

Yazar İşbirliği Ağı Analizi ve Görselleştirmesi

Muhammet Berat Ak
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
230202037

Lütfi Alpaslan Hazar
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
230202067

I. ÖZET

Bu proje, akademik bir veri seti kullanarak yazarlar arasındaki iş birliği ilişkilerini modelleyen bir graf yapısı oluşturmayı ve bu graf üzerinden çeşitli veri yapısı ve algoritma konseptlerini uygulamayı amaçlamaktadır. Proje kapsamında, öğrencilerden hem teorik bilgilerini hem de uygulamalı becerilerini geliştirmeleri beklenmektedir. Ayrıca proje sonunda, web arayüzü ile birlikte analiz sonuçları etkileşimli olarak sunulmaktadır.

II. GİRİŞ

Bu proje, akademik bir veri setinden yola çıkarak yazarlar arası işbirliklerini grafik tabanlı modellemeyi ve bu model üzerinde çeşitli algoritmalar uygulamayı hedefler. Proje, veri yapıları ve algoritmalar alanındaki bilgileri pratiğe dökmeyi amaçlamaktadır. Ayrıca oluşturulan arayüz sayesinde kullanıcıların etkileşimli olarak analiz yapabilmeleri hedeflenmiştir.

III. YÖNTEM

Bu projede, Python programlama dili ve Dash framework'ü kullanılarak etkileşimli bir web arayüzü oluşturulmuştur. Graf veri yapılarını modellemek ve algoritmaları uygulamak için kendi özel sınıflarımızı oluşturduk.

A. Veri İşleme

Veri işleme aşamasında, 'pandas' kütüphanesi kullanılarak Excel dosyasından veriler okunmuş ve temizlenmiştir. Bu aşamadaki temel adımlar şunlardır:

- **Excel Dosyasını Okuma:** Pandas kütüphanesinin 'read_excel' fonksiyonu kullanılarak, Excel dosyası bir DataFrame objesine dönüştürülmüştür.
- **Yazar ID'leri Oluşturma:** Her yazara benzersiz bir ID atanmış, bu ID'ler ve yazar isimleri bir sözlükte tutulmuştur. Bu, yazarların program içinde daha kolay yönetilmesini sağlar.
- **Makale Bilgilerinin Saklanması:** Makale adları, DOI'ler ve yazarları içeren bilgiler sözlük veri yapısında tutulmuştur. Bu, makalelerin analiz ve görselleştirme aşamasında daha kolay erişilebilir olmasını sağlar.

B. Graf Veri Yapısı

Yazarlar arasındaki işbirliği ağını modellemek için kendi özel 'Graph' sınıfımızı oluşturduk. Bu sınıf, temel olarak düğümleri (yazarlar) ve kenarları (işbirlikleri) saklamak için

Python'ın sözlük veri yapısını kullanır. Graf yapımızda aşağıdaki özellikleri implement ettik:

- **Düğüm Ekleme:** Graf yapısına yeni yazarları eklemek için 'add_node' metodunu kullanılır. Bu metod, her yazara özel bir ID atar. K
- **Düğüm ve Kenar Sayısını Getirme:** Grafik üzerindeki düğüm ve kenar sayısını almak için 'get_number_of_nodes' ve 'get_number_of_edges' metodları yazılmıştır.

C. İster Uygulamaları

Projede istenen işlevsellikleri yerine getirmek için çeşitli algoritmalar ve veri yapıları kullanılmıştır. Bu kısımda istelere göre yapılan uygulamalar detaylı bir şekilde anlatılmıştır.

- **İster 1: En Kısa Yol Algoritması** İki yazar arasındaki en kısa yolu bulmak için BFS (Breadth-First Search) algoritmasını uyguladık.
 - **Kuyruk Kullanımı:** BFS algoritması için bir kuyruk veri yapısı kullandık. Bu kuyruk, ziyaret edilecek düğümleri sırayla tutar.
 - **Ziyaret Edilmiş Düğümleri Takip Etme:** Ziyaret edilmiş düğümleri takip etmek için set veri yapısı kullandık. Bu, sonsuz döngülerin önüne geçmemize yardımcı oldu.
 - **Mesafe ve Önceki Düğüm Takibi:** Her düğümün başlangıç düğümüne olan mesafesini ve yol boyunca önceki düğümü saklamak için dictionary veri yapıları kullandık.
- **İster 2: Ağırlıklı Kuyruk Oluşturma** Bir yazarın işbirliği yaptığı diğer yazarları, makale sayılarına göre sıralamak için bir kuyruk oluşturduk. Bu adımda, kenar yoğunluğu ve her yazar için makale sayıları kullanılarak yazarların ağırlıkları belirlenmiştir. Ağırlıklar belirlendikten sonra yazarlar ağırlıklarına göre büyükten küçüğe sıralanarak liste olarak geri döndürülmüştür.
- **İster 3: BST (Binary Search Tree) Oluşturma** İkili Arama Ağacını (BST) oluşturmak ve silme işlemini yapmak için kendi veri yapımızı oluşturduk. Bu adımda, ağaca eleman ekleme, eleman silme ve eleman listeleme fonksiyonları kullanılmıştır.
 - **Düğüm Ekleme ve Silme:** BST ağacına düğüm ekleme ve çıkarma metodları, ağacın dengesini ve sıralamayı koruyarak doğru bir şekilde çalışmasını sağlamıştır.

- **Özyinelemeli Yaklaşım:** Ekleme ve silme işlemlerini kolaylaştırmak için özyinelemeli bir yaklaşım kullanılmıştır.
- **İster 4: Tüm En Kısa Yolları Bulma** Bir yazardan diğer tüm yazarlara olan en kısa yolları bulmak için yine BFS algoritmasından faydalandık. Verilen yazar merkezli olarak diğer tüm yazarlara giden en kısa mesafeleri ve yolları bulduk. Bu adım için ayrıca bir tablo oluşturarak bu verileri kullanıcıya sunuyoruz.
 - **Kuyruk ve Ziyaret Takibi :** BFS algoritmasını uygularken, daha önce kullandığımız kuyruk ve ziyaret takibi sistemlerini kullandık.
 - **Mesafe ve Önceki Düğüm Takibi** Daha önceki iterde de olduğu gibi her düğümün başlangıç düğümüne olan mesafesini ve yol boyunca önceki düğümü saklamak için dictionary veri yapıları kullandık.
- **İster 5: İşbirliği Yapan Yazar Sayısını Hesaplama** Verilen bir yazarın işbirliği yaptığı yazarların sayısını bulduk. Bu adımı gerçekleştirirken sadece yazarın komşularını (graf üzerindeki bağlantıları) bulmamız yeterliydi.
- **İster 6: En Çok İşbirliği Yapan Yazarı Bulma** Tüm yazarlar arasında en çok işbirliği yapan yazarı bulmak için tüm düğümleri tek tek gezerek ve her yazar için işbirliği sayılarını hesaplayarak bir karşılaştırma yapıldı. Son olarak bu karşılaştırma sonucu en büyük değere sahip olan yazar tespit edilerek sonuç döndürüldü.
- **İster 7: En Uzun Yolu Bulma** Verilen bir yazar düğümünden başlayarak gidilebilecek en uzun yolu bulmak için DFS (Depth First Search) algoritması kullanılmıştır.
 - **Özyineleme:** DFS algoritması için özyinelemeli bir yaklaşım kullandık. Bu, kodun daha okunabilir ve yönetilebilir olmasını sağladı.
 - **Ziyaret Takibi:** Düğümleri gezerek bir kez gidilen düğümleri tekrar ziyaret etmemek için her düğümün ziyaret edilme durumu bir set veri yapısında tutulmuştur.

D. Arayüz (GUI) ve Görselleştirme

Arayüz için Dash kütüphanesi kullanılmıştır. Dash, Python ile web tabanlı interaktif uygulamalar geliştirmek için çok kullanışlı bir araçtır. Uygulamamızda şu özellikleri destekliyor:

- **Dash Layout:** Uygulamanın arayüzü, ‘html’ ve ‘dcc’ bileşenleri kullanılarak tanımlanmıştır. Bu bileşenler, başlıkları, input alanlarını ve butonları içerir.
- **Cytoscape Bileşeni:** ‘dash-cytoscape’ kütüphanesi ile grafiğin etkileşimli olarak gösterilmesini sağladı. Bu bileşen, düğümleri ve kenarları web sayfasında görüntüler.
- **Girdi Alma:** Yazarların adlarını veya ID’lerini girdi olarak alabilmek için ‘dcc.Input’ ve ‘html.Button’ bileşenleri kullandık.
- **Çıktı Gösterme:** İşlem sonuçlarını göstermek için ‘html.Div’ bileşenleri kullandık.

- **Görselleştirme Özellikleri:** ‘dash-cytoscape’ bileşenine sağlanan stylesheet ile düğümlerin görünümü (boyut ve renk) makale sayılarına göre ayarlanmıştır.

E. Veri İşleme:

Veri işleme aşamasında, ‘pandas’ kütüphanesi kullanılarak Excel dosyası okunmuş ve veriler temizlenmiştir. Temizleme aşamasında, yazar isimleri normalize edilmiş, her yazara benzersiz bir ID atanmış ve makale bilgileri saklanmıştır.

F. Graf Veri Yapısı:

Yazarlar arasındaki iş birliğini temsil etmek için kendi graf veri yapımızı oluşturduk. Bu yapı, düğümleri (yazarlar) ve kenarları (işbirlikleri) saklamak için sözlüklerden oluşur.

G. Arayüz (GUI) ve Görselleştirme:

Arayüz için Dash framework’ü, graf görselleştirmesi için ise ‘dash-cytoscape’ bileşeni kullanılmıştır. Arayüzde, yazarlar arası etkileşim gösterilmiş ve her bir istere kolayca erişilebilmesi sağlanmıştır.

IV. KAZANIMLAR

Bu proje, hem teknik hem de bireysel/ekip olarak birçok kazanım sağlamıştır. Bu süreçte aşağıdaki kazanımları elde ettik:

- **Veri İşleme ve Temizleme:** Veri setlerini okuma, temizleme ve manipüle etme yeteneği kazanılmıştır. Özellikle ‘pandas’ kütüphanesi kullanılarak veri ön işleme ve dönüştürme süreçleri tecrübe edilmiştir.
- **Graf Veri Yapılarını Anlama:** Yazarlar arası işbirliklerini temsil etmek için kendi graf veri yapımızı oluşturarak, graf teorisi ve ağ analizine giriş yapılmıştır. Bu sayede, düğüm ve kenarların nasıl modellendiği ve verinin nasıl saklandığı konusunda bilgi sahibi olunmuştur.
- **Algoritmaları Uygulama:** BFS, DFS ve BST gibi temel algoritmaların nasıl çalıştığı anlaşılmış ve bu algoritmaların kodları, kütüphanesiz bir şekilde, sıfırdan yazılmıştır. Bu sayede algoritmaların mantığını daha iyi anlamamız sağlanmıştır.
- **Web Tabanlı Arayüz Geliştirme:** Dash framework’ünü kullanarak, interaktif web uygulamaları oluşturma becerisi kazanılmıştır. Kullanıcıların isteklerle etkileşim kurmasını sağlayacak arayüzler oluşturularak web tabanlı uygulamalar geliştirme tecrübesi elde edilmiştir.
- **Etkileşimli Görselleştirme:** ‘dash-cytoscape’ bileşeni kullanılarak interaktif ağ grafikleri oluşturma ve bu grafikleri web arayüzüne entegre etme yeteneği geliştirilmiştir. Bu, verilerin daha anlamlı ve etkileyici bir şekilde sunulmasına olanak sağlamıştır.

V. YORUMLAR

Projeyi tamamlarken hem teknik hem de süreçle ilgili değerlendirmeler yaptık. Süreçte karşılaştığımız zorlukları ve bu zorluklara bulduğumuz çözümleri aşağıda açıkladık:

A. Karşılaşılan Zorluklar

- **Kütüphanesiz Veri Yapıları ve Algoritmaları Uygulamak:** Kendi veri yapılarımızı ve algoritmalarımızı oluştururken, mevcut kütüphanelerin sunduğu kolaylıkların ve optimizasyonların ne kadar önemli olduğunu fark ettik. Kütüphanesiz bir yapı kurmanın, hem karmaşıklığı artırdığını, hem de zaman ve performans konusunda kısıtlamalara neden olabileceğini gördük.
- **Etkileşimli Grafiklerde Performans Sorunları:** ‘dash-cytoscape’ ile büyük veri kümelerinde interaktif grafikleri oluştururken, performans sorunları yaşadık. Bu sorunları çözmek için farklı layout algoritmalarını denedik ve veriyi ön işleme tekniklerini kullandık.
- **Arayüz Geliştirme:** Kullanıcı odaklı bir arayüz oluşturmak ve tüm istelere erişimi kolaylaştırmak için zaman harcadık. Her bir isteği, farklı input alanları ile birlikte arayüzde göstermeye çalıştık ve bu da tasarımda bazı zorluklara neden oldu.

B. Proje Değerlendirmesi

Projemizi tamamladıktan sonra aşağıdaki değerlendirmeleri yaptık:

- **Genel Başarı:** Verilen proje kapsamını başarıyla tamamladığımızı düşünüyoruz. Graf veri yapılarını kullanarak yazarlar arasındaki işbirliklerini analiz etmeyi, temel algoritmaları uygulamayı ve etkileşimli bir web arayüzü oluşturmayı başardık.
- **Veri ve Algoritma Uygulaması:** Veri yapılarını ve algoritmaları kullanma yeteneğimiz önemli ölçüde gelişti. BFS, DFS ve BST algoritmalarını kütüphanesiz bir şekilde uygulamak, algoritmaların mantığını daha iyi anlamamızı sağladı.
- **Görselleştirme ve Arayüz:** ‘dash-cytoscape’ ile etkileşimli grafikler oluşturarak, veriyi anlamlı bir şekilde görselleştirmeyi başardık. Bu da kullanıcıların analizlere daha kolay bir şekilde ulaşmasını sağladı.

VI. SONUÇ

Bu proje, akademik bir veri seti kullanarak yazarlar arasındaki iş birliği ilişkilerini modellemeyi ve çeşitli veri yapısı ile algoritma bilgilerini pratik uygulamalarla pekiştirmeyi amaçlamıştır. Proje sonunda ortaya çıkan interaktif web arayüzü, kullanıcıların veriyi keşfetmesini ve analiz etmesini kolaylaştırır.

VII. KAYNAKÇA

- **Overleaf Platformu:** IEEE raporu hazırlanırken kullanılan LaTeX düzenleyici.
<https://www.overleaf.com/>
- **Dash Documentation:** Dash kütüphanesi için resmi dokümantasyon.
<https://dash.plotly.com/>
- **Cytoscape.js Documentation:** Cytoscape.js için resmi dokümantasyon.
<https://js.cytoscape.org/>

Yazar İşbirliği Ağı Analizi

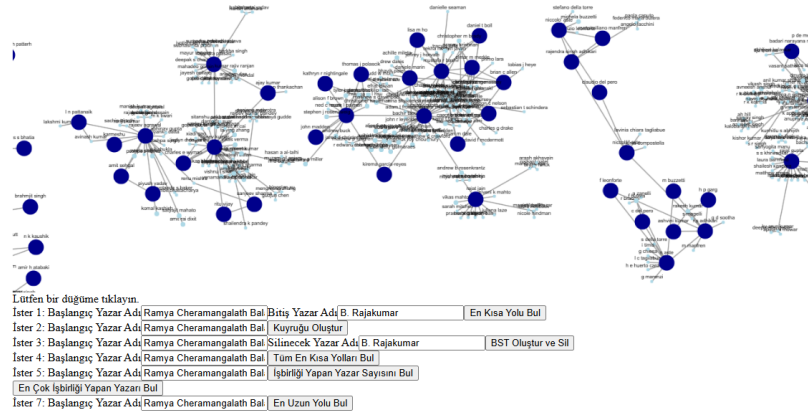


Fig. 1. Etkileşimli Grafik Görünümü

- **Pandas Documentation:** Pandas kütüphanesinin resmi dokümantasyonu.
<https://pandas.pydata.org/docs/>
<https://www.youtube.com/watch?v=H5Kv07648list> = <https://www.youtube.com/watch?v=PLh9ECzBB8tJPFTpuHKhYayis0H9pS6rIchatgpt.com>

