Yazlab Proje I

TAHA HUTOĞLU-
ERDEM NAYIN
KOCAELİ ÜNİVERSİTESİ
4. SINIF
180202109-
180202050

Proje, Local host'ta çalıştırılan bir web uygulaması. Bu web uygulaması kullanıcı tarafından verilen url adresi veya url adresi kümelerini alıp bu url adresleri içerisinde bulunan yazılar üzerinde işlemler yapmaktadır. Bu işlemler url adresi girilen sayfada bulunan kelimelerin frekanslarını hesaplama, anahtar kelimesini çıkarma, verilen iki url adresi arasındaki benzerlik skorlaması yapma, verilen bir web sitesi url adresi kümesi ve bir url adresi için indexleme ve sıralama yapma, verilen url adresi kümesi içerisindeki yazılara semantik analizi yapma gibi 5 tane işlemi bulunmaktadır.

I. YAZILIM DILI VE GELIŞTIRME ORTAMI

Projenin kodları Pycharm kullanılarak python ve html dilinde yazılmış ve geliştirilmiştir.

II. GELIŞTIRME AŞAMALARI

A. Url adreslerinin alınması

Bu projedeki aşamaları gerçekleştirmek adına ilk başta kullanıcı tarafından işlemlerin gerçeklenmesinin istendiği url adresi veya url adres kümelerinin girişinin alınması gerekmektedir. Bu kısımda hangi işlem uygulanacağına bağlı olarak kullanıcı önüne giriş yapabileceği yazı kutuları(text box) getirilir. Bu işlem kullanıcının hangi işlemi uygulayacağını seçtiği butonlar yardımıyla yapılır.

B. Url adreslerindeki yazıların çekilip ayıklanması

Bu aşamada kullanıcının seçtiği işleme bağlı olarak girilen url adresi veya url adres kümelerinin arka tarafta açılıp url adreslerin içerisindeki bütün yazılar Beautiful Soup kullanılarak çekilir. Çekilen yazılar gereksiz kısımlardan ayıklanarak saf yazı haline getirilir ve işlemlerin uygulanmasına hazırlanır.

C. Temizlenen yazılar üzerinde frekans işleminin uygulanması

Bu aşamada url adresinden alınan ve temizlenen yazı içerisindeki kelimeler kontrol edilir. Term Frequency ve Inverse Document Frequency çarpımı kullanılarak kelime frekansları bulunur.

D. Temizlenen yazılar üzerinde anahtar kelime bulma işleminin uygulanması

Bu aşamada url adresinden alınan ve temizlenen yazı içerisindeki kelimeler kontrol edilir. Bu kontrol sonucunda hangi kelimelerin kaç kere kullanıldığı bilgisi elde edilir. Bunlar arasından da en çok tekabül edenler seçilerek anahtar kelimeler elde edilmiş olur.

E. Girilen 2 url adresi arasındaki benzerlik skorlaması yapılması

Bu aşamada kullanıcı tarafından girilen 2 url adresi de açılır. BeautfiulSoup kullanılarak urllerde bulunan textler, html taglerinden ve kod parçacıklarından arındırılarak alınır. Daha sonra bu textlerde bulunan noktalama işaretleri NLTK Corpus kullanılarak kaldırılır, tüm text küçük harflerden oluşacak şekilde düzenlenir, tek başlarına bir şey ifade etmeyen kelimeler olan Stopwords kaldırılır. Daha sonra CountVectorizer kullanılarak için nxk boyutlarında vektör oluşturulur. Burada n, iki textin kombinasyonunda geçen unique kelimeleri; k ise toplam text sayısını temsil eder. Vektörde bulunan değerler ise belirli bir kelimenin ilgili textte kaç kez geçtiğidir. Daha sonra n boyutlu uzayda bu vektörler arasındaki Cosinus Açısı hesaplanarak textlerin benzerlik hesabı yapılmış olur. Literatürde bu hesaplamaya Cosine Similarity denir.

F. Girilen url adresi ve url adres kümeleri için indexleme ve sıralama yapılması

Bu aşamada girilen bu url adresinin içeriği ile url adres kümesindeki her bir web sayfasının içeriklerinin benzerlik skorları ayrı ayrı hesaplanır. Ancak bu sefer skor hesaplaması yaparken bu url adres kümesinde bulunan web sayfalarının içeriğine ilaveten yine bu sayfalarda bulunan tüm alt url adresleri de dikkate alınır. Verilen kümeden seçilen bir url için, bu urlin alt url adresleri içinden bir tanesi rastgele seçilir. Daha sonra rastgele seçilen bu url adresinin içeriğindeki alt urllerden de bir tanesi rastgele seçilir. Böylece ana url, ana urlin alt kümesinden seçilen bir url, ana urlin alt kümesinden seçilen urlin içeriğinde bulunan alt urllerden de bir tanesi seçilmiş olur ve ana kümede verilen bir url için iki tanesi alt urller olmak üzere toplam üç url seçilmiş olur ve bu urllerle benzerlik hesabı yapılır. Seçilen url eğer bir webpage değilse -örneğin (.jpg), (.png), (.js) uzantılı dosyalar- benzerlik hesabı yapılamaz ve benzerlik sonucu sıfır olarak döndürülür.

G. Girilen url adresi ve url adres kümeleri için semantik analizinin yapılması

Bu aşamada girilen bu url adresinin içeriği ile url adres kümesindeki her bir web sayfasının içeriklerinin benzerlik skorları ayrı ayrı hesaplanır. Fakat bu hesaplama yapılırken girilen url adresindeki anahtar kelimelerin benzer anlamları NLTK ve WordNet kullanılarak hesaba katılır. Bu işlem verilen url adres kümesindeki her bir web sayfası için yapılırken aynı zamanda bir önceki isterdekiyle aynı derinlik işlemleri bu web sayfalarının alt url adreslerinde bulunan web sayfalarına da uygulanır.

III. UYGULAMANIN ÇALIŞMASI

Uygulama açıldığında öncelikle kullanıcı karşısına 5 tane seçenek sunulmaktadır. Bunlar Frekans Bul, Keyword ve Similarity Bul, Site İndexle ve Sırala, Semantik Analiz Yap şeklindedir.

Kullanıcı eğer Frekans Bul seçeneğini seçer ise karşısına bir tane yazı kutusu(text box) gelmektedir. Kullanıcıdan bu kutu içerisine hangi web sitesinin içerisisndeki yazının frekansının bulunması isteniyor ise o web sitesinin url adresinin girilmesi beklenir.

Kullanıcı bu url adresini girdikten ve submit butonuna tıkladıktan sonra bu url adresi kullanıcının görmediği arka tarafta açılır. Bu web sitesinin içindeki yazı bilgisi çekilir. Bu yazı bilgisi düzgünce ayıklanıp temizlenerek saf yazı haline dönüştürülür. Ardından Bu yazı içerisindeki kelimelerin yazı içerisinde kaç kere geçtiği bilgisi aranır. Bu aramanın tamamlanmasının ardından kullanıcı karşısına toplam kelime sayısı, toplam cümlesi sayısı, kelimeler ve bu kelimelerin yazı içerisinde kaç kere kullanıldığı bilgileri getirilir.

Kullanıcı eğer keyword ve Similarity Bul seçeneğini seçer ise kullanıcı karşısına iki adet yazı kutusu gelmektedir. Kullanıcıdan bunların ilkinin içerisine anahtar kelimelerin bulunacağı web sitesinin url adresinin, ikincisine ise ilk url adresi ile arasındaki benzerlik skorunun hesaplanacağı ikinci bir url adresinin girilmesi beklenmektedir. Kullanıcı bu bilgilerini girmesinin ve submit butonuna basmasının ardından bu iki url adresi alınır. İki url adresi de sırasıyla açılır. Açılan url adresleri içerisindeki yazılar çekilir ve ayıklanıp, temizlenip saf yazı haline getirilir. Ardından en çok geçen 5 adet kelime bulunur. Bu kelimeler yazıların anahtar kelimeleri olarak atanır. Daha sonra benzerlik oranının da hesaplanmasının ardından kullanıcı karşısına her iki url adresinin de içerisindeki yazıların anahatar kelimeleri, frekansları ve bu iki yazının arasındaki benzerlik oranı yazdırılır.

Kullanıcı eğer Site İndexle seçeneğini seçer ise kullanıcı karşısına biri kısa diğeri uzun olmak üzere iki farklı yazı kutusu getirilir. Bu kutuların ilkine tek bir url adresi, ikincisine ise url adres kümesi girilmesi beklenmektedir. Bu girişler yapıldıktan sonra submit butonuna tıklandığında bu url adres kümelerindeki her bir url adresi açılır. İçerisindeki yazılar alınır, ayıklanır, saf yazı haline getirilir. Ardından bu url adreslerinin içerisinde olan diğer url adresleri de açılır. Hepsi bir liste içerisinde depolanır ve ana kümedeki her url için, urllerin içerisinden birisi seçilir. Daha sonra bu seçilen url için de alt kümeleri bulunur ve bu alt kümenin içinden bir tane url seçilir.Böylece ana url ve ana urlden çıkan 2 alt url olmak üzere toplam 3 url olmuş olur. Bu işlem kullanıcı tarafından girilen url kümesindeki her url için tekrarlanır. Daha sonra bu işlemler url adres kümesindeki diğer url adreslerine de uygulanarak tek tek yazılar çıkarılır. Bu işlemlerin ardından ilk başta kullanıcının tek olarak girdiği url adresi açılır ve aynı yazı ayıklama işlemi uygulanır. Yazı çıkarıldıktan sonra yazı içerisindeki anahtar kelimeler belirlenir. Bu işlemin de tamamlanmasının ardından bu tek url adresinin anahtar kelimeleri url adres kümesindeki url adreslerinin ve alt url adreslerinin yazıları içerisinde bulunma oranına göre benzerlik oranları hesaplanır. Ardından bu orana göre yukarıdan aşşağıya doğru bir sıralama yapılarak url adresleri benzerlik oranlarıyla birlikte kullanıcı karşısına çıkartılır.

Kullanıcı eğer Semantik Analiz Yap seçeneğini seçer isekullanıcı karşısına biri kısa diğeri uzun olmak üzere iki farklı yazı kutusu getirilir. Bu kutuların ilkine tek bir url adresi, ikincisine ise url adres kümesi girilmesi beklenmektedir. Bu girişler yapıldıktan sonra submit butonuna tıklandığında bu url adres kümelerindeki her bir url adresi sırasıyla açılır. İçerisindeki yazılar alınır, ayıklanır, saf yazı haline getirilir. Ardından bu url adreslerinin içerisinde olan diğer url adresleri de açılır. Hepsi bir liste içerisinde depolanır ve ana kümedeki her url için, urllerin içerisinden birisi seçilir. Daha sonra bu seçilen url için de alt kümeleri bulunur ve bu alt kümenin içinden bir tane url seçilir. Böylece ana url ve ana urlden çıkan 2 alt url olmak üzere toplam 3 url olmuş olur. Bu işlem kullanıcı tarafından girilen url kümesindeki her url için tekrarlanır. Daha sonra bu işlemlerin ardından ilk başta kullanıcının tek olarak girdiği url adresi de açılır ve aynı yazı ayıklama işlemi uygulanır. Yazı çıkarıldıktan sonra yazı içerisindeki anahtar kelimeler belirlenir. Bu işlemin bir önceki işlemden farkı burada ortaya çıkmaktadır. Anahtar kelimelerin belirlenmesinin ardından bu anahtar kelimelerin eş ve yakın anlamlıları da tespit edilir. Bu işlemin de tamamlanmasının ardından bu tek url adresinin anahtar kelimeleri ve eş anlamlıları url adres kümesindeki url adreslerinin ve alt url adreslerinin yazıları içerisinde bulunma oranına göre benzerlik oranları hesaplanır. Ardından bu orana göre yukarıdan aşşağıya doğru bir sıralama yapılarak url adresleri benzerlik oranlarıyla birlikte kullanıcı karşısına çıkartılır

IV. KULLANILAN CLASS'LAR

Projede kullanılan class'lar aşağıda belirtilmiştir.

A. frekansFinder

Bu class girilen url adresini açar. İçerisinde bulunan yazıyı çeker. Çekilen yazıyı ayıklayarak saf yazı haline getirir. Saf yazı içerisindeki kelimeleri ve tekrar sayılarını (frekanslarını) bulur. Gönderilen url adresi içerisindeki kelimelerin frekanslarını bulmaya yarar.

B. keywordSimilarity

Bu class gelen url adresi açar. İçerisindeki yazı bilgilerini çeker. Gelen yazı bilgilerini ayıklayarak saf yazı haline getirir. Sonrasında class'a gelen anahtar kelimelerin bu saf yazı içerisinde kaç kere tekrar edildiğine bakılarak bu yazı ile benzerlik oranını bulur. Başka bir yazının anahtar kelimelerini gönderilerek class'a gönderilen url adresi ile anahtar kelimelerin bulunduğu url adresi arasındaki benzerlik oranını bulmaya yarar

C. textSimilarity

Bu class gelen url adreslerini açar. İçerlerindeki yazı bilgilerini çeker. Gelen yazı bilgilerini ayıklayarak saf yazı haline getirir. Ardından bu iki yazı bilgilerini karşılaştırarak iki url adresi içerisindeki yazı bilgilerinin benzerlik oranını bulur.

D. *synonymSimilarity*

Bu class'da gönderilen url adresi içerisindeki yazı bilgisi çekilir. Çekilen yazı bilgilerini ayıklayarak saf yazı haline getirilir. Gelen kelimeler ile bu url adresinden alınan yazı arasındaki benzerlik oranı kelimelerin yazı içerisinde geçme derecesine bağlı olarak hesaplanır.

V. Yalancı Kod

1. Başla.
2. Kullanıcının seçenek seçmesini bekle.
3. Frekans Bul seçilirse.
4. Kullanıcı girişi bekle.
5. Girilen url adresindeki yazının frekansını bul.
6. Frekansı yazdır.
7. Kullanıcının bitirmesini bekle.
8. keyword ve Similarity Bul seçilirse.
9. Kullanıcı girişini bekle.
10. Girilen url adreslerindeki yazının anahtar kelimelerini bul.
11. Benzerlik oranlarını bul.
12. Bulunan değerleri yazdır.
13. Kullanıcının bitirmesini bekle.
14. Site İndexle seçilirse
15. Kullanıcı girişini bekle.
16. Girilen url ile url adresleri ve alt url adresleri arasındaki benzerlik oranlarını anahtar kelimelerin yardımıyla bul.
17. Bu oranlara göre sırala.
18. Sıralı değerleri yazdır.
19. Kullanıcının bitirmesini bekle.
20. Semantik Analiz Yap seçilirse.
21. Girilen url adresindeki anahtar kelimelerin eş anlamlılarını bul.
22. Girilen url ile url adresleri ve alt url adresleri arasındaki benzerlik oranlarını anahtar kelimelerin ve eş anlamlılarının yardımıyla bul.
23. Bu değerleri yazdır
24. Kullanıcının bitirmesini bekle.
25. Bitir.

VI. Karşılaşılan Problemler

1. Web ile hiç uğraşmadığımızdan dolayı projeyi tanımlama ve oluşturma kısmında biraz zorlandık. Bu nedenle kodlamaya başlamadan önce çokça araştırma yapmamız gerekti. Projeyi tamamlama süresinin yarısından çoğu araştırma yapmakla geçti.
2. Dili belirleme konusunda biraz zorlandık. Geçen senelerde c# ile uğraştığımızdan asp.net ile yapmayı düşünmüştük fakat araştırmalarımız sonucunda pythonun doğal dil işlemesi ve matematiksel analiz konularında daha rahat olabileceğini gördük ve python dilini kullanmaya karar verdik.
3. Pythona hakim olmadığımızdan dolayı kodlarken karşılaştığımız basit hatalar karşısında çözüm bulmamız zorlaştı. Örnek olarak bir dizi oluştururken çarpma işlemiyle çoğalttığımızdan dolayı iki boyutlu dizimize bir eleman atadığımızda onu diğer elemanlara da kopyalıyordu. Böyle durumlar biraz zamanımız aldı.

VI. KAYNAKÇA

[1]Extracting main text from any given web page [*https://www.quora.com/How-can-I-extract-the-main-text-from-any-given-webpage]*
[2] Finging frequansy of each word in string in python [https://www.geeksforgeeks.org/find-frequency-of-each-word-in-a-string-in-python/]
[3] Keyword extracting in python [https://www.analyticsvidhya.com/blog/2020/11/words-that-matter-a-simple-guide-to-keyword-extraction-in-python/]
[4] Calculating string simularity in python [https://towardsdatascience.com/calculating-string-similarity-in-python-276e18a7d33a]
[5] Python web development [https://www.youtube.com/watch?v=zuxzE7--RYM]
[6] Python web framework [https://www.youtube.com/watch?v=F5mRW0jo-U4]
[7] Sentiment analysis in python [https://realpython.com/sentiment-analysis-python/]
[8] Extracting text from html file in python [https://stackoverflow.com/questions/328356/extracting-text-from-html-file-using-python]
[9]Extracting text from html file in python [https://www.kite.com/python/answers/how-to-extract-text-from-an-html-file-in-python]
[10] Flask for python [https://www.youtube.com/watch?v=Z1RJmh_OqeA]
[11] Sentiment classification [https://towardsdatascience.com/latent-semantic-analysis-sentiment-classification-with-python-5f657346f6a3]
[12] How to crawl a web page [https://www.digitalocean.com/community/tutorials/how-to-crawl-a-web-page-with-scrapy-and-python-3]
[13]Get links from web page [https://pythonprogramminglanguage.com/get-links-from-webpage/]
[14]beautifulsoup [https://stackoverflow.com/questions/1080411/retrieve-links-from-web-page-using-python-and-beautifulsoup]
[15]Converting html to text [https://stackoverflow.com/questions/14694482/converting-html-to-text-with-python]
[16]html to text [https://github.com/Alir3z4/html2text/blob/master/docs/usage.md]
[17]html to text python [https://www.programcreek.com/python/example/86006/html2text.HTML2Text]
[18]Wordnet with nltk [https://www.guru99.com/wordnet-nltk.html]

[19]deep levels of page with beautifulsoup
[https://stackoverflow.com/questions/53179817/how-to-get-the-deep-levels-of-the-page-while-crawling-by-beautifulsoup]
[20]web scraping error
[https://stackoverflow.com/questions/16627227/http-error-403-in-python-3-web-scraping]
[21]Run flask with debug
[https://garmoncheg.blogspot.com/2018/03/run-flask-with-debug-in-pycharm.html]
[22]Two dimensional arrays
[https://snakify.org/en/lessons/two_dimensional_lists_arrays/]
[23]Text similarities
[https://medium.com/@adriensieg/text-similarities-da019229c894]
[24]String matching with machine learning
[https://medium.com/@ertuodaba/string-matching-using-machine-learning-with-python-matching-products-of-getir-and-carrefoursa-f8ce29d2959f]
[25]Fast string matching with python
[https://medium.com/@sagnik.jena/an-efficient-and-better-method-for-text-matching-d3382c7c9457]
[26]How to rank text content by semantic similarity
[https://towardsdatascience.com/how-to-rank-text-content-by-semantic-similarity-4d2419a84c32]
[27]Dublicate keys in dictionary with python
[https://stackoverflow.com/questions/10664856/make-a-dictionary-with-duplicate-keys-in-python]
[28]Document simlarty with python
[https://dev.to/coderasha/compare-documents-similarity-using-python-nlp-4odp]
[29]Fuzzy string matching in python
[https://www.datacamp.com/community/tutorials/fuzzy-string-python]
[30]overwiev of text similarity
[https://towardsdatascience.com/overview-of-text-similarity-metrics-3397c4601f50]
[31]How to find a similarity metric between two string
[https://www.kite.com/python/answers/how-to-find-a-similarity-metric-between-two-strings-in-python]
[32]Unresolved referance error
[https://stackoverflow.com/questions/50909208/unresolved-reference-in-python-method]
[33]iterating dictionary
[https://stackoverflow.com/questions/3294889/iterating-over-dictionaries-using-for-loops#:~:text=To%20iterate%20over%20key-value,items()%20.&text=This%20is%20a%20very%20common%20looping%20idiom.]
[34]İterating over dictionaries
[https://stackoverflow.com/questions/3294889/iterating-over-dictionaries-using-for-loops#:~:text=To%20iterate%20over%20key-value,items()%20.&text=This%20is%20a%20very%20common%20looping%20idiom.]
[35]Combining two list
[https://stackoverflow.com/questions/7271385/how-do-i-combine-two-lists-into-a-dictionary-in-python]
[36]Flask post request ect
[https://www.youtube.com/watch?v=D5zrAk9mlF8]
[37]Flask tutorial
[https://www.youtube.com/watch?v=UIJKdCIEXUQ&list=PL-osiE80TeTs4UjLw5MM6OjgkjFeUxCYH&index=5]
[38]Bir videoda Flask
[https://www.youtube.com/watch?v=lBFQYkcZ3dw&list=PLIHume2cwmHfDUQ2t6f-Q7VCBvNSEAMAS&index=9]
[39]ToDo list in python flask
[https://www.youtube.com/watch?v=fEqOW6FrokA]
[40]Flask http methods
[https://www.youtube.com/watch?v=9MHYHgh4jYc]