Avatar: Son Hava Bükücü Karakter ve Bölüm Bazlı Kümeleme

Özge Işıklar
Yapay Zeka Mühendisliği
Tobb Ekonomi ve Teknoloji Üniversitesi
Ankara, Türkiye
ozgeisklar@gmail.com

***Abstract*—Bu projede, popüler animasyon dizisi Avatar: Son Hava Bükücü'deki karakter diyaloglarını çeşitli makine öğrenmesi teknikleri kullanarak analiz etme ve kümelendirme amaçlandı. Bazı önemli karakterlerin diyaloglarına odaklanarak, metinlerin tematik benzerliklerine göre gruplandırılması için farklı kümeleme algoritmaları uygulandı. Metin verileri Universal Sentence Encoder ve TF-IDF vektörleştirme yöntemi ile işlendi, ardından PCA kullanılarak boyut indirgeme yapıldı. Daha sonra KMeans, Spectral ve Hiyerarşik Kümeleme (Agglomerative Clustering) yöntemleri kullanılarak karakterlerin diyalogları arasında belirgin gruplar tespit edilmeye çalışıldı.**

Kümeleme algoritmalarının performansı, Silhouette Skoru, Davies-Bouldin İndeksi ve Calinski-Harabasz İndeksi gibi içsel metrikler kullanılarak değerlendirildi. Ayrıca, kümeleme sonuçları, bilinen karakter gruplamalarını temsil eden bir önceden tanımlanmış "ground truth" ile karşılaştırılarak, Adjusted Rand Index (ARI) ve Normalized Mutual Information (NMI) metrikleri ile analiz edildi.

Bulgular, her üç kümeleme algoritmasının da anlamlı gruplamalar sağladığını, ancak KMeans algoritmasının, genel performans açısından diğerlerine kıyasla biraz daha iyi sonuçlar verdiğini göstermektedir. Proje, karakter diyaloglarının tematik analizinde kullanılabilecek farklı makine öğrenmesi yaklaşımlarının etkinliğini ortaya koymuştur.

I. GİRİŞ

Farklı alanlarda veri miktarının hızla artması; büyük metin verilerini analiz etmek, yorumlamak ve metin verileri üstünde veri madenciliği yapmak için etkili yöntemlere olan ihtiyacı artırmıştır. Özellikle televizyon dizileri ve filmler gibi eğlence sektöründe, senaryolar içindeki tematik ve karakter odaklı yapıları anlamak, hikaye anlatımı, karakter gelişimi ve izleyici etkileşimi açısından değerli içgörüler sağlayabilir. Bu proje, Avatar: Son Hava Bükücü adlı ünlü animasyon dizisindeki karakter diyaloglarını kümeleyerek, önemli karakterler arasındaki konuşma içeriklerine dayalı desenleri ve ilişkileri ortaya çıkarmayı amaçlamaktadır.

Avatar: Son Hava Bükücü, karmaşık karakterleri, zengin dünyası ve detaylı anlatımıyla tanınan bir dizidir. Dizi, her biri kendine özgü kişiliklere ve hikayelere sahip çeşitli karakterler içermektedir ve bu karakterlerin her biri, genel anlatıya katkıda bulunmaktadır. Bu projede, bu karakterlerin diyaloglarını analiz ederek, tematik benzerlikleri ve farklılıkları belirlemeyi ve bu temalar doğrultusunda karakterleri ve bölümleri kümelere ayırmak amaçlandı. Bu kümeler, karakterlerin anlatı içindeki rollerini ve birbirleriyle olan ilişkilerini yansıtırken aynı zamanda bölümlerin birbirine olan benzerliğine ayna tutar.

Bu amaçla, metin vektörleştirme, boyut indirgeme ve kümeleme gibi çeşitli makine öğrenmesi tekniklerinden yararlanıldı. Bu süreçte, hem metinlerin içeriğini anlamak hem de karakterler ve bölümler arasındaki ilişkileri daha iyi kavrayabilmek için uygun algoritmalar kullanılarak karakterlerin diyalogları incelendi. Bölüm ve kitap bazlı IMDB analizi, karakterlerin en çok kullandığı kelimeler ve karakter etkileşimleri görselleştirilerek kümeleme sonuçları desteklendirildi. Bu çalışmanın sonuçları, karakter diyaloglarının tematik analizi için kullanılabilecek etkili yöntemler sunarak, dizi analizinde yeni bakış açıları kazandırmayı hedeflemektedir.

Kümeleme sonuçları çeşitli kümeleme algoritmaları metrikleri (Silhoutte Score, Dunn İndex, NMI gibi) kullanılarak kıyaslanıp en iyi performans gösteren kümeleme algoritması çalışmanın ana metodolojisi olarak seçildi.

II. LITERATÜR ARAŞTIRMASI

- Text Clustering Algorithms: A Review [2] makalesi çeşitli metin öbekleme tekniklerini karşılaştırıp avantajlarını ve dezavantajlarını ifade etmiştir. Hiyerarşik algoritmaları, K-Means gibi partitioning based algoritmaları ve DBSCAN gibi denstiy based algoritmaları ele alıp açıklayıcı bir kıyaslama yapmıştır. Fakat daha gelişmiş GMM, Spectral Clustering gibi tekniklerden bahsetmemiştir ve çeşitli kümeleme metrikleri ile performans ölçümü yapmamıştır.
- Text clustering with LLM embeddings [1] makalesi birçok farklı embedding ve kümeleme metotlarını kullanıp hem derin öğrenme tekniklerini hem de klasik teknikleri kombine etmiştir. Ana kümeleme tekniklerden hiyerarşik kümelemeden bahsetmemesi bu makalenin eksiklerinden biridir. Hem ground truth kullanılan hem de kullanılmayan metriklerle performans ölçümü performans ölçümü yapması çalışmanın güvenilirliğini arttırmıştır.
- Visualization of Text Document Corpus [3] makalesi metin verilerinden corpus oluşturup bu veriler üstünde boyut indirgeme teknikleri kullanarak metinleri görselleştirme hakkında çalışmalarda bulunmuştur. PCA gibi state of the art teknikler kullanarak vektörize

edilmiş metinleri daha düşük bir boyuta taşımayı anlatmıştır.

- Toward plot de-interlacing in TV series [10] using scenes clustering makalesi hikaye örgüsündeki alt hikayeleri kullanarak TV dizileri bölümleri arasında bir kümeleme yapmıştır. Sahnelerle kümeleme yapmak için kullandığı yaklaşımlar bu konu üstünde farklı fikirler ortaya atmıştır. Agglomerative ve graph based olmak üzere iki farklı teknik kullanmıştır. Agglomerative kümeleme yaparken farklı link metotlarıyla daha kapsayıcı bir çalışma ortaya çıkarmıştır.

III. Veri Seti Tanımları

A. Veri Kaynağı:

https://www.kaggle.com/datasets/ekrembayar/avatar-the-last-air-bender/data

B. Sütun Özellikleri

Veri seti 11 sütun ve 13,736 satırdan oluşmaktadır. Sütunlar: ”id”, ”book”, ”book_num”, ”chapter”, ”chapter_num”, ”character”, ”full_text”, ”character_words”, ”writer”, ”director”, ”imdb_rating” şeklindedir. ”book” sütunu sezon ismini tutarken ”chapter” sütunu bölüm ismini tutar. ”full_text” sütunu karakterlerin sözlerini ve sahne betimlemelerini içerirken ”character_words” sütunu direkt olarak karakterlerin replik-lerini tutar. ”imdb_rating” sütunu o bölümün IMDB rating bilgisini tutar ve bu veri 0-10 arasında float değerler alabilir.

C. Ön İşleme

Çalışma metin verisi ile yürütüldüğü için ön işleme aşamasında corpus oluşturulukren ”nltk” kütüphanesi kullanarak ’stopwords’ kelimeleri atılıp verinin tamamı küçük harfe çevrilmiştir. Yalnızca ”imdb_ratings” sütunu numerik değerlere sahiptir ve 0-10 arası değerler aldığı için ön işlemeye gerek duyulmamıştır.

D. Öznitelik Seçimi ve Düzenlenmesi

Bölüm ve karakter bazında kümeleme yapabilmek için karakterlerin replikleri esas alınmıştır. Sahne betimlemeleri bir katkı sağlamadığı için ”character_words” sütunu kullanılmıştır. Alınan verileri vektörleştirme aşamasında Universal Sentence Encoder ve TF-IDF Vectorizer olmak üzere iki metot kullanılmıştır ve embedding yapılan veriler PCA algoritması ile 2 boyuta indirgenmiştir.

E. Veri Görselleştirme

1) IMDB Rating Analizi: IMDB Rating sonçları bölüm, sezon ve yönetmen bazlı analiz edilmiştir.

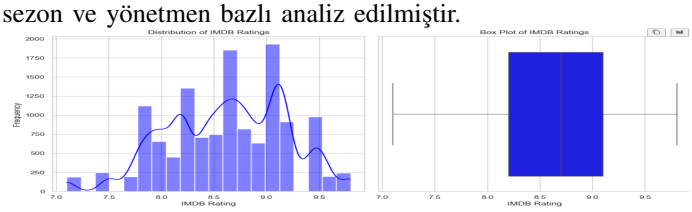

Fig. 1. IMDB Rating Plotları

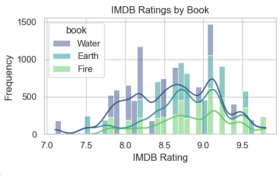

Fig. 2. Sezon Bazında IMDB Ratingleri

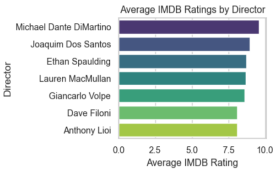

Fig. 3. Yönetmen Bazında IMDB Ratingleri

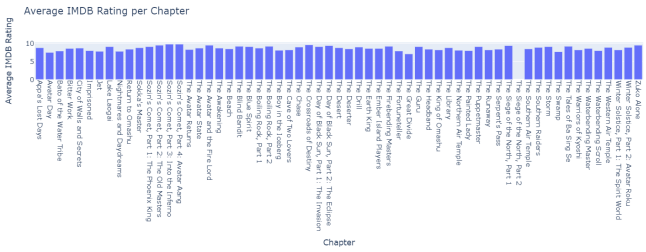

Fig. 4. Bölüm Bazında IMDB Ratingleri

2) Word Clouds: Önemli karakterler filtrelenerek (Aang,Katara,Sokka,Toph,Zuko) bu karakterlerin en çok hangi kelimeleri söylediği word cloud ile temsil edilmiştir ve sentiment analizleri yapılmıştır. Bu görseller karakterlerin birbirine yakınlığını anlayabilmek için en çok söyledikleri kelimeleri ve duygu durumlarını ifade eder.

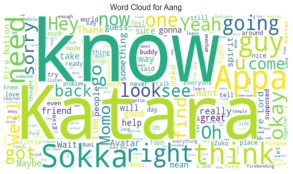

Fig. 5. Aang Word Cloud

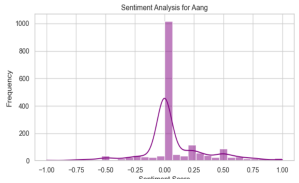

Fig. 6. Aang Sentiment Analysis

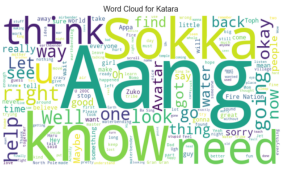

Fig. 7. Katara Word Cloud

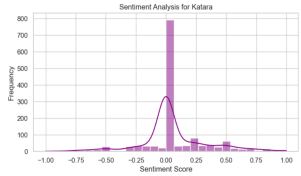

Fig. 8. Katara Sentiment Analysis

Fig. 9. Sokka Word Cloud

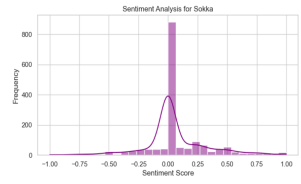

Fig. 10. Sokka Sentiment Analysis

Fig. 11. Toph Word Cloud

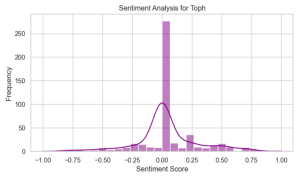

Fig. 12. Toph Sentiment Analysis

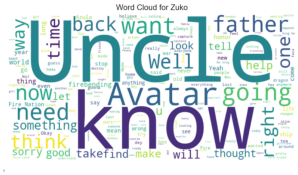

Fig. 13. Zuko Word Cloud

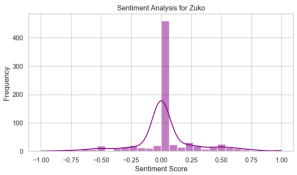

Fig. 14. Zuko Sentiment Analysis

3) Karakter Etkileşimleri: Karakterlerin aynı bölümlerde beraber görülme oranlarına göre bir matris çıkartılıp bu etkileşimlere göre graph yapısı oluşturulmuştur.

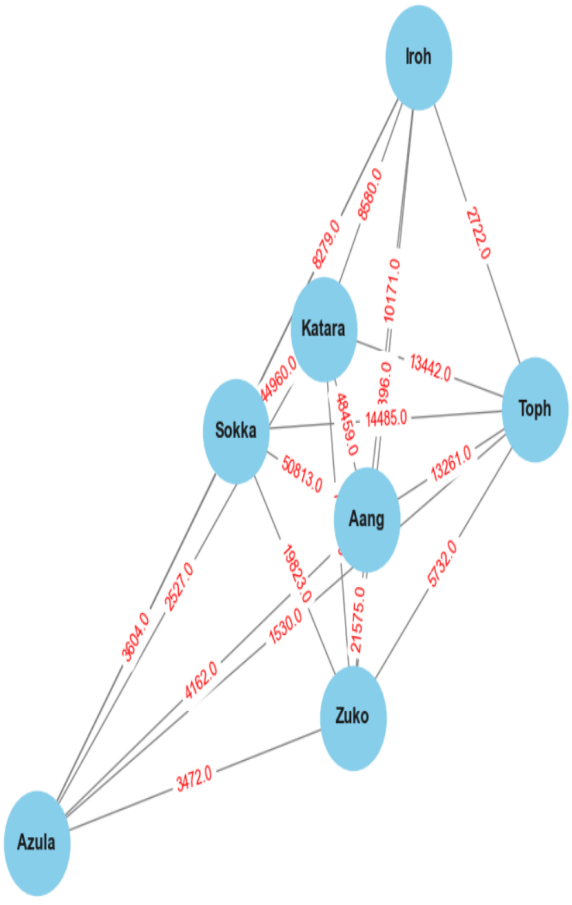

Fig. 15. Karakter Etkileşimleri

IV. Kullanılan Metodoloji

Bu çalışmada, Doğal Dil İşleme (NLP), Makine Öğrenimi ve Veri Madenciliği tekniklerini kullanarak bir metin kümesinin ve karakterlerin özelliklerine dayalı olarak kümelenmesi hedeflenmiştir. Kümelenen veriler, dizinin bölümleri ve önemli görülen karakterlerinden oluşmaktadır. Çalışmada kullanılan metodoloji aşağıda detaylandırılmıştır.

A. Kelime Gömme (Vektörleştirme) ve Boyut İndirgeme

Metinleri vektörleştirme işlemleri için 2 farklı yaklaşım kullanılıp her iki yaklaşımın da faydalarından doğru alanlarda yararlanılmak istenmiştir.

1) Universal Sentence Encoder: USE modeli Transformer ve Deep Averaging Network (DAN) olmak üzere iki kodlayıcı temel alınarak geliştirilmiştir.Bu modellerin her ikisi de bir kelimeyi veya cümleyi girdi olarak alır ve embedding yapar. Modeller, cümleleri girdi olarak alır, bunları tokenleştirir ve her cümleyi 512 boyutlu bir vektöre dönüştürür.Bu çalışmada metinleri vektörize etmek için Universal Sentence Encoder (USE) kullanılmıştır. USE metinleri anlamsal olarak benzerliklerini daha iyi temsil eder ve bu metinlerin makine öğrenimi modelleri için uygun hale getirilmesine yardımcı olur [4].Aynı motivasyonla bu çalışmada da bölüm bazlı kümeleme yapılırken kullanılacak vektör bütün bölüme reprezente edeceği için anlamsal olarak iyi performans gösteren USE metodu kullanılmıştır.

2) TF-IDF Vectorizer: TF-IDF (Term Frequency-Inverse Document Frequency), metin verisini sayısal verilere dönüştürmek için kullanılan bir yöntemdir. TF-IDF, bir kelimenin bir belgede ne kadar önemli olduğunu belirlemek için iki temel ölçüt kullanır. TF (Term Frequency) bir kelimenin belirli bir belgede ne kadar sık geçtiğini ölçer. Daha sık geçen kelimeler daha yüksek TF değerine sahip olur. IDF (Inverse Document Frequency) ise bir kelimenin tüm belgeler arasında ne kadar nadir olduğunu ölçer. Bir kelime ne kadar az belgede geçiyorsa, IDF değeri o kadar yüksek olur. Bu, yaygın kelimelerin önemini azaltır ve nadir kelimelere daha fazla ağırlık verir. TF-IDF, TF ve IDF değerlerini çarparak, bir kelimenin o belgede ne kadar önemli olduğunu belirler. Sonuç olarak, yaygın olan kelimeler (örneğin ”and”, ”the”, ”is” gibi stopwords) düşük bir ağırlık alırken, belirli bir belgeye özgü olan kelimeler daha yüksek bir ağırlık alır. [5] Bu yaklaşımdan yola çıkarak projenin karakter bazlı kümeleme kısmında TF-IDF vectorizer kullanılarak belirli karaktere özgü kelimelere yüksek ağırlık vererek karakterlerin eşsiz özniteliklerinin daha iyi ifade edilmesi amaçlanmıştır.

3) Principal Component Analysis (PCA): PCA (Ana Bileşen Analizi), yüksek boyutlu verilerin daha düşük boyutlu bir uzaya projeksiyonunu sağlayan istatistiksel bir tekniktir. PCA, verilerdeki temel varyasyon kaynaklarını belirleyerek bu varyasyonları temsil eden yeni, daha az sayıda bileşen oluşturur. Bu yeni bileşenler, orijinal verinin bilgilerini korurken, verilerin boyutunu azaltır. Bazı makalelerde kümeleme algoritmalarının daha iyi çalışabilmesi için verilerin düşük boyutlu bir uzaya indirgenmesi faydalı olabileceği ifade edilmiştir. [6] PCA ile indirgenmiş veriler, kümeler arasındaki farkların daha net görülmesine ve daha başarılı kümeleme sonuçlarının elde edilmesine olanak tanır.

B. Kümeleme Algoritmaları

Kümeleme; veri analizi, veri madenciliği ve makine öğreniminde önemli bir yere sahip olan bir tekniktir. Kümeleme, veri setinde doğal olarak var olan alt grupları veya kümeleri belirlemeyi amaçlar. Kümeleme analizi, veriyi sınıflandırma veya gruplama yoluyla, veri setindeki gizli yapıları keşfetmeye ve bu yapıları anlamlandırmaya yardımcı olurken veri noktaları arasındaki benzerliği de öçlmemize olanak sağlar. Kümeleme algoritmalarındaki önemli zorluklardan biri verinin kaç gruba ayrılacağını yani k değerini belirlemektir. Bu zorluğun önüne geçebilmek adına Elbow ve Silouthhe Score gibi bazı metotlar vardır.

Elbow Methodu, kümeler arasındaki toplam içsel varyans (SSE - Sum of Squared Errors) ile küme sayısı (k) arasındaki ilişkiyi değerlendirir. SSE, her bir veri noktasının kendi küme merkeziyle olan uzaklığının karesinin toplamını ifade eder. Bu değer, küme içindeki veri noktalarının ne kadar homojen olduğunu gösterir. Küme sayısı arttıkça, her bir kümenin içsel varyansı azalır, çünkü kümeler daha küçük ve homojen hale gelir. Ancak, küme sayısı çok arttığında, varyans azalması çok küçük olur ve eklenen yeni kümelerin faydası azalır. Aynı zamanda Silhoutte Score değeri farklı k değerleri için hesaplanıp en yüksek SS değerine sahip olan k değeri de seçilebilir. Bu çalışmada her iki yöntem de kullanılmıştır ve hem bölüm bazlı hem karakter bazlı küme sayısı 3 olarak belirlenmiştir.

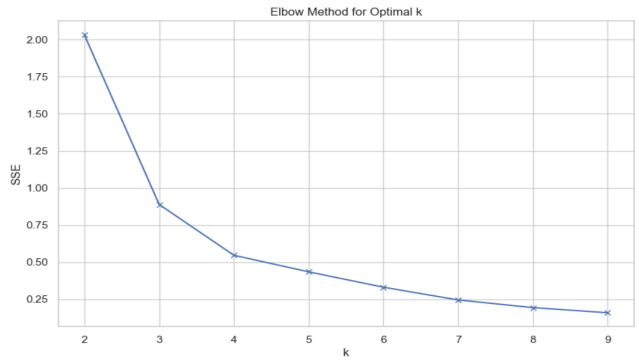

Fig. 16. Bölüm Kümemelemesi için Elbow

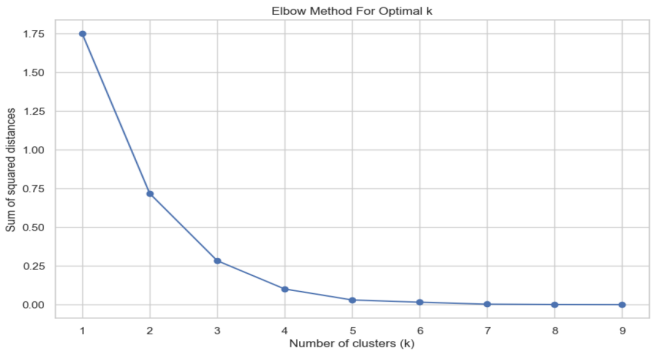

Fig. 17. Karakter Kümemelemesi için Elbow

Küme sayısına karar verildikten sonra K-Means, Agglomerative ve Spectral olmak üzere 3 farklı kümeleme algoritması kullanılmıştır.

1) K-Means Kümeleme Algoritması: K-Means Kümeleme, denetimsiz öğrenme algoritmalarından biridir ve veri setini belirli sayıda (k) kümeye ayırmayı amaçlar. Her küme, bir küme merkezi veya "centroid" etrafında şekillenir ve veri noktaları, en yakın küme merkezine atanır. K-Means algoritması, belirlenen küme sayısına göre veri noktalarını gruplar ve her grubun içinde veri noktaları birbirine daha yakın, diğer gruplardan ise daha uzak olmalıdır.Algoritma, başlangıçta veri kümesinden k adet rastgele küme merkezi seçer. Bu merkezler, başlangıç kümeleri için referans noktası olarak kullanılır.Her veri noktası, en yakın küme merkezine atanır. Bu yakınlık, genellikle Öklidyen mesafe ile ölçülür.Her bir kümenin yeni merkezi, o kümeye atanmış veri noktalarının ortalaması alınarak hesaplanır. Yani, her kümenin merkez noktası, o kümeye en yakın noktalar tarafından güncellenir.Veri noktaları, güncellenmiş merkezlere göre yeniden atanır. Bu adımlar, merkezler değişmez hale gelene kadar tekrarlanır.Küme merkezleri stabil hale geldiğinde ve veri noktaları artık farklı küme merkezlerine atanmadığında, algoritma durur.Bu sürecin sonunda, veri seti k sayıda kümeye ayrılmış olur ve her küme, kendi merkezi etrafında toplanmış veri noktalarından oluşur.

Bu projede, metin tabanlı veriler (karakter konuşmaları ve hikaye bölümleri gibi) kümelere ayrılmaktadır. K-Means, bu metinleri vektör uzayında anlamlı gruplara ayırma yeteneğine sahiptir. Her küme, benzer temalara veya konulara sahip metinlerden oluşabilir, bu da metinlerin anlamsal yapısını ortaya çıkarmaya yardımcı olur. Aysun Güran, Murat Can Ganiz, et.al [7] tarafından yazılan makalede NMF ile boyut indirgeme yapıldıktan sonra optimal k ile K-Means algoritmasının diğer algoritmalara göre daha iyi performans gösterdiği ifade edilmiştir. Bu yaklaşımdan da yola çıkılarak seçilen kümeleme algoritmalarından biri K-Means olmuştur.

2) Agglomerative Kümeleme Algoritması: Agglomerative kümeleme, hiyerarşik kümeleme yöntemlerinden biridir ve alt-birleştirici (bottom-up) bir yaklaşımla çalışır. Bu yöntem, başlangıçta her bir veri noktasını kendi başına bir küme olarak kabul eder ve bu kümeleri adım adım birleştirerek daha büyük kümeler oluşturur. Süreç, tüm veri noktaları tek bir küme haline gelene kadar devam eder veya belirli bir küme sayısına ulaşılana kadar durur. Youjin Rong ve Yi'an Liu et. al[8] makalelerinde K-Means ve Agglomerative kümelemeyi birleştirerek bir çalışmada bulunmuşlardır ve agglomerative kümelemenin birleştirme işlemlerinde yüksek doğruluk gösterdiğini ifade etmişlerdir.

Bu projede, metin tabanlı veriler (karakter konuşmaları, hikaye bölümleri) üzerinde çalışılırken, veri setinde karmaşık ve gizli yapılar bulunabilir. Agglomerative kümeleme, veri setinin hiyerarşik yapısını ortaya çıkararak, bu tür karmaşık ilişkileri daha iyi keşfetmeye yardımcı olmuştur.

3) Spectral Kümeleme Algoritması: Spektral Kümeleme, veri noktaları arasındaki bağlantıları bir grafik (graph) yapısı olarak ele alan ve bu grafın spektral özelliklerini kullanarak kümeler oluşturan bir kümeleme yöntemidir. Spektral

Kümeleme, geleneksel yöntemlerin zorlandığı karmaşık veri yapılarında ve şekillerde etkili olabilir.Veri noktaları, bir grafın düğümleri olarak ele alınır ve aralarındaki mesafeye dayalı bağlantılar belirlenir. Genellikle, komşuluk grafiği veya k-en yakın komşu grafiği kullanılır.Düğüm çiftleri arasındaki mesafelere göre bir ağırlık matrisi (affinity matrix) oluşturulur. Bu matris, düğümler arasındaki benzerlikleri gösterir.Grafın komşuluk yapısını temsil eden bir Laplace matrisi oluşturulur. Bu matris, düğümler arasındaki bağlantıları optimize etmek için kullanılır.Laplace matrisine eigendecomposition uygulanarak, matrisin özvektörleri ve özdeğerleri hesaplanır. Bu işlem, veriyi alt uzayda yeniden düzenler.Hesaplanan özvektörlerle veri noktaları yeni bir alt uzayda temsil edilir. Bu uzayda kümeler daha belirgin hale gelir.Son olarak, bu yeni uzayda veri noktalarına geleneksel bir kümeleme algoritması (genellikle K-Means) uygulanarak nihai kümeler oluşturulur. R. Janani ve Dr. S. Vijayarani et.al [9] Spectral kümeleme yönteminin özellikle kümelerin dairesel olmadığı ve lineer ayrımın zor olduğu durumlarda kullanılabileceğini göstermişlerdir.

Bu projede ,vektörleştirme yöntemlerinden (TF-IDF ve USE) elde edilen vektörler uzayda lineer bir şekilde ayrılamayabileceği için Spectral kümeleme yönteminden faydalanılmıştır. Bu şekilde veri noktaları arasındaki bağlantılar ve graf yapısı dikkate alınıp metinlerin doğal yapısını daha iyi anlamak ve anlamlı kümeler oluşturmak amaçlanmıştır.

V. Test Sonuçları ve Tartışma

Seçilen K-Means, Agglomerative ve Spectral kümeleme algoritmaları sonucunda çıkan kümeler ”plotly” kütüphanesi kullanılarak görselleştirilmiştir. Bu sayede herhangi bir bölümün hangi bölümlerle aynı kümede olduğu daha anlaşılır bir şekilde gösterilebilmiştir.

- Bölüm Bazlı Kümeleme Görselleri

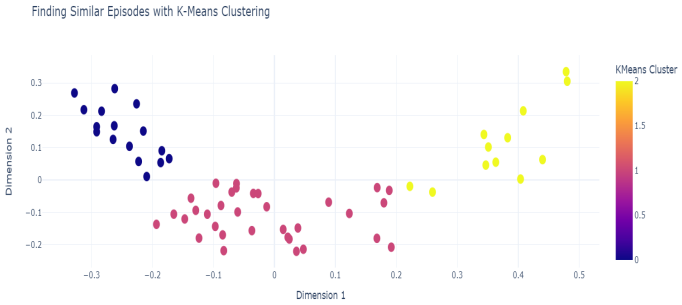

Fig. 18. K-Means ile Bölüm Kümemelemesi

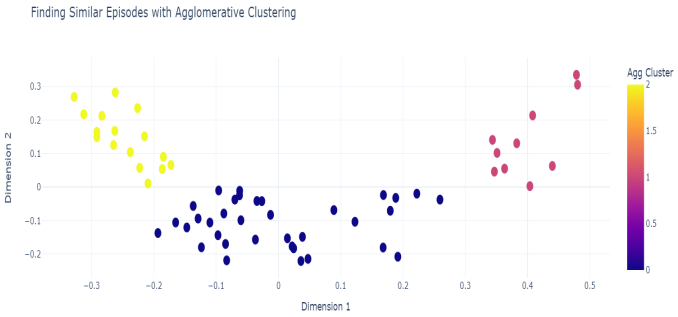

Fig. 19. Agglomerative ile Bölüm Kümemelemesi

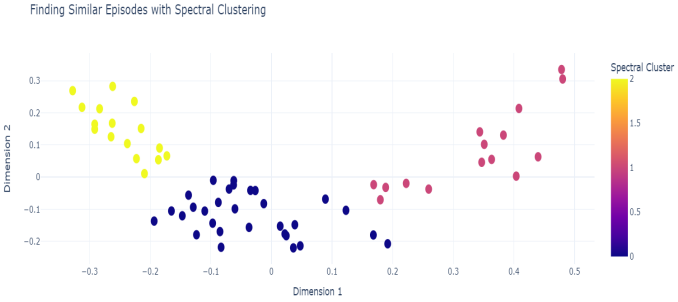

Fig. 20. Spectral ile Bölüm Kümemelemesi

- Karakter Bazlı Kümeleme Görselleri

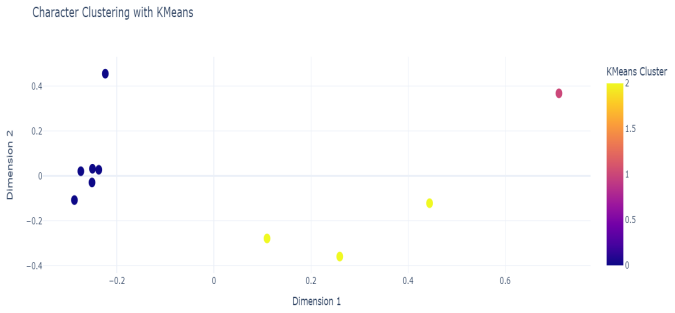

Fig. 21. K-Means ile Karakter Kümemelemesi

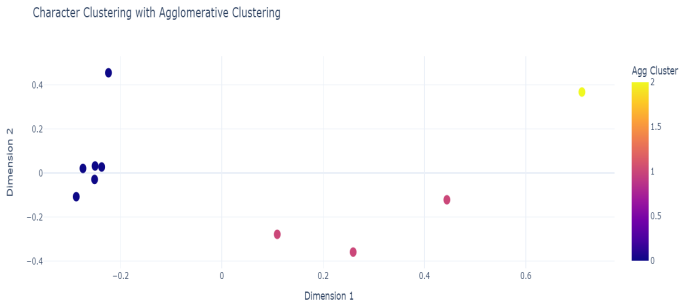

Fig. 22. Agglomerative ile Karakter Kümemelemesi

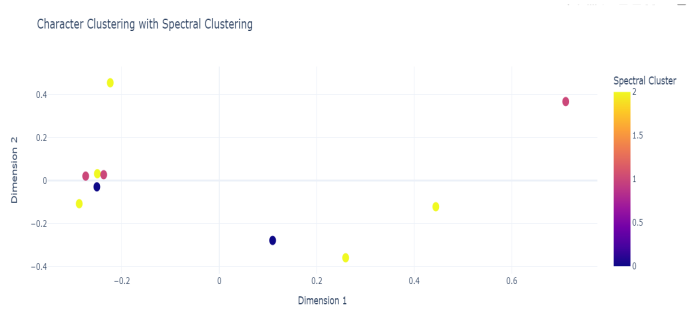

Fig. 23. Spectral ile Karakter Kümemelemesi

A. *Performans Metrikleri ile Ölçüm*

Performans ölçümü her iki başlık için ayrı metriklerle yapılmıştır. Bölüm bazlı kümeleme için ”ground truth” değeri belirlenemeyeceği için veri noktalarının kendi kümeleri içinde ne kadar iyi yerleştiğini ve diğer kümelerden ne kadar iyi ayrıldığını ölçen Silhoutte Score, kümeler arasındaki benzerliği ve kümelerin içsel tutarlılığını ölçen Dunn Index ve kümeler arasındaki mesafenin, kümeler içindeki mesafeye oranını hesaplayan Calinski-Harabasz Index metrikleri kullanılmıştır. Karakter bazlı kümeleme için ise ”ground truth” değeri yazar tarafından (Ozai,Azula),(Zuko,Iroh),(Aang,Katara,Toph,Sokka),(Suki,Jet) olarak belirlenmiştir. Bu ”ground truth” değerine göre NMI ve ARI metrikleriyle performans ölçümü yapılmıştır.

Metrik	K-Means	Agglomerative	Spectral
Silhouette Score	0.54853	0.540326	0.551405
Davies-Bouldin Index	0.550505	0.553673	0.562259
Calinski-Harabasz Index	116.353819	108.854291	115.399292

TABLE I
BÖLÜMLER İÇIN KÜMELEME ALGORITMALARININ PERFORMANS METRIKLERI

Metrik	Spectral	K-Means	Agglomerative
Adjusted Rand Index (ARI)	-0.091954	0.444444	0.444444
Normalized Mutual Information (NMI)	0.284953	0.680963	0.680963

TABLE II
KARAKTERLER İÇIN KÜMELEME ALGORITMALARININ PERFORMANS METRIKLERI

Silhoutte Score ve Calinski-Harabasz Index değerlerinin daha yüksek olması, Dunn Index değerinin ise daha düşük olması kümelemenin daha doğru sonuç verdiğini gösterir. ARI ve NMI değerlerinde ise daha büyük değer kümelerin ”ground truth” ile daha yakın olduğunu gösterir.

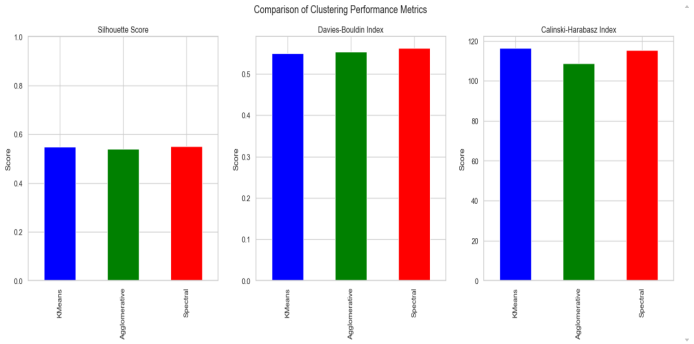

Fig. 24. Bölüm Kümelemesi Ölçümü

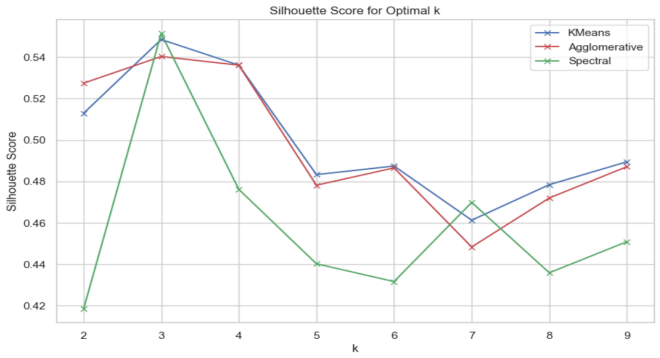

Fig. 25. Silhoutte Score değerleri

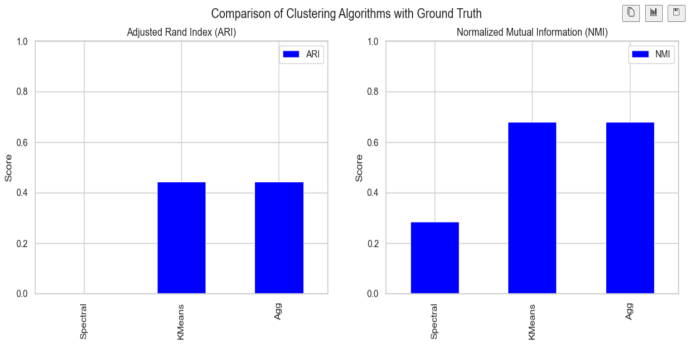

Fig. 26. Karakter Kümelemesi Ölçümü

B. Tartışma

1) Bölüm Kümelemesi için Tartışma: Performans metrikleri değerlerine bakılırsa K-Means algoritması hem Dunn Index hem de Calinski-Harabasz Index açısından en iyi perfromansı göstermiştir. Spectral kümeleme algortimasının ise Silhoutte Score değeri en yüksektir. Genel olarak 3 algortima için değerler yakın olsa da K-Means iki metrikte daha iyi sonuç vermiştir ve en iyi algoritma olarak seçilmiştir.

2) Karakter Kümelemesi için Tartışma: Karakter kümelemesinde K-Means ve Agglomerative algoritmaları aynı ve en iyi performansı göstermiştir. Kümelenecek veri sayısı az olduğu için Spectral kümeleme algoritması oldukça kötü bir performans sergilemiştir.

VI. Sonuç

Bu çalışmada, doğal dil işleme ve veri madenciliği tekniklerini kullanarak, metin tabanlı verilerin kümeleme algoritmaları ile sınıflandırılmasını hedeflenmiştir. Çalışmada farklı kümeleme algoritmaları (K-Means, Agglomerative Clustering, Spectral Clustering) uygulanarak, karakter konuşmaları ve hikaye bölümleri gibi metin verileri anlamlı gruplara ayrılmaya çalışıldı. Bu süreçte, veriler ön işleme tabi tutuldu, boyutları azaltıldı ve çeşitli performans metrikleri kullanılarak kümeleme algoritmaları değerlendirildi. Sonuç olarak K-Means kümeleme algoritmasının metin verilerinde diğer modellere göre daha iyi bir performans gösterdiği sonucuna varıldı.

A. Öğrenilenler

Farklı kümeleme algoritmalarının, metin tabanlı veriler üzerinde nasıl performans gösterdiği anlaşıldı. Özellikle K-Means algoritmasının, genel kümeleme performansı açısından güçlü bir seçenek olduğu ve Silhouette Score gibi metriklerde Spektral Kümelemenin öne çıktığı gözlemlendi.

B. Kazanç

Çeşitli kümeleme algoritmaları karşılaştırılarak, veri setleri üzerinde en uygun yöntemi seçebilme yeteneği geliştirildi. Bu, özellikle doğal dil işleme ve veri madenciliği projelerinde doğru algoritma seçiminin ne kadar kritik olduğunu gösterdi.

C. Katkı

Bu çalışma, benzer veri yapıları üzerinde çalışan araştırmacılara, farklı kümeleme algoritmalarının performansını değerlendirmek için bir referans olabilir. Ayrıca, metin verilerinin sınıflandırılması ve analiz edilmesi için kullanılabilecek bir yol haritası sunuldu.

D. Karşılaşılan Zorluklar

Çalışmada metin verilerinin doğru vektörize edilmesi ve doğru kümeleme algoritmasının seçilmesi oldukça zorlu bir süreç olmuştur. Algoritmaların kıyaslanması ve performansları metin verisi olduğu için yeterince iyi sonuç gösterememiştir.

E. Gelecek Çalışmalar

Bundan sonraki çalışmalarda farklı vektörleştirme metotları denenip derin öğrenme tabanlı modeller ile kümeleme yapılabilir. Vektörlerin bölümleri ve karakterleri farklı derin öğrenme tabanlı tekniklerle reprezente etmesi sağlanıp daha doğru gruplamalar sağlanabilir.

References

[1] Petukhova, A., Matos-Carvalho, J. P., Fachada, N. (2024). Text clustering with LLM embeddings. arXiv preprint arXiv:2403.15112.

[2] Suyal, H., Panwar, A., Negi, A. S. (2014). Text clustering algorithms: a review. International Journal of Computer Applications, 96(24). Citeseer.

[3] Fortuna, B., Grobelnik, M., Mladenic, D. (2005). Visualization of text document corpus. Informatica, 29(4).

[4] Pramanik, A., Das, A. K., Pelusi, D., Nayak, J. (2023). An Effective Fuzzy Clustering of Crime Reports Embedded by a Universal Sentence Encoder Model. Mathematics, 11(3), 611. MDPI.

[5] Bafna, P., Pramod, D., Vaidya, A. (2016). Document clustering: TF-IDF approach. In 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) (pp. 61-66). IEEE.

[6] Mekala, S., Rani, B. P. (2020). Kernel PCA based dimensionality reduction techniques for preprocessing of Telugu text documents for cluster analysis. International Journal of Advanced Research in Engineering and Technology, 11(11), 1337-1352.

[7] Gü ran, A., Ganiz, M. C., Naiboğlu, H. S., Kaptıkaçtı, H. O. (2013). NMF based dimension reduction methods for Turkish text clustering. In 2013 IEEE INISTA (pp. 1-5). IEEE.

[8] Rong, Y., Liu, Y. (2020). Staged text clustering algorithm based on K-means and hierarchical agglomeration clustering. In 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA) (pp. 124-127). IEEE.

[9] Janani, R., Vijayarani, S. (2019). Text document clustering using spectral clustering algorithm with particle swarm optimization. Expert Systems with Applications, 134, 192-200.

[10] Ercolessi, P., Sénac, C., Bredin, H. (2012). Toward plot de-interlacing in TV series using scenes clustering. In 2012 10th international workshop on content-based multimedia indexing (CBMI) (pp. 1-6). IEEE.