YAZILIM LABORATUVARI II PROJE 3

Beyhan KARADAĞ
Kocaeli Üniversitesi
Bİlgisayar Mühendisliği
190201012@koaceli.edu.tr

Alperen KOLAT
Kocaeli Üniversitesi
Bİlgisayar Mühendisliği
200201006@kocaeli.edu.tr

Barış KAKİLLİ
Kocaeli Üniversitesi
Bİlgisayar Mühendisliği
200201012@kocaeli.edu.tr

***Abstract*—,**
Bu projede bir masaüstü uygulaması geliştirilmiştir. Masaüstü uygulaması döküman yükleme kısmı ile çalıştırılır. Dosyadan alınan metin çeşitli temizleme işlemlerinden geçerek saflaştırılır. Bu saflaştırma işlemi kelime benzerlik algoritmalarının çalıştırılaması için gerekli olan adımlara içermekte ve farklı algoritmalar için farklı adımlar uygulanmaktadır. Bu işlem özet oluşturma algoritması amacıyla yapılmaktadır. Daha sonra elde edilen cümle skorları birbirleri ile kıyaslanarak benzerlik puanları elde edilir. Elde edilen skorlar ve cümeler toplu şekilde bir graph bağlantısı formunda ekranda gösterilir graphlara basılarak ilgili cümlenin tüm skor hesaplamaları görülebilir ve arayüzden seçilen treshold parametreleri ile belirlenen değerin üstündekilerin boyanma işlemi gerçekleştirilir. Hesaplanan benzerlikler yardımıyla yüksek orandaki cümleler seçilerek bir metin özetleme algoritması gerçeklenmiştir. Özetlenen metnin gerçek metin ile benzerliği rouge skorlaması kullanılarak ölçülür ve kullanıcıya bildirilmesi amaçlanmıştır.

I. GIRIŞ

Döküman yükenmesiyle analiz edilecek metni alma kısmı yapılmış olunur. Bilgisayarlar aracılığıyla dilin anlaşılması, temsil edilmesi ve manipüle edilmesiyle ilgilenen disiplin olan doğal dil işleme (NLP) alanında büyük bir öneme sahiptir.

Dil işleme, doğal dildeki metinlerin ve konuşmaların anlamlarını anlama ve analiz etme sürecini ifade eder. Gelişmiş dil işleme yöntemleri, makine öğrenimi ve yapay zeka teknikleriyle birleştirilerek metinleri otomatik olarak işleyebilen ve anlayabilen sistemlerin geliştirilmesine olanak sağlamaktadır.

Word embedding ise, dil işleme için kullanılan bir dizi teknikten biridir. Bu teknik, kelime ve kelime öbeklerini vektörlerle temsil etmeyi sağlar. Kelimelerin vektörel uzayda temsil edilmesi, onların anlamsal ilişkilerini ve benzerliklerini ölçmek için kullanılabilir. Bu sayede, metinler arasında semantik olarak yakın olan kelimelerin belirlenmesi ve anlam açısından benzer cümlelerin tanımlanması mümkün hale gelir.

BART (Bidirectional and AutoRegressive Transformers) ise, dil işleme için kullanılan bir öğrenme modelidir. BART, derin öğrenme tabanlı bir model olarak bilinir ve özellikle metin özetleme gibi görevlerde başarılı sonuçlar verir. BART, bir metin girdisi alır ve özetlenmiş bir çıktı üretir. Özetleme, metnin ana fikrini veya önemli bilgilerini kısaltarak sunma sürecidir.

Metin özetleme, uzun ve karmaşık metinlerin daha kısa ve öz bir şekilde sunulmasını sağlayan bir NLP görevidir. Özetleme modelleri, metinlerin önemli bilgilerini anlayarak ve seçerek, bu bilgileri daha kısa bir formda sunarlar. BART gibi modeller, büyük veri kümeleri üzerinde eğitilerek metin özetleme görevlerinde yüksek başarı elde edebilirler.

Son olarak, cümle benzerlik skorları, iki veya daha fazla cümle arasındaki benzerlik derecesini ölçen bir metrik veya skordur. Cümle benzerlik skorları, dil işleme modelleri tarafından kullanılarak metinler arasında karşılaştırma ve eşleştirme yapılmasına yardımcı olur. Bu skorlar, bir metin girdisiyle diğer metinlerin benzerlik düzeyini ölçerek, metin eşleştirme, çeviri ve bilgi alıntılama gibi birçok uygulamada kullanılabilir.

Projemiz bu çalışmaları birleştirerek onları kullanılabilir ve istenen görevleri, analizleri gerçekleştirebilir bir arayüzde sunarak 2 algoritmayı kıyaslayabilir hale getirebilir ve özetleme işlemini kullanıcıya sunar.

II. YÖNTEM

Proje geliştirilirken kullanım kolaylığı ve geniş kütüphane desteğinden dolayı python programlama dili tercih edilmiştir. Arayüz kısmı PyQt ile NLP algoritmaları için Pytorch, Transformers, Sklearn, Word2Vec kütüphanelerinden faydalanılmıştır. Graf gösterimi için networkx kütüphanesi kullanılmıştır. Program arayüzün açılıp dosya yükleme kısmının seçilmesiyle başlar. Pdf, metin belgesi veya .doc uzantılı dökümanların seçilmesi ile metin belleğe alınıp işlenmeye başlanır. İlk adımda cümleler ayrıldıktan sonra graf halinde göster butonuna basılarak görselleştirme kısmı çalıştırılır. Bu görselleştirme kısmı cümlelerin graflara ayrılamsını ve bütün algoritmaların hesaplanıp sonuçlarının toplanma işlemi ile birlikte çalışır. Bu işlemlerin sonuçları graf gösterim kısmında ilgili grafa tıklanarak ulaşılabilir. Bu veriler anlık olarak başka cümleler ile kıyaslanabilir. Arayüzden seçilen algoritmaya göre (word embedding veya BERT) hesaplanan cümle benzerlik skorları ve parametrelerden elde edilen skorlar kullanılarak özet elde edilir ve özet göster butonu ile özet için ayrılan yerde kullanıcıya sunulur.

Ana hatlarıyla tarif edilenler yapılırken arka planda dönen aşamaları 6 temel kısma ayırabiliriz. Bunlar sırasıyla metnin okunması, modele göre cümlelerin ayıklanmesı veya kelimelerin ayıklanması, word embedding, BERT, cümle parametre skorları, görselleştirme ve özetleme olarak sayabiliriz.

Bazı kısımları burada açıklarken kalan kısımları detaylarıyla alt bölümlerde açıklayacağız.

*1) **Metnin Okunması**:* Metin okuma işlemi basit gibi gözükse de başlık ayrımı konusuna gelince modelleme konusudna sıkıntılar çıkmaktadır. Bunu noktasız olması ve alt satıra geçmesi gibi özelliklerini göz önüne alarak bir yapı oluşturuldu ve ayırt etme işlemi gerçekleştirildi. Ayrıca pdf, doc, metin belgesi gibi farklı dosya türlerini okuyabilmekteyiz.

*2) **Cümle/Kelime Ayıklama**:* Cümlelerin temizlenmesi sırasında aşağıdaki adımlar izlenir.

- Tokenization: Bir metnin küçük parçalara ayrılmasıdır.
- Stemming: Kelimelerin kökünün bulunması işlemidir.
- Stop-word Elimination: Bir metindeki gereksiz sözcükleri çıkarma işlemidir. Stop word'ler, genellikle yaygın olarak kullanılan, ancak metnin anlamını belirlemekte önemli bir rol oynamayan kelime ve ifadelerdir.
- Punctuation: Cümledeki noktalama işaretlerinin kaldırılmasıdır.

Bu adımlar ile kelimeler temiz hale getirilir. Bu ilk aşama daha çok word embedding yöntemi için kullanılır. Word embeding kelimeler üzerinden çalıştığı için kelimelerin temizlenmesi ve köklerine ayrılamsı önemlidir.

BERT için ise cümleler yine aynı adımdam geçirilir ancak bu sefer kelime parçalaması yapılmaz cümleler cümle olarak kalmaya devam ederler ve modele bu şekilde verilirler.

*3) **Özetleme**:* Projede cümle seçerek özetleme yapılmalıdır, yani "var olan cümle yapısı bozulmadan cümleler seçilerek çıkarılıp özet elde edilecektir". Bu isteri yerine getirecek şekilde cümlelerin geldiği sıra ile eşiği geçen her cümlenin sırasıyla eklenmesi ile özet metin oluşturulmaktadır. Bu özetin başarısının hesaplanması kısmında ROUGE Skoru hesaplaması mevcuttur. Gerçek özet ile elde edilen özet bir fonksiyon aracılığı ile benzerliği hesaplanır ve Rouge skoru elde edilir.

A. Word Embedding

İlk olarak, veriye bir dizi adım uygulanarak işlenir. Örneğin, metin satırları yeni satır karakterlerine göre bölünür ve gereksiz boşluklar temizlenir. Bu şekilde, filtrelenmiş satırlar elde edilir.

Sonraki adımda, filtrelenmiş satırlar cümlelere bölünür ve her cümledeki gereksiz kelimeler (stop words) filtrelenir. Bu adımda, metindeki kelimelerin önemli olduğu varsayılarak, stop words olarak adlandırılan yaygın ve anlamsız kelimeler (örneğin "the", "and", "but" gibi) çıkarılır. Böylece, filtrelenmiş kelimeler elde edilir.

Daha sonra, Word2Vec modeli, filtrelenmiş kelimeleri alarak bir kelime gömme (embedding) uzayı oluşturur. Bu gömme uzayı, kelime vektörlerini temsil eder, yani her kelimenin anlamını ve bağlamını içeren sayısal temsillerdir. Model, metindeki kelime ilişkilerini ve benzerliklerini öğrenir.

Daha sonra, her cümlenin kelime vektörleri toplanarak cümle vektörleri elde edilir. Bu vektörler, cümlelerin anlamsal temsillerini içerir. Örneğin, bir cümledeki tüm kelimelerin vektörlerini toplayarak cümle vektörünü oluştururuz.

Son adımda, cümle vektörleri arasındaki semantik benzerliklerin ölçülmesi için kosinüs benzerliği kullanılır. Kosinüs benzerliği, iki vektör arasındaki açıyı temsil eder ve bu açı ne kadar küçükse, o kadar benzer olduklarını gösterir. Bu sayede, metindeki cümlelerin birbirleriyle olan semantik benzerlikleri ölçülerek bir benzerlik matrisi oluşturulur.

Word embedding, kelime ve cümleleri sayısal temsillerle ifade ederek metin işleme algoritmaları için daha anlamlı ve etkili hale getirir. Bu temsiller, metinler arasındaki semantik ilişkileri ve benzerlikleri ölçmek, anlamsal öznitelikler çıkarmak ve çeşitli NLP görevlerini gerçekleştirmek için kullanılabilir.

B. BERT

BERT, word embedding'in bir türü olarak kabul edilir. Model, kelime ve cümleleri vektör temsilleriyle ifade eder. Bu temsiller, metinlerin anlamını ve yapısını içerir. Benzerlik ölçümü için genellikle kosinüs benzerliği kullanılır. BERT, büyük metin veri kümesi üzerinde eğitilmiş olduğu için, dilin karmaşıklıklarını daha iyi yakalayabilir ve daha doğru semantik benzerlik sonuçları üretebilir.

Öncelikle, BERT modeli ve BERT tokenizer'ı ilgili önceden eğitilmiş ağırlıklarla yüklenir. Bu önceden eğitilmiş model, geniş bir metin veri kümesi üzerinde eğitilerek kelime ve cümlelerin anlamını yakalamayı öğrenir.

Metin girişi, "text" değişkeni olarak alınır ve BERT modeline uygun hale getirilir. Her cümle, tokenizer aracılığıyla belirli bir kelime dağılımına dönüştürülerek listelere eklenir.

Daha sonra, en uzun cümlenin uzunluğu bulunur ve diğer cümleler de aynı uzunluğa getirilerek doldurulur. Bu işlem, tüm cümlelerin aynı boyuta sahip olmasını sağlar.

Sonraki adımda, BERT modeline uygun tensorlar oluşturulur. Oluşturulan liste PyTorch tensor formatına dönüştürülerek "input_tensors" oluşturulur.

BERT modeli, giriş tensorlarını alır ve cümleleri kodlar. Kodlanmış katmanlar cümlelerin anlamsal temsillerini içerir.

Kodlanmış vektörler alındıktan sonra, kosinüs benzerliği hesaplanır.

C. Parametreleri Oluşturma

Özet çıkarma işlemi için toplam 5 adet parametre bulunmaktadır. Bu parametrelerin bulunması için ana metinde düzenlemelere gidilmesi gerekiyor. Bu düzenlemeler spacy kütüphanesi kullanılarak stop-wordlerin kaldırılması, noktalama işaretlerinin kaldırılması ve kelimelerin en yalın halinin bulunması işlemleri ile gerçekleşmiştir. İşlemlerden sonra her cümlenin skoru sırası ile hesaplanmaya koyuluyor. Cümle skorunu belirleyen 5 adet parametre vardır bu 5 parametrenin sonunda bir de ağırlıklı ortalaması alınarak final cümle skoru elde edilir. Bu 5 parametre aşağıdaki gibidir:

1) Özel isim kontrolü - P1: Cümledeki özel ismin cümlenin ne kadarını oluşturduğunu kontrol etme işlemidir. Formülü Cümledeki özel isim sayısı / Cümlenin uzunluğu şeklindedir.

Özel isimleri bulmak için spacynin label(etiket) sistemini kullandık. Bu etiket sistemi cümleyi tarıyor ve varlıkları bulup uygun etiketi uyguluyor. Özel isim bulma kısmı için bize 2 adet etiket lazım bu etiketler 'PERSON' ve 'ORG'dur. 'PERSON' etiketi cümledeki kurgusal da dahil olmak üzere insanları buluyor 'ORG' etkiketi ise cümledeki şirket, ajans, kurum ve benzerlerini buluyor.

2) Numerik veri kontrolü - P2: Cümlede ne kadar numerik veri geçtiğini kontrol etme işlemidir. Formülü ise şöyledir Cümledeki numerik veri sayısı / Cümlenin uzunluğu.

Numerik veri bulmak için özel isimde olduğu gibi yine spacy kütüphanesinde mevcut olan 'likenum' kullanılmıştır. Likenum numara olan her veriyi almaktadır bu şekilde aldığımız sonuç daha doğru ve daha güvenilir oluyor.

3) Node bağlantı kontrolü - P3: Cümle benzerliği eşiğini geçen nodeların bulunması işlemi. Formülü şu şekildedir Cümledeki başlıkta geçen kelime sayısı / Cümlenin uzunluğu.

Node bağlantılarını skorları Cümle sayısı*Cümle sayısı boyutunda bir Dizide tutulmakatadır bu dizi üzerinden 1. nodeun diğer nodelarla bağlantısı sırasıyla kontrol edilmektedir ve eşiği geçen nodelar ile tüm node bağlantı sayısı bölünerek oran bulunur. Bu oran cümlelerde asıl benzerliği ölçen kısım olduğundan epey önemli bir kısımdır.

4) Başlıktaki kelimeyi içerme kontrolü - P4: Cümlede başlıkati kelimelerin olup olmadığının kontrolüdür. Formülü Cümledeki başlıkta geçen kelime sayısı / Cümlenin uzunluğu şeklindedir.

Başlığı bulmak için verilen dökümandaki ilk yeni satır gelene kadarki kısım başlık olarak alınır eğer hiç yeni satır yoksa başlık da yok demektir. Eğer başlık yoksa başlığın ağırlıklı ortalamaya etkisi 0 olarak alınıyor ve diğer yerlere aktarılıyor bunu sebebi başlıksız metinlerde özet çıkarırken hatalı olmasını önlemek içindir

5) Kelime frekansı(TF/IDF) kontrolü - P5: Cümlede geçen tema kelime sayısının oranıdır. Formülü Cümlenin içinde geçen tema kelime sayısı / Cümlenin uzunluğu şeklindedir.

Tema kelime bulma işlemi için TF-IDF değerinin hesaplanması gerekir ama sadece 1 adet döküman olduğundan IDF değeri her zaman 1 olacaktır. Bu sebepten ötürü sadece TF kısmı hesaplanmaktadır. Dökümanda geçen tüm kelimelerden en çok geçen %10u tema kelime olarak alınmaktadır.

6) Final Skor - P6: Final skorun hesaplanması için ağırlıklı ortalama yöntemi kullanılmıştır. Eğer dökümanda başlık varsa final skor şu şekilde hesaplanır:

- Özel isim oranı * 0.13
- Numerik veri oranı * 0.10
- Eşiği geçen node oranı * 0.30
- Başlıkta geçen kelime oranı * 0.22
- Tema kelime oranı* 0.25

Eğer dökümanda başlık yoksa final skor şu şekilde hesaplanır:

- Özel isim oranı * 0.20
- Numerik veri oranı * 0.15
- Eşiği geçen node oranı * 0.35
- Başlıkta geçen kelime oranı * 0.00
- Tema kelime oranı* 0.30

Skorların hesaplanmasında ağırlık daha çok P3 ve P5 tarafına verilmiştir bunun sebebi ise özet çıkarma işleminde en önemli kısımlar olmalarıdır. Elde edilen bu final skorlar özetleme kısmında kullanılır.

D. Arayüz Tasarımı

Arayüz tasarımında PyQT kullanılmıştır. Arayüz iç içe pencere şeklindedir yapılmıştır.

Arayüz üzerinden Cümle Benzerliği Eşiği, Cümle Skoru Eşiği, Word embedding ve Bert modelleri arasında değişim, Dosya seçme, Cümleleri gösterme, Dosyayı graf olarak görüntüleme, özetleme ve özet metin, Rouge skoru kısımları gözükür.

Arayüz kısmında grafı göstermek için matplotlib ve networkx kütüphanelerini de PyQT yanında kullandık. Networkx graf oluşturma kısmında matplotlib grafı göstermek için kullanılmıştır.

III. Sonuç

Bu proje ile python da çok çeşitli kütüphaneler kullanarak bir özetleme uygulaması gerçekleştirdik. Masaüstü platformu hedefleyen bu projede kullanıcının kolay erişebielceği ve kullanabileceği bir arayüz amaçladık. Kullanıcı özetleme için parametrelere ağırlık vermenin yanında hem word embedding hem de BART modelini kullanarak oluşturabilecektir. Ayrıca oluşan düğümleri de cümle skorlarıan kadar kontrol edebileceği bir arayüzde parametrelerini görüntüleyebilmektedir. BART Modeli büyüklüğünden dolayı bazen yavaş da olsa kısa bir bekleme süresi ile hesaplamalara dahil edilmektedir. En son seçenekte özet görüntüleme ekranı ile de oluşturulan yeni metin ve asıl metin arasındaki ROGUE skorlaması görülebilmektedir.

IV. Deneysel Sonuçlar

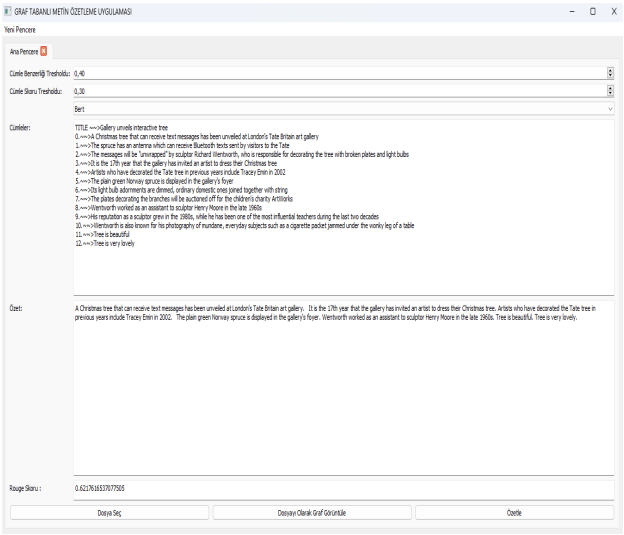

Fig. 1. **Ana Ekran**

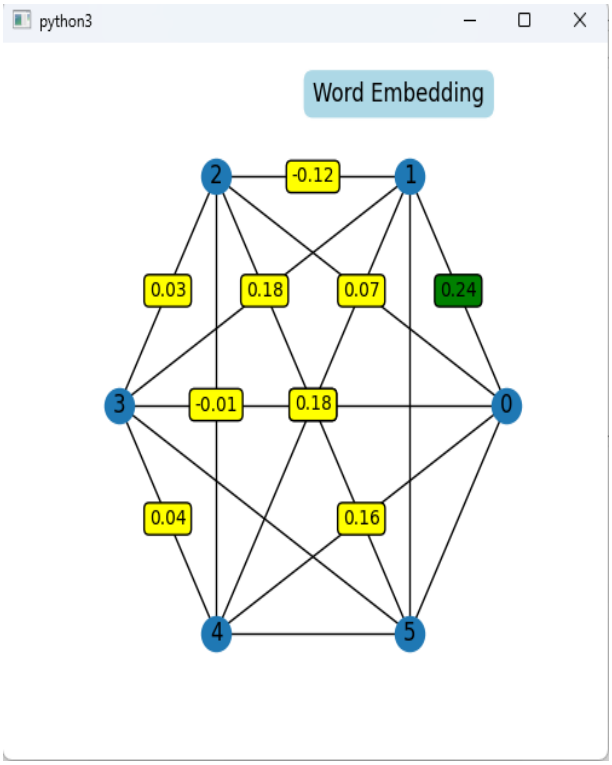

Fig. 2. **Graf Görüntüsü**

Kaynakça

- https://realpython.com/natural-language-processing-spacy-python/
- https://spacy.io/
- https://stackoverflow.com/
- https://www.pythonguis.com/pyqt5/
- https://doc.qt.io/qtforpython-6/
- https://app.diagrams.net/

V. Akış Diyagramı

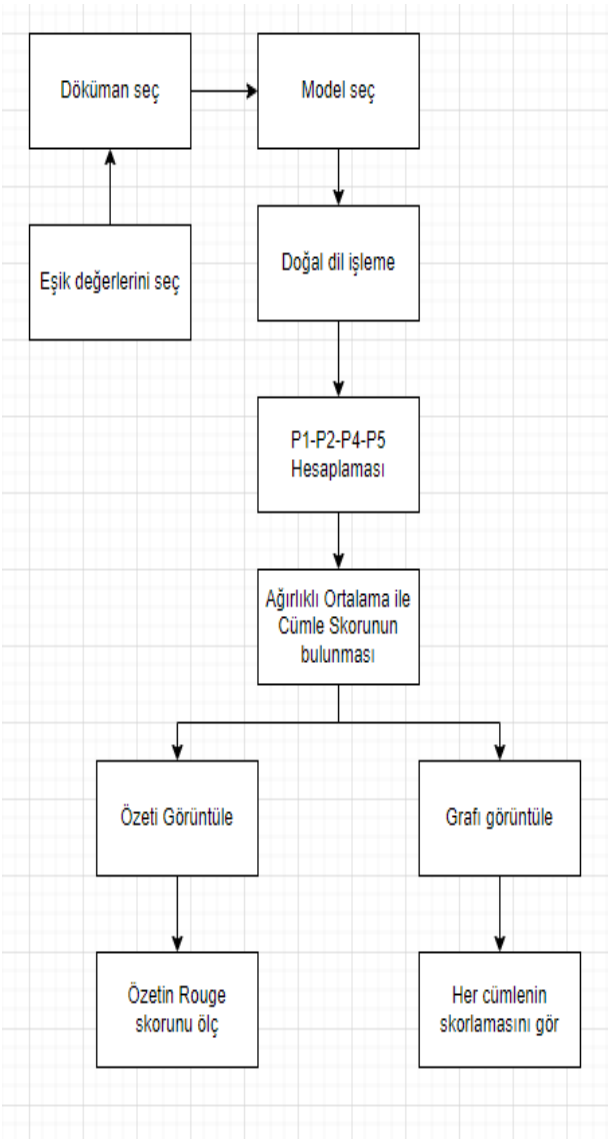

Fig. 3. **Akış Diyagramı**

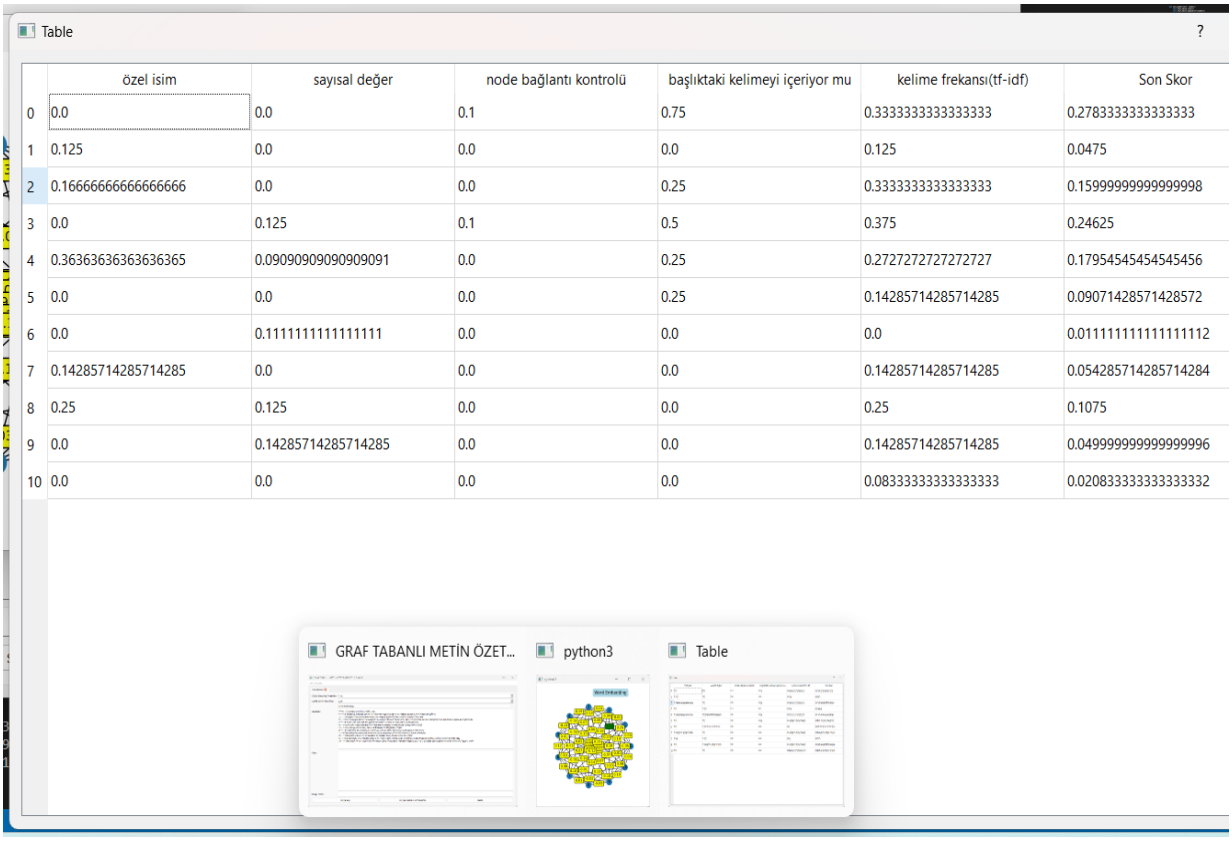

Fig. 4. **Tablo Görüntüsü**