algorithm to the project."

Print "ROUGE score measures the similarity between an automatically generated summary and a reference summary."

End

End

Print "Executing the required steps in order:"

DesktopApplicationDevelopment()

ReadAndSeparateDocument()

StoreInNeo4j()

EstablishSemanticRelationships()

CalculateSentenceScores()

GenerateTextSummary()

CalculateRougeScore()

Print "All steps have been completed. Results are displayed in the user interface."

End

III. SONUÇLAR

Bizden istenen tüm isterler gerçekleştirilmiştir. Proje sonunda cross platform şekilde masaüstü uygulama geliştirme hakkında, doğal dil işleme ve doğal dil işleme algoritmaları hakkında ve graph veritabanları hakkında bilgiler edinilip öğrenildi. Gerekli testler yapılarak programın runtime hataları vermesi önlenmiştir.

KAYNAKLAR

[1] Masaüstü Uygulama Geliştirme

<https://www.electronjs.org/docs/latest/>

<https://medium.com/batech/>

[angular-io-ve-electron-js-ile-masa%C3%](#)

[BCst%C3%BC-uygulamas%C4%B1-geli%](#)

[C5%9Ftirmek-42827a4005a0](#)

<https://angular.io/quick-start>

<https://visjs.org/>

<https://visjs.github.io/vis-network/docs/network/>

[2] Graph Veritabanı

<https://neo4j.com/>

<https://www.baeldung.com/>

[spring-data-neo4j-intro](#)

<https://www.codementor.io/@asadali555/>

[introduction-to-spring-data-neo4j-17brd2jfvd](#)

<https://neo4j.com/developer/spring-data-neo4j/>

[3] Backend API's

http://www.robotiksistem.com/arduino_mega_2560_ozellikleri.html

<https://spring.io/projects/spring-boot>

<https://flask.palletsprojects.com/en/2.3.x/>

<https://atacomsian.com/blog/>

spring-boot-resttemplate-post-request-json-headers
[https://towardsdatascience.com/
creating-restful-apis-using-flask-and-python-655bad51b24](https://towardsdatascience.com/creating-restful-apis-using-flask-and-python-655bad51b24)

[4] Apache Kafka ve Websocket

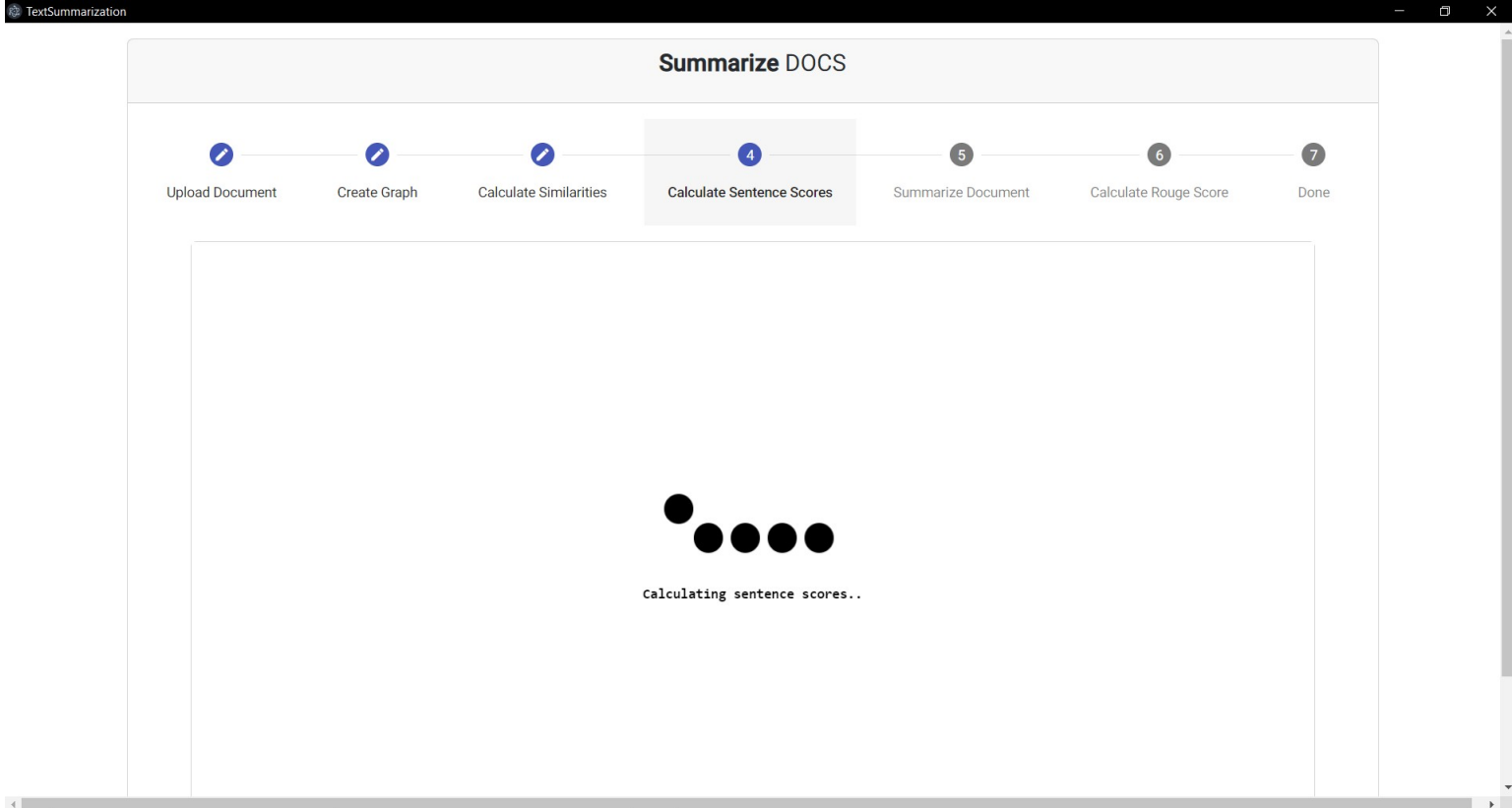
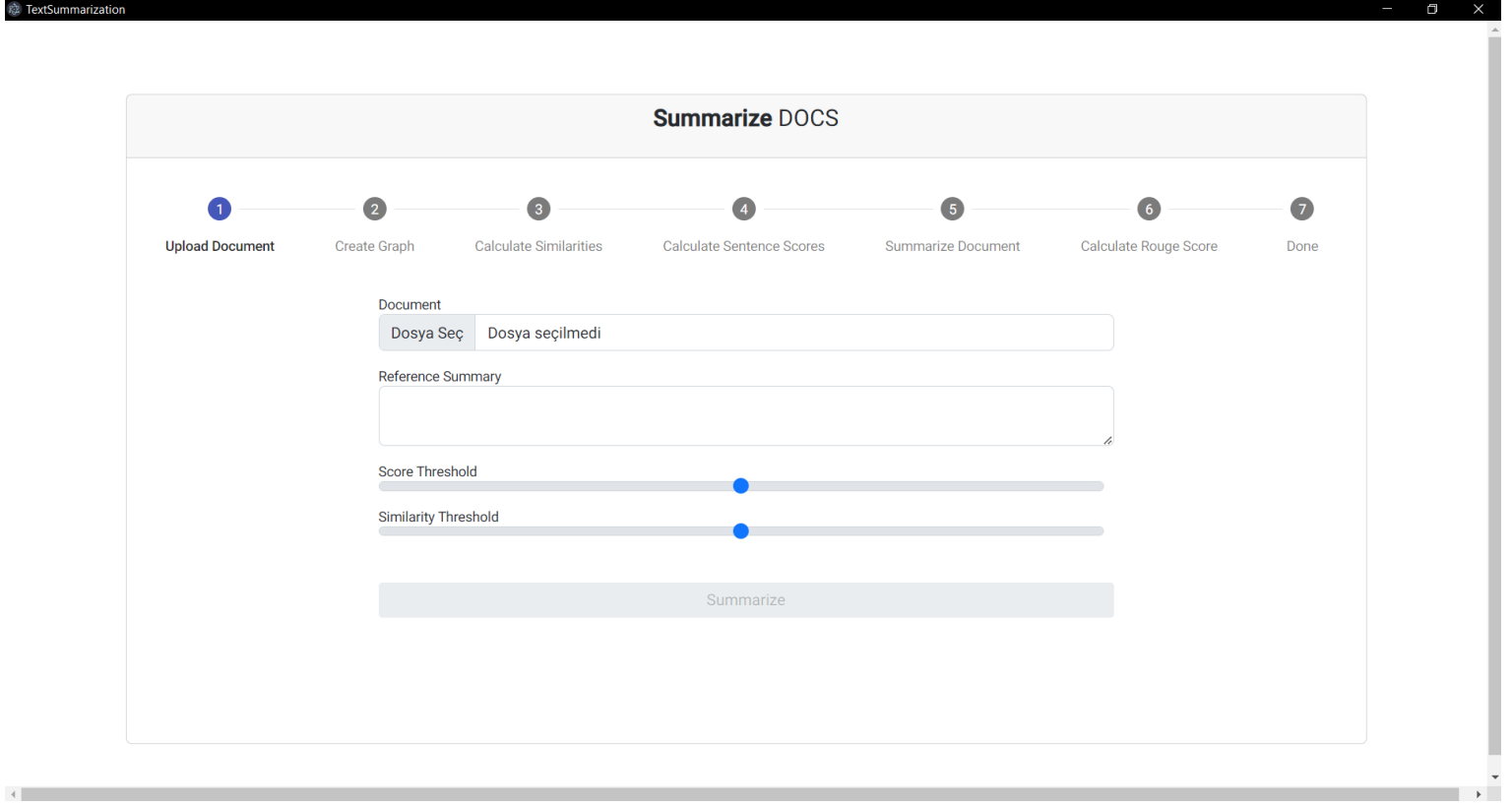
<https://kafka.apache.org/documentation/>
[https://medium.com/devopsturkiye/
apache-kafkaya-giri%C5%9F-3399e5f33f8e](https://medium.com/devopsturkiye/apache-kafkaya-giri%C5%9F-3399e5f33f8e)
<https://spring.io/projects/spring-kafka>
[https://seyhmusaydogdu.medium.com/
spring-boot-ile-apache-kafka-kullan%C4%B1m%C4%B1-66a793268a79](https://seyhmusaydogdu.medium.com/spring-boot-ile-apache-kafka-kullan%C4%B1m%C4%B1-66a793268a79)
[https://www.tutorialsbuddy.com/
kafka-python-producer-example](https://www.tutorialsbuddy.com/kafka-python-producer-example)
[https://towardsdatascience.com/
getting-started-with-apache-kafka-in-python-604b3250aa05](https://towardsdatascience.com/getting-started-with-apache-kafka-in-python-604b3250aa05)
<https://www.baeldung.com/websockets-spring>
[https://www.javainuse.com/spring/
boot-websocket](https://www.javainuse.com/spring/boot-websocket)
[https://muratdemirci.me/2023/02/05/
javada-spring-boot-ile-websocket-uygulamasi/](https://muratdemirci.me/2023/02/05/javada-spring-boot-ile-websocket-uygulamasi/)

[5] NLP

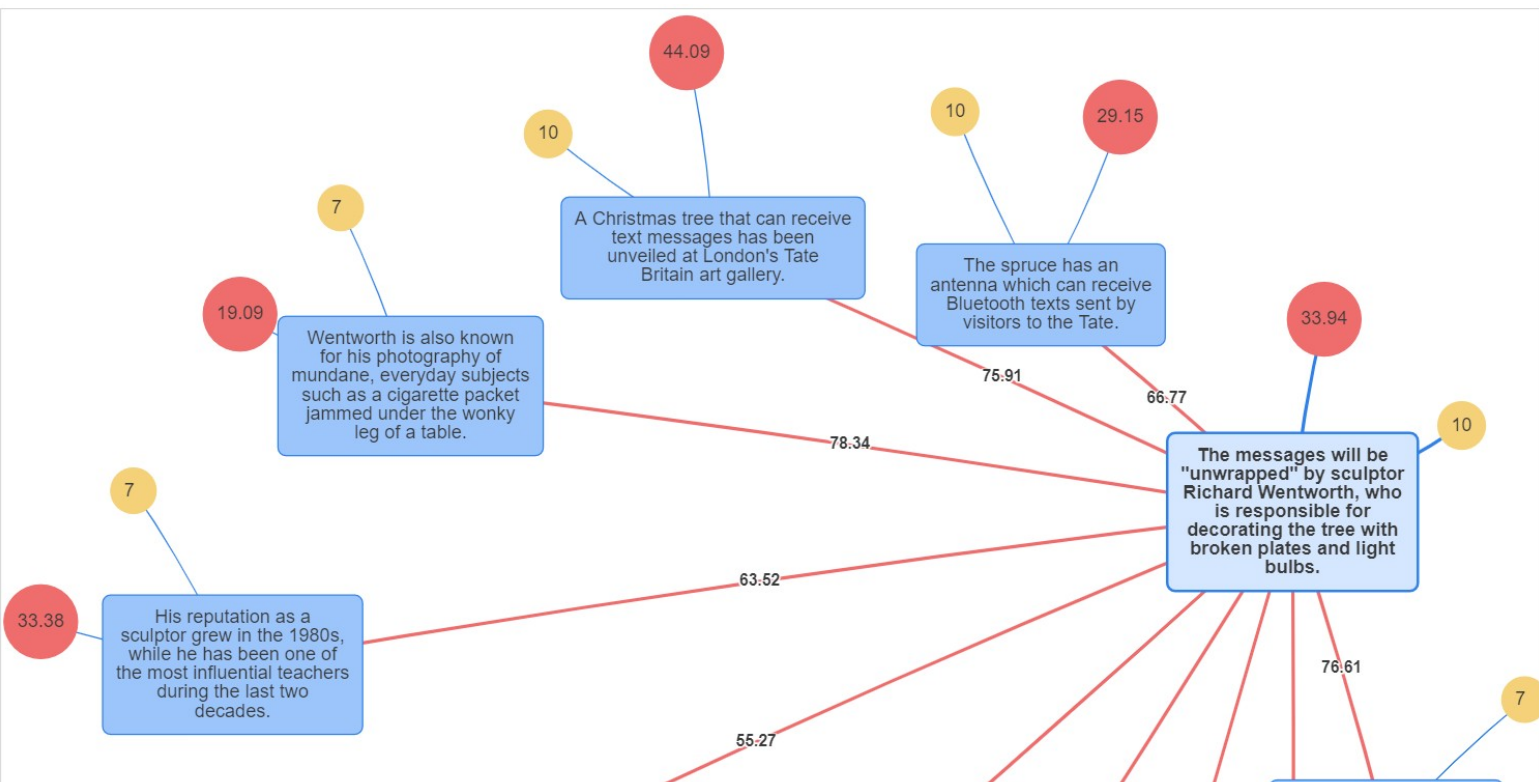
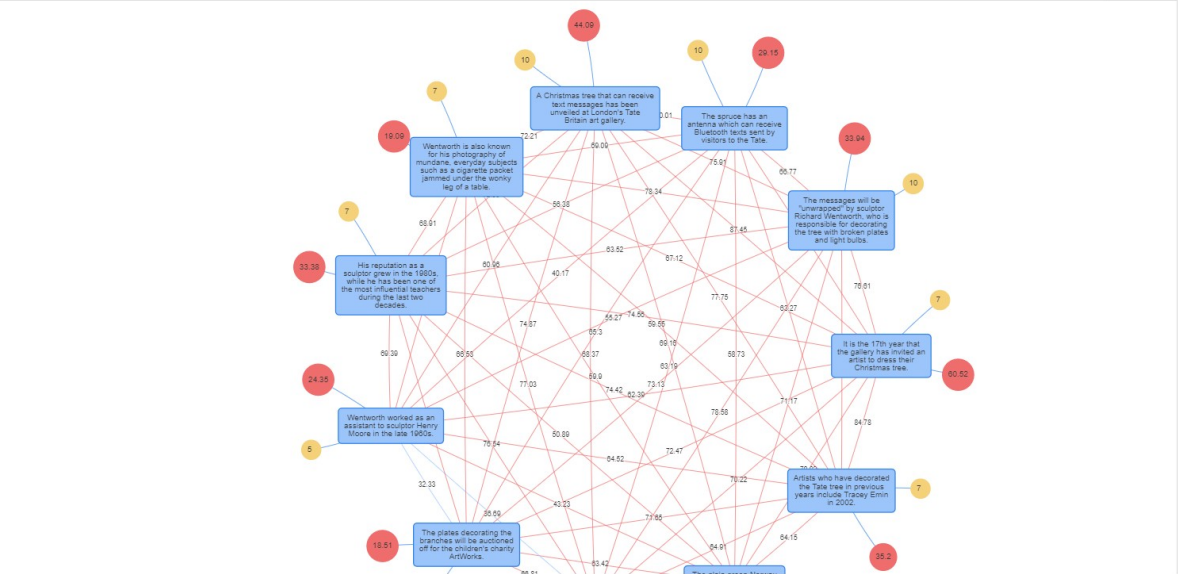
<https://nlp.stanford.edu/projects/glove/>
[https://www.kaggle.com/code/adeptvenugopal/
nlp-text-similarity-using-glove-embedding](https://www.kaggle.com/code/adeptvenugopal/nlp-text-similarity-using-glove-embedding)
[https://towardsdatascience.com/
word-embeddings-exploration-explanation-and-exploitation-with-code-in-python-5dac99d5d795](https://towardsdatascience.com/word-embeddings-exploration-explanation-and-exploitation-with-code-in-python-5dac99d5d795)
[https://mccormickml.com/2019/05/14/
BERT-word-embeddings-tutorial/](https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/)
[https://is-rajapaksha.medium.com/
bert-word-embeddings-deep-dive-32f6214f02bf](https://is-rajapaksha.medium.com/bert-word-embeddings-deep-dive-32f6214f02bf)
[https://towardsdatascience.com/
nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b](https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b)
[https://towardsdatascience.com/
tf-idf-explained-and-python-sklearn-implementation-b020c5e83275](https://towardsdatascience.com/tf-idf-explained-and-python-sklearn-implementation-b020c5e83275)
[https://blog.desdelinux.net/tr/
spacy-una-biblioteca-de-procesamiento-de-lenguaje-natural/](https://blog.desdelinux.net/tr/spacy-una-biblioteca-de-procesamiento-de-lenguaje-natural/)

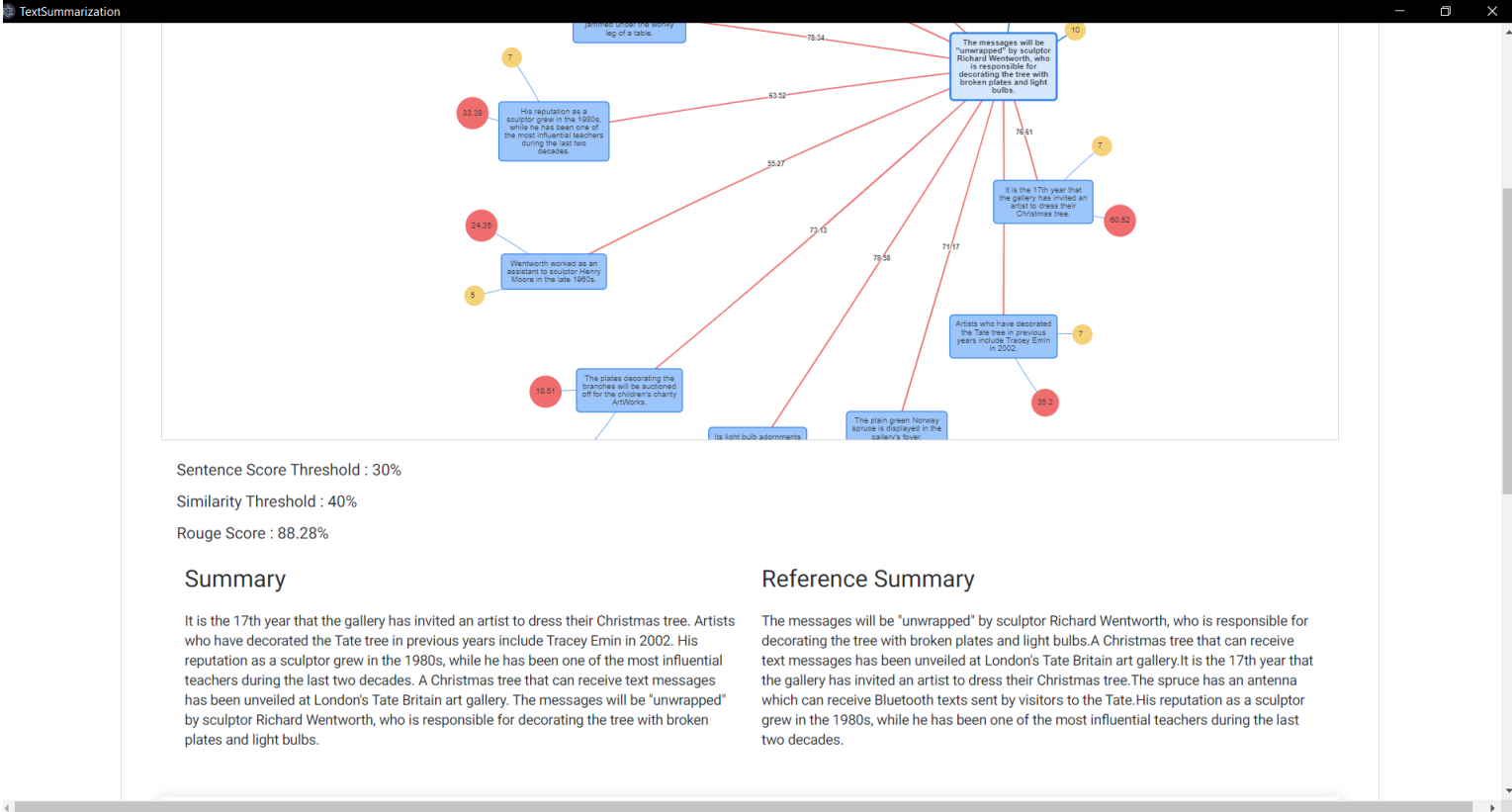
▪
▪

EKRAN GÖRÜNTÜLERİ



Summarize DOCS





TextSummarization

No.	Sentence	Score	Exceeded Edge Count	Is Included Summary
4	It is the 17th year that the gallery has invited an artist to dress their Christmas tree.	60.52	7	YES
5	Artists who have decorated the Tate tree in previous years include Tracey Emin in 2002.	35.2	7	YES
6	The plain green Norway spruce is displayed in the gallery's foyer.	24.81	7	NO
7	Its light bulb adornments are dimmed, ordinary domestic ones joined together with string.	16.36	6	NO
8	The plates decorating the branches will be auctioned off for the children's charity ArtWorks.	18.51	6	NO
9	Wentworth worked as an assistant to sculptor Henry Moore in the late 1960s.	24.35	5	NO
10	His reputation as a sculptor grew in the 1980s, while he has been one of the most influential teachers during the last two decades.	33.38	7	YES
11	Wentworth is also known for his photography of mundane, everyday subjects such as a cigarette packet jammed under the wonky leg of a table.	19.09	7	NO
1	A Christmas tree that can receive text messages has been unveiled at London's Tate Britain art gallery.	44.09	10	YES
2	The spruce has an antenna which can receive Bluetooth texts sent by visitors to the Tate.	29.15	10	NO
3	The messages will be "unwrapped" by sculptor Richard Wentworth, who is responsible for decorating the tree with broken plates and light bulbs.	33.94	10	YES

Reset