T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

FİNANS SEKTÖRÜNDE DOĞAL DİL İŞLEME (NLP) İLE RAPOR KÜMELENDİRME VE TALEP BAZLI RAPOR ÖNERİLERİ OLUŞTURMA

YÜKSEK LİSANS TEZİ

Seda AYDİN TUZCUAY

Enstitü Anabilim Dalı	**:**	**BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Dr. Öğr. Üyesi Tuğrul TAŞCI**

Haziran 2022

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

FİNANS SEKTÖRÜNDE DOĞAL DİL İŞLEME (NLP) İLE RAPOR KÜMELENDİRME VE TALEP BAZLI RAPOR ÖNERİLERİ OLUŞTURMA

YÜKSEK LİSANS TEZİ

Seda AYDİN TUZCUAY

Enstitü Anabilim Dalı : BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

Bu tez 29.06.2022 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Seda AYDİN TUZCUAY
09.05.2022

TEŞEKKÜR

Yüksek lisans eğitimim boyunca deneyimlerinden faydalandığım, desteğini esirgemeyen değerli hocam Dr. Öğr. Üyesi Tuğrul TAŞCI'ya, aynı ekipte çalıştığım, sektörel tecrübelerinden faydalandığım ekip arkadaşım Consulting Data Designer İsmail KILIÇ'a, eğitim hayatım boyunca beni cesaretlendiren, yanımda olan annem Sevgi'ye, babam Yurtdaş'a, ablam Eda'ya, kardeşim Yunus'a ve canım eşim Ömer'e teşekkürlerimi sunarım.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

EDW	: Enterprise Datawarehouse - Kurumsal Veri Ambarı
NLP	: Natural Language Processing - Doğal Dil İşleme
TF-IDF	: Terim Frekansı-Ters Doküman Frekansı
ETL	: Extract – Transform – Load / Çek – Yükle – Dönüştür
WCSS	: Within Clusters Sum of Square – Kümeler İçi Kareler Toplamı
TDA	: Tekil Değer Ayrışımı - Singular Value Decomposition (SVD)
KTDA	: Kesik Tekil Değer Ayrışımı – Truncated SVD
TBA	: Temel Bileşen Analizi - Principal Componenet Analysis (PCA)
S	: Silhoutte Katsayısı
DB	: Davies-Bouldin İndeksi
CH	: Calinski-Harabasz İndeksi

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Doğal dil işleme, TF-IDF, Word2vec, Doc2vec, kelime gömme, boyut indirgeme, kümeleme, kümeleme değerlendirme, doküman benzerliği

Yapay zeka alanında yer alan teknolojilerin hızla gelişmesi ile hem akademinin hem de çeşitli sektörlerin ilgileri bu alana çevrilmiştir. Bu bağlamda müşteri ve ürün çeşitliliği fazla olan sektörlerde, yapay zeka teknolojileri arasında yer alan, doğal dil işleme ve makine öğrenmesi çalışmaları oldukça artmıştır. Doğal dil işleme, doküman sınıflandırma - kümeleme, adlandırılmış varlık tanıma, duygu analizi, yazım denetleme, sohbet botları, dil çeviri vb. çalışma konularına sahiptir.

Bu çalışmada, finans sektöründe yer alan özel bir bankanın, rapor talep içerikleri kullanılarak derin öğrenme temelli doğal dil işleme ve makine öğrenmesi çalışması yapılmıştır. Çalışma kapsamında rapor talep içeriklerinin, TF-IDF, Word2vec ve Doc2vec kelime gömme yöntemleri ile temsilleri oluşturulmuş, makine öğrenmesi yöntemlerinden olan K-Means, K-Medoids ve Agglomerative kümeleme algoritmaları ile kümelenmiş ve aynı kümede yer alan birbirine en benzer rapor talep içerikleri adreslenmiştir. Çalışmada kümeleme başarıları Silhouette katsayısı, Calinski-Harabasz ve Davies-Bouldin indeksleri ile yorumlanmıştır. Çalışma ile yeni gelen bir rapor talebine benzeyen, daha önce yapılmış bir rapor talebi var ise bu talep & taleplerin adreslenmesi hedeflenmiştir.

Çalışma sonucunda, üç kelime gömme yönteminde de en iyi kümeleme sonucu K-Means ile elde edilmiştir. Word2vec ve Doc2vec ile yapılan kümelemede, değerlendirme metriklerinin benzer sonuçlar verdiği, benzerlik çalışmasında ise 3 kelime gömme yöntemi ile elde edilen sonuçların benzer olduğu görülmüştür.

REPORT CLUSTERING AND CREATING DEMAND-BASED REPORT RECOMMENDATIONS WITH NATURAL LANGUAGE PROCESSING (NLP) IN THE FINANCIAL INDUSTRY

SUMMARY

Keywords: Natural language processing, TF-IDF, Word2vec, Doc2vec, word embedding, dimensionality reduction, clustering, cluster validation, document similarity

With the rapid development of technologies in the field of artificial intelligence, the interests of both academia and various sectors have been turned to this field. In this context, natural language processing and machine learning studies, which are among the artificial intelligence technologies, have increased considerably in sectors with a wide variety of customers and products. Natural language processing, document classification - clustering, named entity recognition, sentiment analysis, spell checker, chatbots, language translation, etc. has study subjects.

In this study, a deep learning-based natural language processing and machine learning study was carried out by using the report request contents of a private bank in the financial sector. Within the scope of the study, the representations of the report request contents were created with TF-IDF, Word2vec and Doc2vec. K-Means, K-Medoids and Agglomerative machine learning clustering algorithms were clustered and report requests in the same cluster were listed. In the study, clustering successes were interpreted with Silhouette coefficient, Calinski-Harabasz and Davies-Bouldin indices. With the study, if there is a report request made similar to a new report request, it is aimed to list this report request or report requests.

As a result of the study, the best clustering result was obtained with K-Means in all word embedding methods. In the clustering with Word2vec and Doc2vec, it was seen that the evaluation metrics gave similar results, and in the similarity study, the results obtained with the three word embeddings method were similar.

BÖLÜM 1. GİRİŞ

Teknolojinin hızla gelişmesi ile hemen hemen her sektörün bilişim sistemlerine duyduğu ilgi artmıştır. Şirketlerin, rekabet üstünlüğü sağlayabilmek, var olan statüyü koruyabilmek, tercih edilebilir olabilmek, hizmet alanlarının başarılarını arttırabilmek, çalışan bağlılığı sağlayabilmek vb. sebepler ile teknolojiden faydalanması zorunlu hale gelmiştir. Bu bağlamda şirketlerin entegre bilişim sistemleri programlarına sahip olması, süreçlere bir noktadan müdahale edilebilir olması için kaçınılmaz hale gelmiştir.

Günümüzde şirketlerin, hizmet ettiği müşteriler tarafından tercih edilebilir olabilmelerine etki eden en önemli hususlardan biri de müşterilerin ihtiyaç duydukları durumlarda hızlı sorun çözen yapılara sahip olmalarıdır. Bu sebeple dış ihtiyaçlar kadar iç ihtiyaçlar da önem kazanmıştır. Şirket içi durağan yapıların var olması, manuel yürütülen operasyonel işlerin fazlalığı, çalışanların verimliliğinin düşmesine, zaman kaybına ve maliyetin artmasına sebep olmaktadır. Bu tarz durumların ortadan kaldırılması için güncel yöntemlerin şirket içi süreçlere uyumlu bir şekilde tasarlanarak ilgili bilişim sistemine entegre edilmesi önem kazanmıştır.

Şirketler, karar alma süreçlerini, tecrübe etkeninin yanı sıra hesaplanmış sayısal veriler ile desteklenmiş görsel raporlara evirmektedir. Dolayısıyla günümüzde şirketlerin stratejik, yönetimsel ve operasyonel süreçlerinde karar alma aşamalarındaki en önemli hususlardan biri ilgili konular için ihtiyaca yönelik hazırlanmış olan raporlar (tablo, çok boyutlu görsel hale getirilmiş grafikler vb.) haline gelmiştir.

Tüm sektörlerde olduğu gibi finans sektörü için de şirket içi raporlama yapıları önemli bir ihtiyaçtır. Büyük operasyonel süreçlerin ilerleyebilmesi, yönetimsel kararların sayısal veriler ile desteklenmesi, çok fazla konunun ve ürünün anlaşılabilir hale

gelmesi için görsel panoların oluşturulması finans sektöründe yer alan şirketlerin raporlama ortamlarına olan ihtiyaçlarını doğrudan arttırmaktadır.

Bireysel veya kurumasal raporlama ihtiyaçları için piyasaya sürülmüş birçok raporlama programı vardır. Finans sektöründe yer alan şirketler arasında yer alan bankalar, raporlama süreçlerinde maksimum verimi alabilmek, raporlama altyapısı sağlayabilmek için yazılımsal ve donanımsal ürünlere azami ölçüde önem vermektedir ve bu alandaki yatırımlardan kaçınmamaktadırlar.

Ürün çeşitliliği fazla olan şirketler raporlama programlarını daha çok model tabanlı seçmektedirler. Raporlama programları konusunda çok hassas davranan bu kurumlar en az bir veya birden fazla raporlama programı kullanmaktadırlar.

İhtiyaca göre seçilen raporlama programları pek çok yeteneğe sahiptir. Modelleme tabanlı raporlama programlarında temel mantık nesne yönelimli olmasıdır. Her bir değişken için bir obje oluşturulur ve bu obje ihtiyaç olan tüm raporlarda kullanılabilir. Örneğin; ‘müşteri numarası’ şeklinde bir obje oluşturulduğunda bu obje, bankada kullanılan tüm raporlarda müşteri numarasını görebilmek için kullanılabilir. ‘Müşteri numarası’objesinin değiştirilmesi ise objenin kullanıldığı tüm raporları etkiler. Bu sebeple büyük şirketler raporlama modellerini dikkatle oluşturmakta ve çalışmalarını titizlikle yürütmektedir. Gerektiğinde hem donanımsal hem yazılımsal konularda uzman dış kaynaklardan destek almaktadırlar.

Bankalar çok büyük yapılanması olan, farklı alanlarda yoğunlaşan onlarca birime bölünmüş, yüzlerce ekibe sahip organizasyonel yapılanmaya sahip olan kurumlardır. Dolayısıyla iş ihtiyacı çok büyük ve yönetilmesi zordur. Bankalarda raporlama ihtiyaçları opersayonel ve yönetimsel ana başlıkları ile yüzlerce alt başlık altında incelenebilir. Risk yönetimi, insan kaynakları, iç denetim, dış denetim, uyum, karlılık, skorlama, kampanya, kredi, kredi kartı, alternatif dağıtım kanalları (atm, mobil, internet), limit, muhasebe, mevduat, hazine, performans vb.

Kurumlarda raporlama ihtiyaçlarının karşılanması için ekipler oluşturulmaktadır. Bu ekipler de kendi içerisinde birden fazla ekibe bölünebilmektedir. Bilgi teknolojileri ekiplerinde yer alan, ihtiyacı müşteriden (bankalar için genel müdürlük ekiplerinde yer alan iş birimleri olarak adlandırılırlar) alan ve ihtiyaç dokümanlarını oluşturan analiz ekibi, ihtiyacı anlayarak uygun veriyi kaynak sistemden alan kurumun ilgili veri ambarı sistemine aktaran tasarım & yazılım ekibi ve raporlamaya uygun hale getirilen verilelerin depolandığı, veri ambarı sisteminde yer alan tabloları kullanarak istenilen formatta raporları tasarlayan ve müşteriye ileten raporlama ekibi şeklindedir. Üç başlık altında belirtilen analiz – tasarım & yazılım – raporlama ekipleri de konu bazlı alt ekiplere bölünebilmektedirler.

Genellikle bankaların da içerisinde olduğu büyük şirketlerde raporlama konusu ve ürün çeşitliliğinin çok fazla olması sebebi ile müşteriden gelen rapor ihtiyaçları havuz uygulamalarda toplanmaktadır. Bu havuz uygulamalardan raporlar konularına göre ilgili ekibin iş listesine atanır ve yöneticiler tarafından işi yapacak kişiye aktarılır.

Büyük kurumlarda organizasyon çeşitliliğinin, ürün çeşitliliğinin fazla olması dolayısıyla çalışan kişi sayısı da bir hayli fazladır. Öyle ki bankalarda binlerce kişi çalışmaktadır ve her geçen gün bu sayılar artmaktadır. Yüzlerce ekibe bölünmüş büyük hacimli kurumlarda ekiplerin ilgilendikleri konular ve ürünler ne kadar farklı olsada aynı konuları barındıran rapor ihtiyaçları çok fazladır. Dolayısıyla rapor talebinde bulunan müşteriler (bankalarda bilgi teknolojileri ekipleri için iş birimleri) farkında olunmadan aynı konulara sahip raporları birden fazla kere iş ihtiyacı olarak bilgi teknolojileri ekibine talep olarak iletebilmektedir. Bu çalışmada ise bilgi teknolojileri ekiplerine iletilen raporların NLP (Natural Language Processing - Doğal Dil İşleme) teknikleriyle işlenerek kümelenmesi ve benzerlikleri üzerinden talebe bağlı rapor adreslenmesi yapılmıştır.

1.1. Problemin Tanımı / Çalışmanın Gerekçesi

Büyük ölçekli kurumların, bilgi teknolojileri birimlerinde otomatize edilen süreçler kadar manuel ilerletilen süreçler de mevcuttur. Kurumların yenilikçi bakış açıları

sayesinde manuel sürdürülen süreçlerin otomatize edilmesi ve var olan süreçlerin teknoloji odaklı olması için yeniden yapılandırma çalışmaları yapılmaktır. İlgili çalışmanın yapıldığı kurumda bilgi teknolojileri ekiplerine gelen rapor taleplerinin operasyonel süreçleri manuel olarak ilerletilmektedir. Çalışmanın yapıldığı kurumda bilgi teknolojilerinde ilgili ekibin havuzuna düşen rapor talebi, ilgili ekip ve ekip yöneticisinin adına oluşturulmaktadır. İlgili ekip yöneticisi, talep içeriğini inceleyerek talep maliyetini ve talebin görev olarak atanacağı kişiyi belirler ve talebi maliyetlendirerek ilgili kişiye atar.

Müşteriler tarafından oluşturulan rapor taleplerinde, ilgili ekipler kimi zaman doğru adreslenememektedir. Bu da rapor talebini karşılayan ilgili ekip yöneticisi ve rapor talebini yapacak olan bilgi teknolojileri çalışanı tarafından farkedilerek doğru ekibe atanmasını gerektirmektedir. Talep yoğunluğunun çok fazla olduğu kurumlarda, rapor talebinin ilgili ekibe ulaşması uzun süreler alabilmektedir.

Bu çalışmada özel bir bankanın, bilgi teknolojileri ekipleri müşterileri tarafından, aynı konuları içerebilecek raporların birden fazla kere talep edilmesi ve raporların doğru ekiplere adreslenememesi gibi sorunların çözülebilmesi sebebi ile manuel talep karşılama süreçlerini hızlandırabilecek derin öğrenme temelli NLP (Natural Language Processing - Doğal Dil İşleme) algoritmaları kullanılarak, makine öğrenmesi çalışması yapılmıştır.

1.2. Çalışmanın Katkıları

Çalışmada, ilgili bankanın bilgi teknolojileri ekipleri müşterilerinin oluşturduğu rapor talep içerikleri ve talep detay bilgileri kullanılarak bir veri seti oluşturulmuştur. Oluşturulan veri seti NLP algoritmaları aracılığı ile modellenerek makine öğrenmesi yöntemlerinden olan kümeleme algoritmalarına girdi olarak sağlanarak kümeleme ve benzerlik çalışması yapılmıştır.

Bu çalışma ile ilgili bankanın bilgi teknolojileri ekibine ait yeni bir rapor talebi oluşturulduğunda, yeni talebe benzeyen daha önceden oluşturulmuş bir rapor talebi var

ise önceden yapılan talebin detay bilgilerine ulaşılır ve yeni oluşturulacak talep ile ilgili bilgi edinilerek talep tamamlanma maliyeti düşürülür.

Aynı veya benzer, başlık ve içeriğe sahip birden fazla talep varlığı tespiti kolaylaştırılarak, tekrarlı açılan rapor talepleri red edilir ve gereksiz talep maliyeti engellenir.

Var olan akışta her talep ilgili ekip yöneticisi tarafından incelenir ve konularına göre çalışanlara görev olarak atanır. Çalışma ile ilgili ekibe ait açık rapor talepleri kümelenerek, benzer talepler aynı gruba atanır ve yöneticiler talepleri küme bazında ilgili kişiye görev olarak atar. Bu sayede yöneticilerin talep atama maliyetleri azalır.

İlgili kişiler tarafından oluşturulan rapor talepleri kimi zaman yanlış ekiplere adreslenebilmektedir. Bunun sebebi ise birden fazla konuyu barındıran raporların talep edilmesi sebebi ile ağırlıklı konunun belirlenememesi, ilgili konunun hangi ekibin bilgisi dahilinde olduğunun bilinmemesi olabilmektedir. Çalışma ile ilgili rapor talebine ait benzer talepler önerildiğinde, önerilen talepler aracılığı ile doğru ekip bilgisi edinilerek talep yönlendirilir ve yönlendirme maliyetleri azalır.

Çalışmanın gerçekleme ortamı ve yöntemi sayesinde, yeni oluşturulacak rapor talebi ve adreslenen benzer taleplerin, başlık, içerik ve ekip gibi detay bilgilerine ulaşımı tek bir tablodan sağlanarak, havuz uygulamada önceden yapılan talep araştırma maliyetleri azaltılır.

Çalışmada her bir rapor bir metini ifade edecektir.

1.3. Tez Organizasyonu

Finans sektöründe temelli NLP (Doğal Dil İşleme) ile rapor kümelendirme ve talep bazlı rapor önerileri oluşturulmasına yönelik altı bölümden oluşan bu çalışma aşağıda belirtilen şekilde organize edilmiştir.

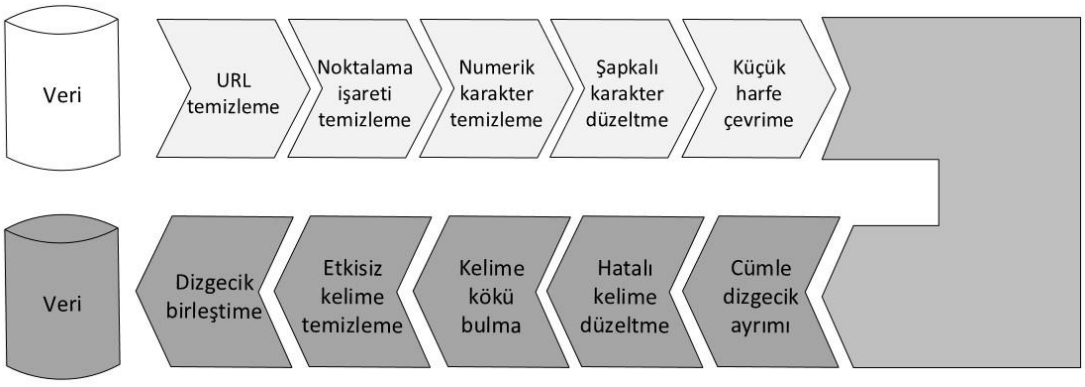
Bölüm 1’de problem tanımı, çalışmanın gerekçesi, çalışmanın katkıları ve tez organizasyonu hakkında bilgi verilmiştir. Bölüm 2’de çalışmada kullanılan yöntemler ve temel kavramlar ile birlikte bu yöntemlerin teorik olarak açıklamaları, denklemleri hakkında bilgi verilmiştir. Bölüm 3’de literatür araştırması yapılmış olunup önceki çalışmalara ait örnekler verilmiştir. Bölüm 4’de ilgili çalışmanın yapılmasını sağlayan probleme önerilmiş sistemin, bilgi/veri akışları hakkında diyagramlar oluşturulmuştur. Çalışma esnasında kullanılan parametreler / bileşenler ve çalışmada kullanılan veri setinin oluşturulma sürecine ek olarak veri seti ile ilgili detay bilgiler verilmiştir. Bölüm 5’de araştırma kapsamında uygulanan senaryoların her biri ile ilgili detay bilgiler verilmiş, senaryo sonuçları grafikler ile açıklanmış ve çalışmanın yapıldığı kurumda projenin uygulanması hakkında bilgileri verilmiş ve sonuçları değerlendirilmiştir. Bölüm 6’da ise çalışma hedefi, veri seti, çalışmada benimsenmiş olan yöntem ve özeti hakkında verilmiş, çalışma esnasında yaşanılan zorluklardan bahsedilerek araştırma sonuçları belirtilmiş olunup, çalışma sonucunda edinilen yol gösterici bilgiler paylaşılarak sonraki çalışmalar önerilmiştir.

BÖLÜM 2. TEMEL TANIM, KAVRAM VE YÖNTEMLER

Çalışmanın bu bölümünde, literatür araştırması sonucunda çalışmada kullanılması belirlenen yöntemler ile ilgili temel kavramlar açıklanarak, yöntemlerin teorik açıklamaları hakkında bilgiler verilmiştir. Bu bağlamda ilk olarak veri temizleme ve ön işleme süreçleri hakkında bilgiler verilmiştir.

2.1. Veri Temizleme ve Ön-İşlem Yöntemleri

NLP uygulamalarında model başarısı verinin temizliği ve ön hazırlıkları ile doğru orantılıdır. Sayısallaştırılacak verinin gürültüden yani; numaralardan, noktalama işaretlerinden, durak kelimelerden, aşırı değerlerden arındırılması, yapısal bir formata getirilmesi için çok büyük önem taşımaktadır. Bu sebeple sonraki başlıklarda veri manipülasyonu ile ilgili aşamalar hakkında bilgi verilmiştir.



Şekil 2.1. Veri temizleme ve ön-işlem yöntemleri

2.1.1. Veri temizleme

Veri temizleme aşamasında, veri setinde sadece numerik olmayan karakterler kalacak şekilde işlemler gerçekleştirilir. Bu çalışmada veri temizleme işlemleri sırası ile Şekil

2.1.'de belirtilen şekilde yapılmıştır. Dokümanlarda yer alan URL'lerin temizlenerek, noktalama işaretleri kaldırılmış, ardından numerik karakterler temizlenerek, şapkalı harfler şapkasız eşlenikleri ile değiştirilmiş ve veri setinde yer alan tüm karakterler küçük harflere çevrilmiştir.

Çalışmada veri ön işlem aşamaları Türkçe'ye kazandırılmış bir doğal dil işleme kütüphanesi olan Zemberek ile gerçekleştirilmiştir.

2.1.2. Zemberek kütüphanesi

Zemberek kütüphanesi Java programlama dili ile 2007 yılında geliştirilmiş ve halen geliştirmesine devam edilen açık kaynak kodlu, doğal dil işleme kütüphanesidir. Zemberek kütüphanesi, sadece Türkçe için değil ihmal edilmiş Türk dilleri için de temel bir doğal dil işleme sistemi amaçlayarak oluşturulmuştur (Akın, Akın, 2007).

Kütüphanenin farklı dillerde desteği mevcuttur ve bu kütüphane Tablo 2.1.'de belirtildiği gibi pek çok yeteneğe sahiptir.

Tablo 2.1. Zemberek kütüphanesi yetenekleri

Yetenek	Literatür Tabir	Sınıf	İşlev
Kelime kökü bulma	Stemming	TurkishMorphology	Kelimelerin kök hallerini bulur.
Kelime gövdesi bulma	Lemmatization	TurkishMorphology	Kelimelerin gövdesini tespit eder. Bazı kelimeler, özel isim ve kısaltmalar doğru analiz edilemeyebilir.
Noktalı kelimelerin doğru analiz edilmesi	Diacritics Ignored Analysis	ignoreDiacriticsInAnalysis	Türkçe aksan işaretlerini [ç,ğ,i,ö,ü,ş] yoksaymadan kelime analizi yapar. (kisi -> kış,kişi)
Kelime oluşturma	Generate word	TurkishMorphology	Girdi olarak verilen kelimenin çekimlerini oluşturmak için iyelik ve durum eki kombinasyonları kullanarak kelime önerir.

Tablo 2.1. (Devamı)

Yetenek	Literatür Tabir	Sınıf	İşlev
Cümle sınırı denetleme	Sentence Boundary Detection	TurkishExtractor	Doküman veya paragraf olarak verilen girdiyi, paragraflara veya cümlelere bölerek listeler.
Cümleyi dizgeciklere bölme	Tokenization	TurkishTokenizer	Cümlede yer alan kelimeleri boşluk, sekme, satır başı veya satır sonuna göre ayırma görevi yapar.
Hatalı yazımları düzeltme	Noisy Text Normalization	TurkishSentenceNormalizer	Hatalı yazılan kelimelerin doğru hallerine çevirir.
Hatalı kelime kontrol etme	Turkish Spell Checker	TurkishSpellChecker	Girdi olarak verilen kelimenin doğru yazılıp yazılmadığını kontrol etmek için yöntemler sağlar.
Yazım önerme	Spelling Suggestions	TurkishSpellChecker	Girdi olarak verilen kelimeye öneri olarak bir veya birden fazla kelime listeler.
Varlık isim tanıma	Turkish Named Entity Recognition	TurkishMorphology	Türkçe için basit bir NER model sağlar. Girdide yer alan benzersiz varlıkları (insan, mekan, organizasyon vb.) tanımlamak için kullanılır.
Metin sınıflandırma	Text Classification	FastTextClassifier	Metin sınıflandırma, belge sınıflandırma, duygu analizi veya istenmeyen posta algılama gibi çeşitli NLP görevlerinde kullanılabilir.
Etkisiz kelime çıkarma	Stopwords		Cümleden etkisiz kelimeleri çıkarır.

2.1.3. Cümleyi dizgeciklere ayırma (tokenization)

Her bir dokümanın kendi içerisinde kelimelere & dizgeciklere bölünmüş halini ifade etmektedir. Zemberek Kütüphanesi'nin 'TurkishTokenizer' modulü kullanılarak her bir doküman tokenize edilebilmektedir.

2.1.4. Hatalı yazım düzeltme (noisy text normalization)

Hatalı yazım düzeltme (noisy text normalization), ilgili dokümanda yer alan kelimelerdeki gürültüyü azaltma işlemlerinden biridir. Kelimelerin yanlış yazımının

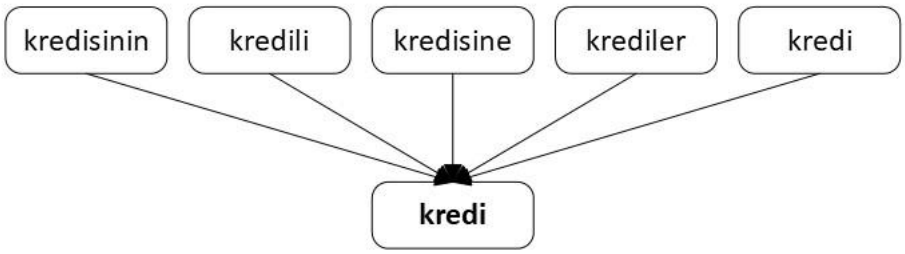
düzeltilmesi, karakter büyüklüğü ve küçüklüğünün düzeltilmesi, kısaltmaların düzeltilmesi gibi işlemlerin yapılması işidir. Zemberek Kütüphanesi'nin 'TurkishSentenceNormalizer' modulü kullanılarak her bir doküman tokenize edilebilmektedir.

2.1.5. Kök bulma (stemming)

Kök bulma (stemming), kelimelerin eklerinden kurtarılarak en basit haline dönüştürülmesi işlemidir.

2012'de yapılan bir çalışmada, kök bulmanın kümeleme kalitesine ek olarak, çok boyutlu metin verilerinde yüksek oranda boyut indirgeme sağladığı ifade edilmiştir (Tunalı ve Bilgin, 2012).

Örneğin; metinlerde, "kredi" kelimesinin "kredisinin", "kredili", "kredisine", "krediler" vb. şeklinde çekimlenmiş halleri yer alabilmektedir. Belirtilen 4 farklı kelime, kök bulma ön işlem sürecine dahil edildiğinde, her birinin kökü "kredi" olarak ortaya çıkacak ve 4 değişken 1 değişkene indirgenmiş olacaktır. Şekil 2.2.'de kredi kelimesinin çekimlenmiş hallerinin kök bulma süreci gösterilmiştir.



Şekil 2.2. Kök bulma (stemming)

2.1.6. Etkisiz kelimelerin kaldırılması (stopwords)

Veri setinin içerdiği, ilgili dokümana katkısı olmayacak olan kelimeler etkisiz kelimeler (stopwords) olarak adlandırılmaktadır. Bağlaçlar, sık kullanıma sahip olan kelimeler, dokümana katkısı olmayan etkisiz kelimelere örnek olup dokümandan

kaldırılması, çalışma başarısını arttırabilmektedir. Etkisiz kelimeler belirlenirken doğal dil işleme için oluşturulmuş kütüphanelerin listeleri veya sıfırdan oluşturulan etkisiz kelime listeleri kullanılabilir.

2.2. Kelime Gömme (Word Embedding) Yöntemleri

Jurafsky ve Martin'e göre doğal dil işlemede kelime gömme (word embedding), kelimelerin anlamlarının bir temsilidir. Yani, dokümanları makinelerin anlayabileceği hale getirebilmek için vektörel hale getirmek & sayısallaştırmaktır. Kısaca her bir kelimeyi sayısal bir ifade ile temsil etme yöntemidir. Bu yöntem bir çeşit dil modelleme tekniğidir. Dokümanlarda yer alan kelimeler sayısallaştırılarak, her biri birer vektör haline getirilip vektör uzayında temsil edilmektedir.

Kelime temsil yöntemleri yapay sinir ağları, istatistik ve olasılık modelleri kullanılarak oluşturulmaktadır. Günümüzde ise birden fazla kelime temsil yöntemi vardır. Bu yöntemlerden bazıları frekans bazlı yöntemler, bazıları ise tahmin bazlı yöntemlerdir.

2.2.1. Frekans bazlı kelime temsil yöntemleri

Kelimelerin dokümanda geçme sıklığı temel alınarak hesaplamalar yapılmakta ve kelimelerin sayısal temsilleri oluşturulmaktadır. Frekans bazlı kelime temsil yöntemlerinde kelimeler arasında anlamsallık aranmaz.

2.2.1.1. TF-IDF (Terim Frekansı - Ters Doküman Frekansı)

Doküman terim ağırlıklandırma 1957 yılında Hans Peter Luhn tarafından, ters doküman ağırlığı ise 1972 yılında Karen Spärck Jones tarafından tasarlanmıştır.

Aizawa'ya göre TF-IDF kelime temsil yöntemi kelimelerin dokümanlar içerisindeki matematiksel önemini belirlemeye yarayan istatistiksel bir ölçüdür (Aizawa, 2003).

Bir vektör uzay modeli olan bu yöntemde kelimelerin metin içerisinde geçme sıklıklarına göre frekanslar oluşturulur.

TF değeri, kelimelerin dokümanda geçme sıklığını belirler. Yani terim sıklığı, doküman içerisindeki ilgili terim sayısının, dokümandaki toplam terim sayısına oranını ifade eder. Ağırlıklandırma yönteminde dokümanda daha fazla geçen kelimeler yani TF değeri büyük olan kelimeler doküman için değerlidir.

IDF değeri ise dokümanda seyrek geçen kelimeler ile ilgili bilgi vermektedir. Yani toplam doküman sayısının, ilgili terimin geçtiği doküman sayısına olan oranının logaritmasının mutlak değerini ifade eder. IDF değeri ilgili doküman için belirleyici özellik olabilmektedir. TF-IDF aşağıdaki şekilde eşitlik (Denklem 2.1) kullanılarak hesaplanmaktadır.

$$(\mathrm{TF}-\mathrm{IDF})_{\mathrm{w_i}} = \mathrm{TF_{i,j}} \times \mathrm{IDF_i} \tag{2.1}$$

$$\mathrm{TF_{i,j}} = \frac{\mathrm{n_{i,j}}}{\sum_{\mathrm{k}} \mathrm{n_{k,j}}} \tag{2.2}$$

$$\mathrm{IDF_i} = \log\left(\frac{|\mathrm{N}|}{|\{\mathrm{j}: \mathrm{w_i} \in \mathrm{x_j}\}|+1}\right) \tag{2.3}$$

Burada $\mathrm{TF_{i,j}}$ i. terimin j. dokümanda geçme oranı, $\mathrm{n_{i,j}}$ i. kelimenin j. dokümanda geçme sayısını (Denklem 2.2), $\sum_{\mathrm{k}} \mathrm{n_{k,j}}$ j. dokümandaki toplam terim sayısını, $\mathrm{IDF_i}$ i. terimin ters doküman frekansını, N toplam doküman sayısını, w_i kelime i'yi, $\mathrm{x_j}$ doküman j'yi ve $|\{\mathrm{j}: \mathrm{w_i} \in \mathrm{x_i}\}|$ i. kelimenin geçtiği doküman sayısını (Denkem 2.3) ifade etmektedir. Denklem 2.3'de eşitliğin alt kısmına 1 eklenmesinin sebebi i. kelime hiçbir dokümanda geçmiyorsa eşitliğin bozulmaması içindir (Qi, Z., 2020).

2.2.2. Tahmin bazlı kelime temsil yöntemleri

Tahmin bazlı kelime temsil yöntemleri, temelinde dokümanları yapay sinir ağı ilkesine bağlı olarak eğitmektedir. Kelimeler arasında anlamsal bağlar kurularak

vektörleştirme yapılmaktadır. Birden fazla tahmin bazlı kelime temsil yöntemi bulunmaktadır.

2.2.2.1. Word2vec

Word2vec, tahminleme bazlı bir kelime temsil yöntemidir. 2013 yılında Mikolov ve arkadaşları tarafından ortaya çıkarılmıştır. Danışmansız / denetimsiz (unsupervised) öğrenme yaklaşımını temel alınır ve yapay sinir ağı ilkelerine bağlı kalınarak kelimeler eğitilir.

Giriş, gizli ve çıkış olmak üzere 3 katmandan oluşur. Hiper parametrik olan bu yöntem, ilgili veri seti için en uygun parametreleri alarak, kelimelerin veri seti içerisindeki konumuna göre vektör oluşturmaktadır. Bu yöntem uygulanırken, oluşturulan model parametreleri için, en iyi değerleri belirleyen hesaplamalar literatürde mevcut değildir. İlgili veri setinin özellikleri göz önünde bulundurularak birden fazla model denemesi ile en iyi parametrelere ve modele karar verilebilmektedir.

Literatürde iki farklı Word2vec algoritması (Şekil 2.3.) mevcuttur. İki yöntem de pencere boyutu kavramı ile çıkış kelimesini tahminlemeye çalışır. Pencere boyutu parametresi, model eğitilirken tahmin edilmesi istenen çıkış kelime veya kelimelerinin sağında ve solunda yer alan n kadar sayıda komşu kelime ile modelin eğitilmesini ifade eder.

- CBoW (Continuous Bag of Words)

Burada tahmin edilmek istenen kelime pencerenin merkezinde yer almaktadır. Merkezde yer alan kelime ise, belirlenen pencere boyutu kadar sağ ve sol komşulara bakılarak tahmin edilmeye çalışılmaktadır.

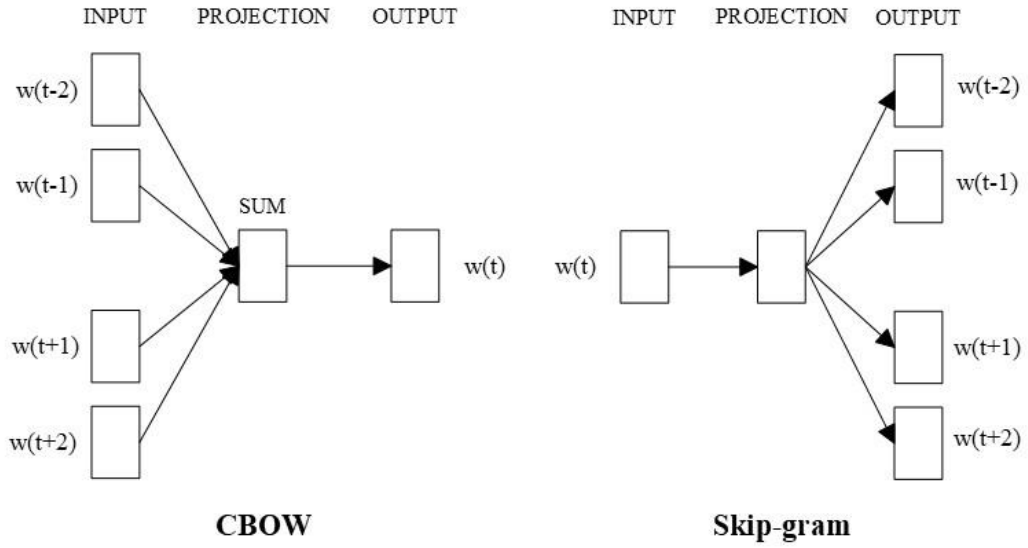
Örneğin; “Türkiye’nin başkenti İstanbul’dur.” model, CBoW yöntemi ile eğitilirse bu cümleden öğreneceği (pencere boyutu 1 olarak belirlendiğinde) merkezdeki kelime

olan ‘başkent’tir. Merkezdeki kelimeyi ise ‘Türkiye – İstanbul’ ikilisinden öğrenecektir.

- Skip-Gram

Bu yöntemde ise merkezdeki kelimeye bakılarak belirlenen pencere boyutu sayısı kadar komşu kelimeler tahmin edilmeye çalışılır.

Örneğin; “Türkiye’nin başkenti İstanbul’dur.” Aynı cümle, Skip – Gram yöntemi ile eğitilirse (pencere boyutu 1 olarak belirlendiğinde) modelin bu cümleden öğreneceği ‘Türkiye – İstanbul’dur. Komşu kelimeler ise merkezde bulunan ‘başkent’ kelimesinden yola çıkılarak öğrenilecektir.



Şekil 2.3. Word2vec - CBoW ve Skip-Gram (Mikolov ve ark., 2013)

2.2.2.2. Doc2vec (Paragraph Vector)

Doc2vec, Word2vec’in temsilcileri olan Le ve Mikalov’un doküman temsilleri elde etmek için 2014 yılında ortaya çıkardıkları bir yöntemdir. Denetimsiz öğrenme temelli Word2vec yöntemini temel alarak oluşturulmuştur. Doküman uzunluğuna bakılmaksızın dokümanların sayısal temsilini oluşturmaktadır.

Word2vec yönteminde yer alan değişkenlere ek olarak Doc2vec yöntemine doküman ID eklenmiştir. Kelime vektörleri w değişkenini eğitirken, doküman ID vektörü D yi eğitir ve dokümanın sayısal bir temsilini oluşturur.

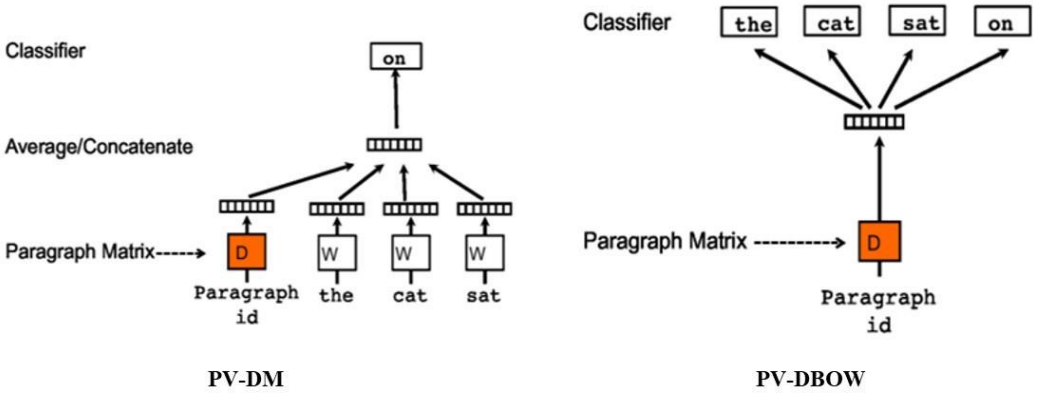
Literatürde iki adet Doc2vec yöntemi (Şekil 2.4.) yer almaktadır.

- PV-DM (Paragraph Vector Distrubuted Memory Model)

Word2vec yönteminin CBoW algoritması ile benzer yapıdadır.

- PV-DBoW (Paragraph Vector Distrubuted Bag of Words)

Word2vec yönteminin Skip – Gram algoritması ile benzer yapıdadır.



Şekil 2.4. Doc2vec - PV-DM ve PV-DBoW (Le, Mikolov, 2014)

2.3. Boyut Küçültme

Boyut küçültme, çok boyutlu veri setlerinde yok denecek seviyelerde bilgi kaybını amaçlayarak, veri setini eşdeğeri olan daha küçük boyutlara dönüştürme tekniğidir. Metin madenciliği çalışmalarında, küçük metinler dahil olmak üzere çok fazla öznitelik (boyut) çıkarımı olabileceğinden boyut küçültme, en önemli aşamalardan biridir. Literatürde birçok boyut küçültme tekniği yer almaktadır.

2.3.1. Kesik Tekil Değer Ayrışımı (KTDA – Truncated SVD)

Kesik tekil değer ayrışımı (Truncated SVD), metin vektörlerini girdi olarak alan anlamsal boyut küçültme tekniğidir.

2020 yılında yapılan bir çalışmada, hesaplama açısından daha verimli olması sebebiyle geleneksel tekil değer ayrışımı (SVD – singular value decomposition) ve temel bileşen analizi (PCA – principal component analysis) yerine kesik tekil değer ayrışımı (KTDA) kullanılmıştır (Subba, Gupta, 2020). KTDA, matris temsilindeki gürültüyü azaltır ve kümeleme doğruluğunu iyileştirir (Djellali,2013).

Kesik tekil değer ayrışımı, yüksek boyutu daha düşük boyuta dönüştürerek zaman ve depolama alanını azaltır, basitliği ve düşük hesaplama maliyeti sebebi ile diğer boyut küçültme tekniklerine tercih edilir (Hansen,1987).

Kesik tekil değer ayrışımı (KTDA), tekil değer ayrışımı (SVD) ve temel bileşen Analizi (PCA) ile karşılaştırıldığında seyrek verilerin işlenmesi için daha uygundur (Subba, Gupta, 2020).

Tekil değer ayrışımı (TDA), lineer cebirde bir matrisi çarpanlarına ayırma tekniklerinden biridir.

$$A = U\Sigma V^T \tag{2.4}$$

Matematiksel olarak, m değerinin doküman sayısını, n değerinin ise terim sayısını ifade ettiği m × n boyutlu ve rankı r olan bir A doküman-terim matrisinde TDA, m ≥ n olmak üzere Denklem (2.4)'deki gibi ifade edilir. Burada $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ ve $\Sigma \in \mathbb{R}^{m\times n}$ dir. U ve V matrisleri ortogonal, Σ matrisi ise diyagonal (köşegen) matrislerdir. A matrisinin tekil değerleri olarak adlandırılan σ_i ler Σ'nun köşegen elemanlarıdır.

A'nın rankına göre durumlardan bahsedilebilir. Bunlardan biri $\mathrm{rank}(A) = n$ ise

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_n \end{bmatrix} \quad (2.5)$$

(Denklem 2.5) şeklindedir ve Σ'nun tekil değerleri $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$ biçiminde sıralanmış olabilir. Diğeri, $\text{rank}(A) = r < \min(m, n)$ ise $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$ sıralaması geçerlidir,

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.6)$$

(Denklem 2.6) şeklinde yazılabilir ve $\Sigma_1 = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$'dir.

Kesik tekil değer ayrışımı (KTDA) ise A matrisinin tekil değerleri olarak adlandırılan $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$ değerleri arasından, ilk k adet en büyük tekil değerin seçilip, geri kalanların tümünün sıfıra eşitlenmesi ve böylece U ve V'nin sadece ilk sütununun kullanılması ile bulunur ve bu şekilde sonraki adımlarda matris hesaplarını kolaylaştırır (Peker, Kubat, 2021).

2.4. Kümelendirme Yöntemleri

Kümeleme analizinin temel amacı, ilgili veri setinde grupları belli olmayan verilerin ilgili benzerlik değerine göre kümelere dahil edilerek gruplandırılmasıdır. En iyi denetimsiz makine öğrenmesi yöntemlerinden biridir.

Kümeleme analizi, veri setinin doğal sınıflamaları hakkında bilgi sahibi olunmayan durumlarda, veri setine ilişkin tahminlerin yapılması için kullanılan teknikler bütünüdür (Hartigan, 1975).

Literatürde yapılan çalışmalarda, farklı kümeleme yöntemlerinin ilgili veri seti için aynı sonuçları vermeyebileceğinden, bu sebeple de tek bir kümeleme yöntemine bağlı

kalınmadan farklı kümeleme yöntemleri ile denemeler yapılması önerilmiştir (Anderberg, 1973).

2020 yılında yapılan bir çalışmada hiyerarşik yöntemlerin, hiyerarşik olmayan yöntemlere göre daha uzun zaman aldığına yer verilmiştir. Aynı çalışmada ilgili araştırmacının, veri setine uygun küme sayısına karar verebilmesi durumunda veya kullanılan kümeleme yöntemine uygun, kümeleme öncesinde küme sayısını belirleyebilen bir yöntem var ise hiyerarşik olmayan kümeleme yöntemlerini kullanması önerilmiştir (Demirkale, Özarı, 2020).

Kümeleme yönteminin dezavantajlarından biri ilgili veri setinin sahip olduğu küme sayısının bilinmemesidir. Bazı kümeleme yöntemleri, kümeleme öncesi küme sayısnın bilinmesine ihtiyaç duymazken bazı yöntemler için küme sayısının bilinmesine ihtiyaç vardır. Bu sebeple de küme sayısının belirlenmesi için literatürde çeşitli yöntemler yer almaktadır.

Belirlenen küme sayıları ile kümeleme işlemi gereçekleştirildikten sonra, kümeleme başarılarının yorumlanabilmesi için literatürde çeşitli yöntemler mevcuttur. Bu yöntemlere küme doğrulama (cluster validation) yöntemleri adı verilmektedir. İlgili yöntemler aynı zamanda küme sayısı belirlemek için de kullanılmaktadır.

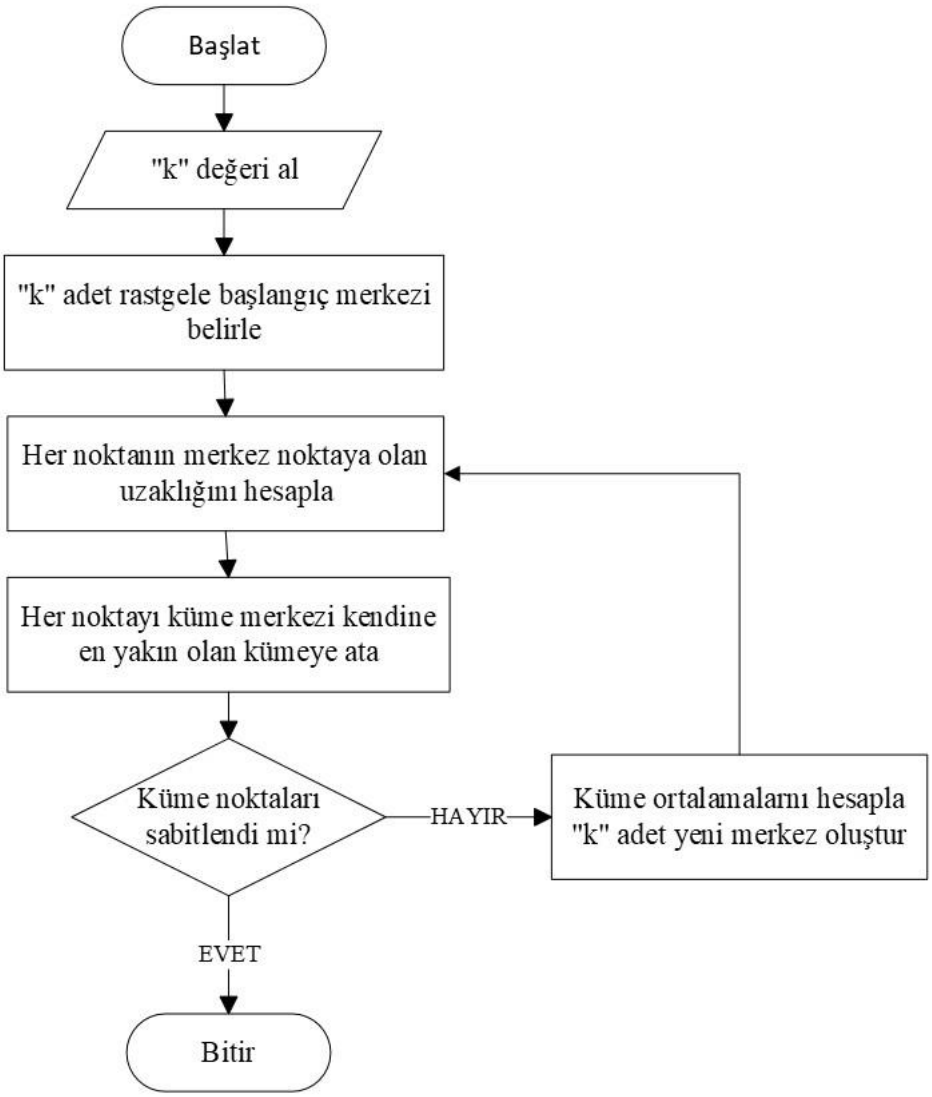
2.4.1. K-Ortalamalar (K-Means)

1967 yılında J.B. MacQueen tarafından geliştirilen K-Ortalamalar (K-Means) kümeleme algoritması en eski kümeleme algoritmalarındandır (MacQueen, 1967). Literatürde en fazla kullanılan denetimsiz öğrenme yöntemlerinden biri olan K-Means keskin bir kümeleme algoritmasıdır. Bu yöntemde her veri ancak bir kümeye ait olabilmektedir. Kümeler içi maksimum homojenlik, kümeler arası maksimum heterojenlik ilkesine dayanır.

K-Ortalamalar yönteminde başlangıçta geçerli olan küme sayısı kullanıcı tarafından belirlenmesi gereken bir parametredir. Küme sayısı hakkında bir bilgi yok ise

literatürde küme sayısını belirleyecek yöntemler aracılığı ile en uygun küme sayısına karar verilmelidir.

K-Means kümeleme algoritması Şekil 2.5.'de gösterilmiştir.



Şekil 2.5. K-Ortalamalar (K-Means) algoritması

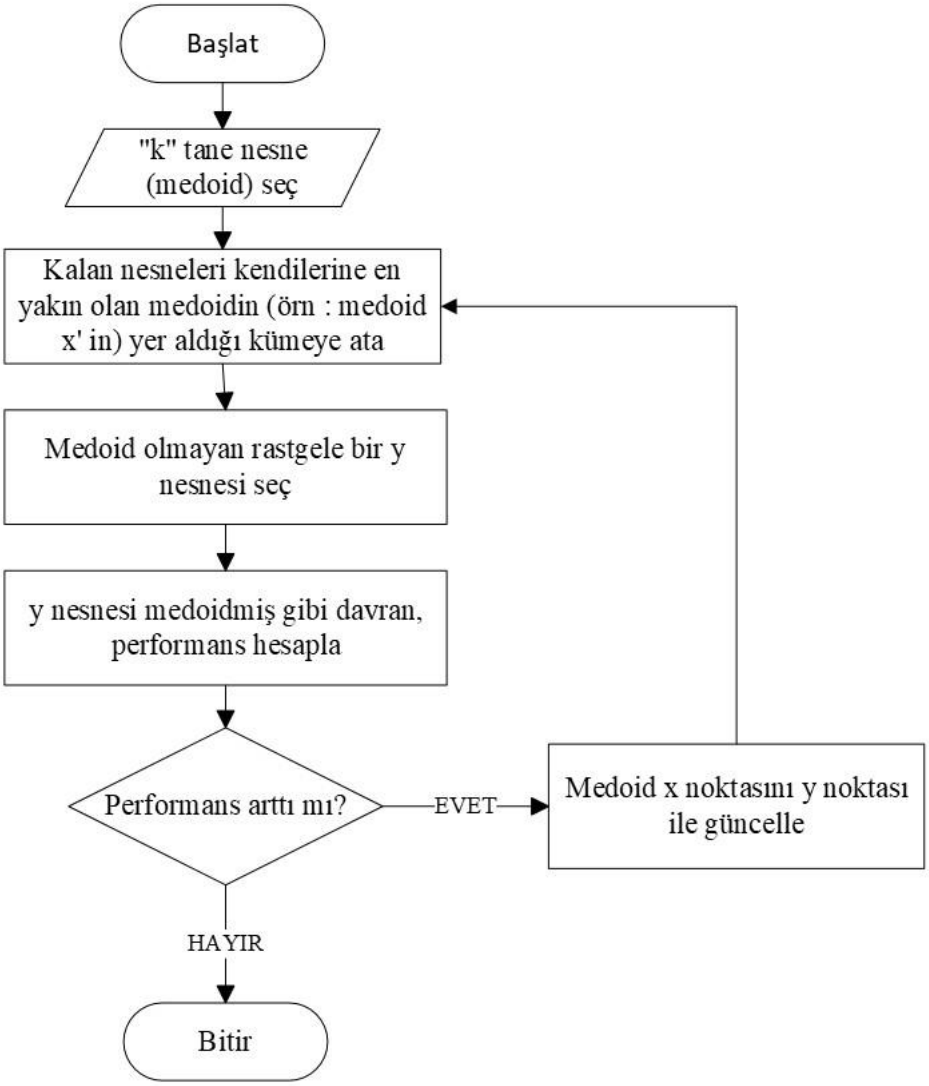
2.4.2. K-Temsilci (K-Medoids)

K-Temsilci (K-Medoids) kümeleme algoritması, veri kümesinin çeşitli yönlerini temsil etmesi istenen, k temsili nesne arayışı temeline dayanmaktadır ve ilgili kümenin temsili nesnesi, kümenin tüm nesnelerine olan ortalama farklılığın minimum olduğu nesnedir (Kaufman ve Rousseeuw, 1987). Bu nesne, orta nokta (medoid) olarak adlandırılır ve küme merkezine en yakın nokta olarak bilinir.

Literatürde, temsilci nesnelerin medoid olarak adlandırıldığı PAM (Partitioning Around Medoids) Algoritması'nda amaç, k tane temsilci nesneyi bulmak olduğundan K-Medoids Algoritması olarak yer almaktadır (Kaufman ve Rousseeuw, 1990).

Hesaplama karmaşıklığı sebebi ile küçük veri kümeleri için uygundur (Silahtaroğlu, 2016).

K-Medoids kümeleme algoritması çalışma mantığı Şekil 2.6.'da gösterilmiştir.



Şekil 2.6. K-Temsilci (K-Medoids) algoritması (Altıntaş, 2006)

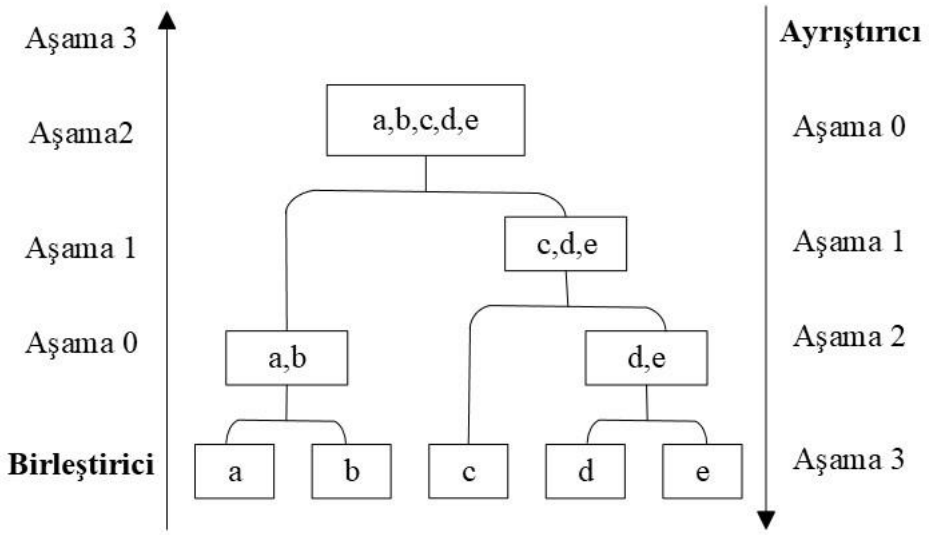
2.4.3. Hiyerarşik kümeleme (Birleştirici (Agglomerative) – Ayrıştırıcı (Divise))

Hiyerarşik kümeleme prensibi, başlangıçta her bir verinin tek tek küme olarak ele alınarak, bir küme oluncaya dek aşamalı olarak birleştirilmesi veya tüm veriyi içeren ana bir küme ile başlatılarak aşamalı olarak alt kümelere ayrıştırılması esasına dayanmaktadır. Birleştirici ve ayrıştırıcı olmak üzere iki türe (Şekil 2.7.) sahiptir.

Birleştirici hiyerarşik kümeleme yönteminde başlangıçta her bir veri, bağımsız bir küme olarak ele alınır ve belirlenen algoritmalar aracılığıyla tekrarlı bir şekilde bütün verileri içeren tek bir küme elde edilene kadar, her bir veri & veri kümesinin kendisine

en yakın olan veri & veri kümesi ile bir küme oluşturması sağlanır. Her bir aşamada n adet veri & veri kümesi için n-1 birleştirme yapılır.

Ayrıştırıcı hiyerarşik kümeleme yönteminde başlangıçta tüm veriler tek bir küme olarak ele alınır ve daha sonra tekrarlı bir şekilde, bütün veriler birbirlerinden bağımsız birer küme oluncaya dek her bir veri & veri kümesi kendisinden en uzak olan veri & veri kümesinden ayrılıp yeni bir küme oluşturması sağlanır. Şekil 2.7.'de 5 adet veriye sahip bir veri setine ait hiyerarşik kümeleme adımları gösterilmiştir.



Şekil 2.7. Hiyerarşik kümeleme

Birleştirici ve ayrıştırıcı kümeleme algoritmalarında hatalı birleştirme veya ayrıştırma düzeltilemez.

Hiyerarşik kümeleme yöntemlerinde k sayısının bilinmesine gerek yoktur. Ağaç yapısında olan bu kümeleme yöntemi için dendrogramlar oluşturularak küme sayısı yorumlanabilmektedir. Lakin dendrogramlar da küme sayısını net olarak veremezler, en uygun küme sayısını belirlemek için hesaplama yöntemi mevcut olmadığından dendrogramdan küme sayısı tespit etmek yalnızca yorumlanabilir niteliktedir.

Literatürde hiyerarşik kümeleme yöntemlerinin küçük veri setleri için kullanılması önerilmektedir.

Hiyerarşik yöntemlerde kümeler, veri & veri kümeleri aralarındaki benzerliğe göre belirlenmektedir. İki veri & veri kümesi arasındaki benzerliği ölçen ise birden fazla yöntem bulunmaktadır. Literatürde bu yöntemlerden en iyi sonuç vereni ve en fazla kullanılanı 'ward' olarak yer almaktadır.

Ward yönteminde amaç, kümeler içerisinde yer alan veriler arasındaki varyans değerinin minimum olmasıdır. İki küme arasındaki uzaklığın hesaplamasında merkezden sapmaları esas alınmaktadır. Kısaca veri & veri kümeleri arasındaki benzememe durumlarına bakılarak, kümeleme işlemi yapılmakta ve buna en küçük varyans yöntemi de denilmektedir.

2.5. Kümeleme Sonuçlarının Değerlendirmesi (Cluster Validation) ve Küme Sayısına Karar Verilmesi

Kümeleme analizlerinde, küme sayısına karar verilmesi ve küme başarılarının değerlendirilmesi için çeşitli yöntemler mevcuttur. Küme başarısını ölçme ifadesi, aynı zamanda ilgili veri seti için en iyi küme sayısının bulabilmek demektir.

Küme doğrulama terimi kümeleme algortimalarının başarısını ölçebilmek için kullanılmaktadır (Kassambara, 2017).

Çeşitli yıllarda yapılan çalışmalarda kümeleme doğrulama yöntemleri 3 kategoride incelenmiştir (Theodoridis and Koutroubas, 2008; G. Brock et al., 2008, Charrad et al., 2014).

- Dahili küme doğrulama (Internal cluster validation): Dış veri olmadan küme sayısını ve uygun kümeleme yöntemini tahmin etmek için kullanmaktadır. Kümeleme performansını değerlendirirken gerçek küme ya da sınıf etiketlerine gerek duyulmadan kullanılabilen yöntemlerdir (Aslanyürek ve Mesut, 2021).

- Harici küme doğrulama (External cluster validation): Kümeleme öncesinde verilerin hangi sınıfa & kümeye dahil olduğu bilinmektedir ve kullanılan

kümeleme yöntemi için referans olarak kullanılmaktadır. Kümeleme performansını değerlendirirken gerçek küme ya da sınıf etiketlerinin bilindiği durumlarda kullanılabilen yöntemlerdir (Aslanyürek ve Mesut, 2021).

- Göreceli küme doğrulama (Relative cluster validation): Aynı kümeleme yöntemi için farklı parametreler kullanılarak en uygun küme sayısını belirlemek için kullanılmaktadır. Örneğin; k küme sayısını değiştirerek en uygun küme sayısını bulmak.

2016 yılında yapılan bir çalışmada, kümeleme analizi çalışmalarında, en uygun küme sayısının belirlenmesinde daha çok içsel kriterlerin (dahili küme doğrulama yöntemleri) kullanıldığının görüldüğü sonucunu elde edilmiştir (Hacıoğlu, 2016). Bunun nedeni ise dahili küme doğrulama yöntemlerinin harici küme doğrulama yöntemlerine göre doğal ve anlamlı küme yapıları oluşturmada daha başarılı sonuçlar vermesine bağlanmıştır (Aydın ve Seven, 2015).

Bu çalışmada kullanılan veri setine ait metinlerin, sınıf & küme bilgisi olmadığı için dahili küme doğrulama yöntemleri kullanılmıştır.

Kümeleme doğrulama yöntemleri aynı zamanda küme sayısı belirlemek için de kullanılmaktadır. İlgili algoritmalar birden fazla küme sayısı için çalıştırılarak, çalışma prensibine göre maksimum ya da minimum değeri veren küme sayısı, en uygun küme sayısı olarak belirlenmektedir.

2.5.1. Silhouette katsayısı (Silhoutte coefficient)

İlgili noktanın kendi kümesi içerisindeki diğer noktalara olan uzaklığına ve diğer kümelerdeki noktalara olan uzaklığına dayanan Silhoutte istatistiğinin, ortalama değerini maksimuma ulaştıran küme sayısı, seçilebilecek en uygun küme sayısıdır (Rousseeuw, 1987).

$$s(i) = \begin{cases} 1 - a(i)/b(i) & a(i) < b(i), \\ 0 & a(i) = b(i), \\ b(i)/a(i) - 1 & a(i) > b(i). \end{cases} \quad (2.7)$$

$$s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \quad (2.8)$$

$$-1 \leq s(i) \leq 1 \quad (2.9)$$

Burada $a(i)$ i. noktanın kendi kümesindeki tüm noktalara olan ortalama uzaklığını, $b(i)$ ise i. noktanın diğer kümelerdeki tüm noktalara olan uzaklıklarının en küçük değerini ifade etmektedir.

Bu bilgiler ışığında i noktası için silhoutte indeksi olan $s(i)$ değeri Denklem 2.7 ve Denklem 2.8 aracılığı ile hesaplanır (Rousseeuw,1987).

$s(i)$ değerinin pozitif olması ilgili noktanın doğal kümesine uygun olduğunu, negatif olması (Denklem 2.9) ise doğal kümesine uygun olmadığını gösterir (Rousseeuw,1987).

Tüm noktalara ait kümeleme işleminin geçerliliği için ise ortalama Silhoutte değeri hesaplanır.

$$S = \frac{1}{n}\sum_{s_i \in S} s(i) \quad (2.10)$$

S değeri ortalama siluet değerini ifade etmektedir ve Denklem 2.10'da yer alan eşitlik ile bulunur (Rousseeuw,1987).

Bu hesaplamaya göre maksimum ortalama Silhoutte değerine ulaşan küme sayısı en uygun küme sayısı olarak belirlenir (Rousseeuw, 1987).

2.5.2. Davies-Bouldin indeksi (DB indeks)

Küme içerisindeki noktaların küme merkezine olan uzaklıklarını minimum, kümeler arasındaki uzaklıkları maksimim yapmayı hedefleyen, DB indeks değerini minimuma ulaştıran küme sayısı, en iyi kümelemeyi ifade eder (Davies, Bouldin, 1979).

i = 1,2,..,k ve j = 1,2,..,k olmak üzere i. ve diğer kümeler arasındaki maksimum karşılaştırma oranı her bir küme için R_{ij} ile gösterilen küme indeksi, Denklem 2.11 ile hesaplanır (Aslanyürek, Mesut, 2021).

$$R_i = \max_{i \neq j} \left(\frac{\delta_i + \delta_j}{d_{ij}} \right) \tag{2.11}$$

Burada d_{ij} değeri i ve j kümelerinin merkezleri arasındaki uzaklığı, δ_i ve δ_j değerleri ise i. ve j. kümelerdeki gözlemlerin kendi küme merkezlerine olan ortalama uzaklığını temsil eder. Davies-Bouldin indeksi;

$$DB(k) = \frac{1}{k} \sum_{i=1}^{k} R_i \tag{2.12}$$

ile tanımlanmaktadır (Denklem 2.12) ve bu değeri minimum yapan k değeri en uygun küme sayısı olarak belirlenmektedir.

2.5.3. Calinski-Harabasz indeksi (CH indeks)

En uygun küme sayısının belirlenmesi için bir diğer yönteme ise CH indeks adı verilmiştir. Kümeler içi ve kümeler arası kareler toplamı esasına dayanan bir yöntemdir. CH İndeks değerini maksimuma ulaştıran küme sayısı, en uygun küme sayısıdır (Calinski ve Harabasz, 1974).

$$BSS(k) = \frac{1}{2} \sum_{i=1}^{k} \sum_{\substack{i \in c_i \\ j \notin c_i}} d(i,j) \tag{2.13}$$

$$\mathrm{WSS(k)} = \frac{1}{2}\sum_{i=1}^{k}\sum_{i,j\in c_i} d(i,j) \quad (2.14)$$

$$\mathrm{CH(k)} = \frac{\mathrm{BSS(k)/(k-1)}}{\mathrm{WSS(k)/(n-k)}} \quad (2.15)$$

Burada $\mathrm{BSS(k)}$ değeri kümeler arası kareler toplamını (Denklem 2.13), $\mathrm{WSS(k)}$ değeri kümeler içi kareler toplamını (Denklem 2.14), $\mathrm{CH(k)}$ değeri ise Calinski-Harabasz indeksi'ni (Denklem 2.15) ifade etmektedir.

2.5.4. Dirsek yöntemi ile küme sayısı belirleme

Kümeleme için gerekli küme sayısı Dirsek (Elbow) yöntemi kullanılarak tespit edilebilmektedir. Bu yöntem, k küme sayısını, her bir noktanın küme merkezine olan uzaklığının karelerinin toplamı (WCSS - kümeler içi kareler toplamı) ile hesaplamaktadır. Yönteme göre k sayısı, WCSS'deki değişim miktarının belirgin olarak azaldığı nokta olarak belirlenmekte ve bu noktaya dirsek noktası adı verilmektedir (Ketchen ve Shook, 1996). Dirsek noktasının en uygun k değeri olduğu söylenmektedir (Taşçı, Onan, 2016).

Dirsek noktasını bulabilmek için hesaplanan WCSS değerleri rastgele k değeri ile başlatılır. Bu yöntemde iki nokta arasındaki uzaklık, öklid mesafesi kullanılarak bulunur.

Düzlemde (M, N) ve (p, q) koordinatlarına sahip iki noktaya ait öklid uzaklığı Denklem 2.16 ile hesaplanmaktadır (Jain, Kashyap, 2021).

$$\mathrm{Uzk(M,N),(p,q)} = \sqrt{(M-p)^2 + (N-q)^2} \quad (2.16)$$

Bu işlem kümedeki tüm noktalar için hesaplanır ve ardından Denklem 2.17'de yer alan formül ile tek bir küme içerisindeki iki nesnenin minimum mesafesi bulunur.

$$\mathrm{Min}\left(m\sum_{m=1}^{m} B(C_m)\right) \quad (2.17)$$

Burada (Denklem 2.17) c_m ve B_{C_m} , sırası ile k'nıncı kümeyi ve küme içindeki varyasyonu ifade eder (Jain, Kashyap, 2021).

Dirsek noktasını belirlemek için aşağıdaki adımlar izlenir.

- k'nın farklı değerleri için örnek veri seti üzerinde K-Ortalamalar (K-Means) algortiması işletilir.
- Her k için Denklem 2.17 kullanılarak WCSS hesaplanır.
- Kümeler içi kareler toplamı (WCSS) ile k küme değerleri arasındaki eğri aracılığı ile dirsek noktasının belirlenmesi için grafik çizdirilir ve dirsek noktasına karar verilir.

2.6. Benzerlik Hesaplama Yöntemleri

N boyutlu vektör uzayında her bir metin belirli bir konuma sahiptir. Metinlerin konumlarını ifade eden vektörler kullanılarak metinlerin birbirlerine olan benzerlikleri hesaplanabilmektedir. Vektörler arasındaki benzerlikleri hesaplayabilmek için benzerlik ölçütü olarak kosinüs benzerliği kullanılmıştır.

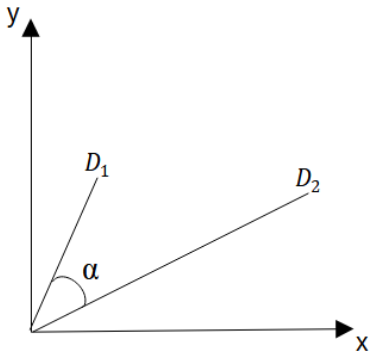
2.6.1. Kosinüs benzerliği (Cosine similarity)

Kosinüs benzerliği, doküman sınıflandırma, dokümanlar arasındaki benzerliği bulma gibi bilgi çıkarımı uygulamalarında en fazla kullanılan benzerlik ölçüm metodudur (Huang, 2008).

Kosinüs benzerliğinde vektörler arasındaki açı ile benzerlik oranı ters orantılıdır. Açı ne kadar küçük ise benzerlik o kadar yüksektir.

Şekil 2.8.'de D_1 ve D_2 metinlerinin vektör uzayındaki konumları gösterilmiştir. Denklem 2.18'de ise D_1 ve D_2 metin vektörlerinin arasındaki açının kosinüs değeri verilmektedir. D_1 ve D_2 metinlerine ait benzerlik oranı, kosinüs benzerliği ile hesaplanmaktadır. D_1 ve D_2 metinlerinin vektörleri arasındaki iç çarpım değerinin,

vektör uzunluklarının çarpım değerine oranı iki metin arasındaki benzerlik oranını vermektedir (Denklem 2.19).



Şekil 2.8. Kosinüs benzerliği (Cosine similarity)

$$\mathrm{Cos}(\mathrm{D}_1, \mathrm{D}_2) = \mathrm{Cos}(\alpha) \tag{2.18}$$

$$\mathrm{Cos}(\mathrm{D}_1, \mathrm{D}_2) = \frac{\mathrm{D}_1 \bullet \mathrm{D}_2}{\|\mathrm{D}_1\| * \|\mathrm{D}_2\|} \tag{2.19}$$

Metinler arasındaki ortak kelime sayısı, vektör uzunluklarını doğrudan etkilemektedir. Bu sebeple iki metin arasında ortak kelime ne kadar fazla ise vektör uzunlukları da o kadar yakın olacaktır. Dolayısıyla benzerlik oranları da artacaktır.

BÖLÜM 3. LİTERATÜR ARAŞTIRMASI VE ÖNCEKİ ÇALIŞMALAR

Son yıllarda yapay zeka alanında yaşanan gelişmeler ile derin öğrenme temelli doğal dil işleme ve makine öğrenmesi yöntemlerinin kullanımı, araştırmacıların ilgisini bu alana çekmeyi başarmıştır. Özellikle son 5 yıl içerisinde her geçen gün bu alanda yapılan çalışmalar ve araştırmacıların ilgili bilgi kaynaklarına erişim imkanlarının kolaylaşması ile literatürde yer alan çalışmaların çeşitliliği de artmıştır.

Doğal dil işleme (NLP) ve makine öğrenmesi (ML) algoritmalarının iç içe kullanılması, araştırmacıların metin madenciliği yötemlerine ait bakış açılarını genişletmiş, akademik çalışmalarda ve sektörel problemlerde bütünleşik çalışmalar ile yeni süreçler oluşturulmuş veya iyileştirilmiştir.

Doğal dil işleme ve makine öğrenmesi çalışmalarında, çözüm aranan problemlerin ve bu problemlere ait verilerin yapısı ve içeriği sebebi ile her problemde aynı yöntem başarılı sonuç veremeyebilmektedir. Bu sebeple bu yöntemlere ait çeşitli algoritmalar, ilgili probleme uygulanarak, birden fazla yöntemin karşılaştırmalı analizi yapılarak, en başarılı sonucu veren algortimalar ile probleme, veriye en uygun algoritmalar belirlenmektedir.

Bu çalışmada kullanılacak olan algoritmaların belirlenmesi için öncelikle araştırma problemine uygun, literatürde daha önce (öncelik son 5 yıl olmak üzere) yapılan çalışmalar incelenmiştir ve kullanılan yöntemler gruplandırılarak probleme uygun, en fazla tercih edilen üç adet NLP algoritması, üç adet kümeleme algoritması ve 3 adet kümeleme sonuçları değerlendirme algoritması belirlenmiştir.

Çalışmanın bu kısmında NLP yöntemleri ile metin temsilleri oluşturularak, ilgili metin temsillerinin makine öğrenmesi yöntemleri ile kümelendirilmesi ve kümelenen

temsillerden ilgili metine en benzer metinlerin adreslenmesi için literatür araştırması yapılmıştır.

2020 yılında ortaya çıkan Covid-19 virüsüne karşı üretilen aşılar konusunda insanlar, tedirginliklerini, meraklarını ve korkularını sosyal medya ortamlarında dile getirmiştir. 2021 yılında bir çalışmada ise Covid-19 aşısına yönelik kamuoyu algısını ortaya çıkarabilmek için konuya ilişkin tweetler, Twitter'dan alınarak Doc2vec metin temsil yöntemi ile vektörel temsilleri oluşturulmuş ve ana konu başlıklarının belirlenmesi için K-Ortalamalar tabanlı birden fazla kümeleme yöntemi kullanılarak dört ana konu başlığı belirlenmiştir. Kümeleme başarısının değerlendirilmesi için Gap istatistiği, Calinski-Harabasz, Silhoutte ve Davies Bouildin indeksleri kullanılmıştır (Wang, Kwok, 2021).

Yine aynı yılda yapılan bir çalışmada, Wikipedia makale özetlerinden oluşturdukları iki farklı veri kümesi kullanılarak, veri ön-işleme adımlarını gerçekleştirmiş, TF-IDF temsil yöntemi ile dokümanların vektörel karşılıkları elde edilmiş, K- Ortalamalar, K-Medoids ve CLARANS kümeleme yöntemleri ile kümeleme yapılmış, dahili yöntemler olarak belirttikleri Silhouette, Calinski-Harabasz ve Davies-Bouldin indekslerine ek olarak yeni önerilen kümeleme performans ölçüm yöntemini kullanılarak kümeleme yöntemlerinin performansları ve kümeleme başarıları değerlendirilmiştir. İlgili çalışmanın sonucunda, K-Ortalamalar algoritmasının en iyi kümeleme yöntemi olduğunu ardından K-Medoids'in ona yakın sonuçlar verdiğini, CLARANS'ın ise en kötü kümeleme yapan yöntem olduğunu, en hızlı kümeleme algoritması K-Ortalamalar, en hızlı kümeleme performansı ölçen yöntemin de önerilen yöntem olduğu, en hızlı kümeleme performansı ölçen dahili yöntemin ise Davies-Bouldin olduğunu gözlemlenmiştir (Aslanyürek ve Mesut, 2021).

Worl Wide Web (WWW)'in ürettiği büyük boyutlardaki veriler arasında benzer ilgi alanlarına sahip kullanıcıların, benzer bilgi erişimlerine dayalı olarak gruplandırılması için K-Ortalamalar kümeleme algoritması uygulanmış olup en uygun küme sayısı Dirsek Yöntemi ile belirlenmiştir (Jain ve Kashyap, 2021).

Türkçe haber kaynaklarından 6 farklı kategoriden elde ettikleri veri setini, çeşitli veri ön işlem aşamalarından geçirmiş TF-IDF, Word2vec ve Fasttext temsil yöntemleri ile ayrı ayrı vektörize etmiş ve Destek Vektör Makinesi (Support Vector Machine, SVM), Naive Bayes, Logistic Regression, Random Forest ve Yapay Sinir Ağı (Artificial Neural Network, ANN) yöntemlerini kullanarak sınıflandıran araştırmacılar, en başarılı sonucu FastText yöntemi ile elde ettikleri metin temsillerini %95,75 sınıflandırma başarısı ile SVM'de elde etmişlerdir (Çelik, Koç, 2021).

Ma ve ark. UC Berkeley ders programının 9205 dersinin açıklamalarını tarayarak, dersi oluşturan kategori, kod, ünite, başlık vb. değişkenleri alarak ders seçiminde anlamsal benzerlik analizi için bir veri kümesi oluşturmuştur. Oluşturdukları verileri ön işlem aşamalarından geçirdikten sonra TF-IDF, Word2vec ve Doc2vec ile metin temsillerini oluşturarak anlamsal benzerlik analizi yapmış ve başarı ölçmüşlerdir (Ma, Wang, Hu, Lu, 2017).

NLP ve makine öğrenmesi çalışmalarındaki en önemli aşama yapılandırılmamış verinin temizlenme ve ön işlem aşamalarıdır. Model başarısını doğrudan etkileyen bu aşamalardan biri de veriye uygun bir kütüphane aracılığı ile kelimelerin kökünü elde etmektir. Bu amaçla 2012 yılında iki araştırmacı, kök bulmanın yüksek boyutlu metin verilerindeki etkisini araştırmak için Zemberek, Ek Çıkaran ve Sabit Örnek 5 kök bulma yöntemlerini kullanarak, TF-IDF kelime temsil yöntemi ile metin verilerini vektörel hale getirmiş ve Türkçe metinlerin, kümeleme üzerindeki etkisini deneysel olarak araştırarak sonuçlarını ortaya koymuştur. Zemberek ve Sabit Örnek 5 yöntemlerinin Türkçe metin kümeleme için en uygun yöntemler olduğunu ve kök bulmanın kümeleme kalitesine ek olarak çok boyutlu metin verilerinde yüksek oranda boyut indirgeme sağladığını ortaya koymuştur (Tunalı ve Bilgin, 2012).

BÖLÜM 4. GÜNCEL YAKLAŞIMLARLA RAPORLARIN KÜMELENMESİ VE ADRESLENMESİ

Bu çalışmada, özel bir bankanın rapor taleplerine cevap verme süresini kısaltmak, rapor oluşturma maliyetini düşürmek, gelen talebi doğru kişiye iletmek amacı ile metin madenciliği süreçleri benimsenmiş olup, derin öğrenme temelli doğal dil işleme (NLP) ve makine öğrenmesi yöntemlerinden olan kümeleme çalışması ve benzerlik çalışması yapılmıştır.

Söz konusu X bankasının tüm iş birimlerinden, bilgi teknolojileri şirketine gelen iş talepleri A uygulamasında havuz mantığı ile toplanmaktadır.

Bankalar çok büyük yapılanması olan, yüzlerce ekibe bölünmüş organizasyona sahip olan kurumlardır. Dolayısıyla iş ihtiyacı çok büyük ve yönetilmesi zordur. İlgili bankada müşteriler tarafından oluşturulan talepler havuz uygulamalarda toplanmaktadır. Sonrasında ise talepler, ilgili ekiplere görev olarak dağılmaktadır.

Tüm dünyada olduğu gibi bankalar için veri her şeydir. Veriden bilgi elde edebilmek için veri çeşitli aşamalardan geçerek anlamlandırılmaktadır. İlgili bankanın iş birimlerinin ihtiyaçları doğrultusunda veriler anlamlandırılarak bilgi elde edilmesi için oluşturulan rapor talepleri, EDW ekibine görev olarak atanmaktadır.

Çalışmada, X bankasının ilgili iş birimlerinden EDW ekibine gelen rapor talepleri derin öğrenme temelli doğal dil işleme yöntemleri ile modellenmiştir ve ardından raporlar makine öğrenmesi yöntemleri ile kümelendirilerek ilgili rapor talebine benzeyen daha önceden oluşturulmuş rapor & raporlar var ise o rapor & raporların adreslenmesi sağlanmış olup, talep maliyeti ve içeriği konusunda daha fazla bilgi edinilmesi sağlanmıştır.

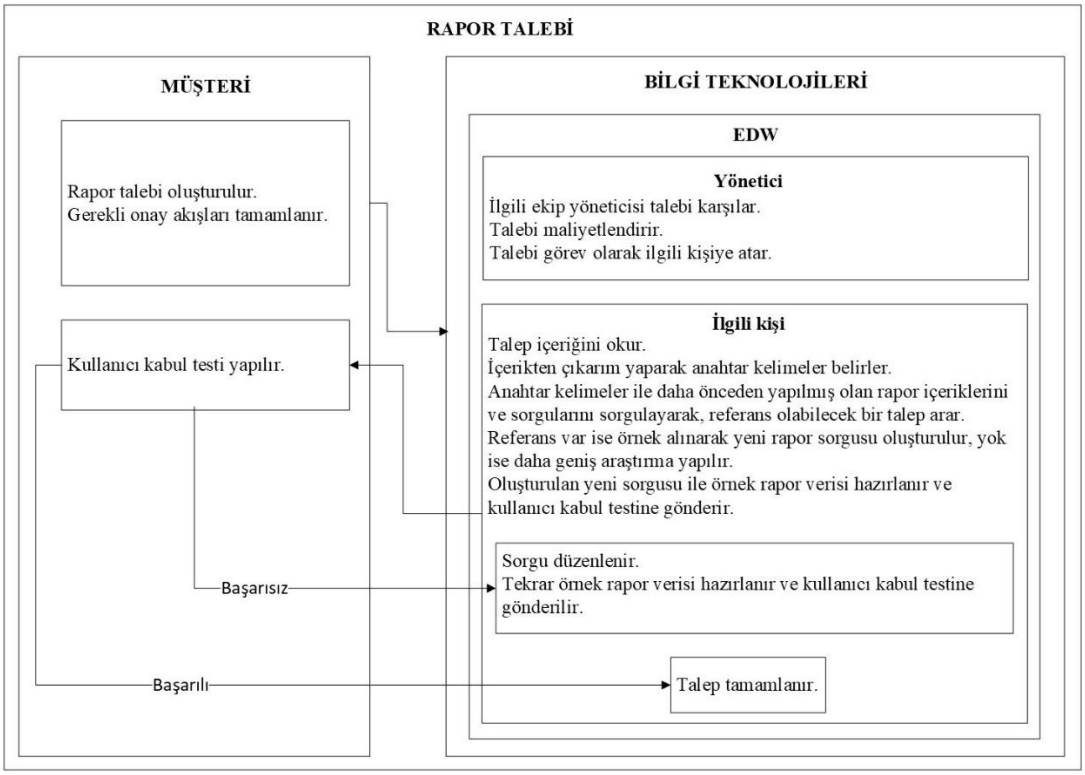
Çalışmanın yapıldığı tarihte günlük talep adedi incelendiğinde sadece EDW ekibine ortalama yaklaşık 6 (5.724) adet talep gelmektedir. Bu talep sayısı ayda ise ortalama yaklaşık 112 (112.279) adettir. Talep maliyetleri ise 1-5 gün ve 5+ şeklinde maliyetlendirilmektedir. 5+ olan talepler yazılım geliştirme ihtiyacı duyan taleplerdir.

Talep geliş akışındaki yoğunluk sebebi ile rapor ihtiyaçlarına cevap veren kişi sayısı ile talep maliyetleri zaman zaman orantısız bir hal alabilmektedir. Bu sebeple talep maliyetlerinin doğru verilebilmesi ve zamanında tamamlanabilmesi için de bu çalışmanın fayda getirmesi beklenmektedir.

Var olan sistem iş akışı:

1. İş birimi talep açar.
2. Talep onayları alınır.
3. Talep, bilgi teknolojilerinden ilgili ekibin iş listesine atanır.
4. İlgili ekip yöneticisi talebi karşılar ve maliyet vererek kişiye atamasını yapar.
5. Talebin atandığı kişi, ilgili talebe ait daha önceden yapılmış bir talep var ise o talebi araştırır. Araştırmayı ise gelen talebe ait bir fikir var ise buradan yola çıkılarak daha önceden yapılmış talep sorgularının ve detaylarının tutulduğu, şu ana kadar yapılan ve bankanın ilgili raporlama programında yer alan yaklaşık 200 bin adet kayıt içeren tabloları sorgular ve tek tek gözle bakarak yeni talebe ait rapor oluşturmaya çalışır.
6. Rapor sorgusu tamamlanır ve ihtiyaca göre rapor periyodik olarak çalışacak şekilde programlanır veya tek seferlik olarak talep sahibi iş birimine aktarır.

Bu akış esnasında başta yapılan maliyetlendirme bazen doğru olmayabilmektedir veya iş biriminin ek ihtiyaçları doğabilmektedir. Bu tarz durumlarda da rapor kapsamı ve maliyeti değişebilmektedir. Yapılan çalışma, ilgili raporların doğru maliyetlendirilmesini sağlayacak ve daha önce yapılmış olan benzer talepler kümelenip, belirlenen kurallar çerçevesinde adrselenecektir. Şekil 4.1.'de bir talebin süreci belirtilmiştir.

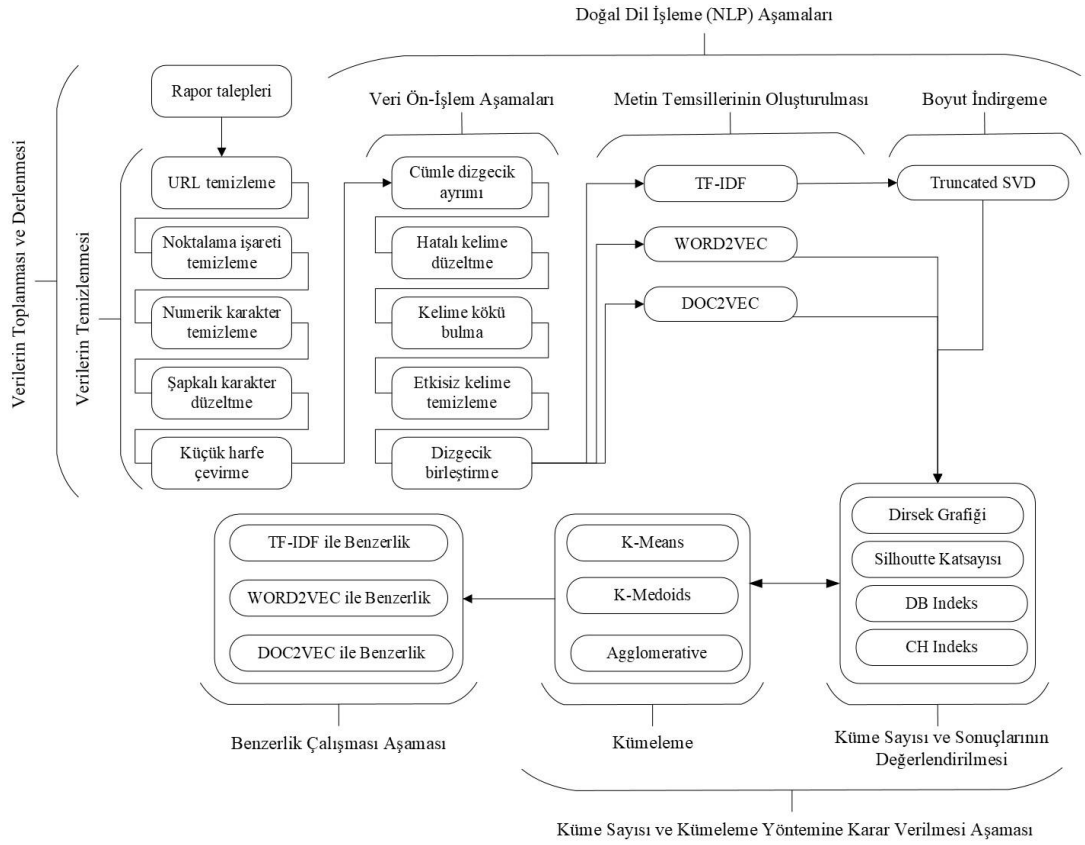


Şekil 4.1. Rapor talep akışı

Bu çalışmada bir derin öğrenme yöntemi olan NLP teknikleri kullanılarak veriler modellenmiş, makine öğrenmesi yöntemleri kullanılarak kümeleme ve benzerlik çalışması yapılmıştır.

Çalışmada, verilerin toplanması ve derlenmesi, veri temizleme ve ön-işlem aşamalarının gerçekleştirilmesi, verilerin sayısallaştırılması için yöntemlerin belirlenmesi ve uygulanması, küme sayısının & sonuçlarının değerlendirilmesi, kümeleme yöntemlerinin belirlenmesi ve uygulanması, dokümanlar arası benzerliğin bulunması için yöntemlerin belirlenmesi ve uygulanması aşamaları gerçekleştirilmiştir.

Bu başlık altında her bir ‘talep’, birer ‘metin’ veya ‘doküman’ olarak adlandırılacaktır. Çalışmaya ait belirlenen yöntemler ve uygulama aşamaları aşağıdaki bilgi şemasında yer almaktadır. İlgili çalışma süreçleri Şekil 4.2.’de gösterilmiştir.



Şekil 4.2. Finans sektöründe doğal dil işleme (NLP) ile rapor kümelendirme ve talep bazlı rapor önerileri oluşturma proje süreci

4.1. Önemli Parametreler ve Bileşenler

Çalışmanın yapıldığı tarihte, ilgili bankanın havuz sistem olarak adlandırılan web uygulaması kaynak veritabanı olarak Microsoft SQL Server kullanmaktadır. Çalışma kapsamında kullanılan veri setini içeren tablolar, data mart olarak adlandırılmakta olup ilgili bankanın veri ambarı sistemine ETL uygulaması ile beslenmiştir. İlgili bankanın veri ambarı sistemini besleyen veritabanı makinesi ise Oracle Exadata X9M'dir. Kaynak sistemden veriyi alan, uygun formata dönüştüren ve veri ambarına data mart şeklinde besleyen uygulama Oracle Data Integrator 11g sürümüdür. Bankanın kullanmış olduğu raporlama programı Microstrategy'dir. Bu çalışma ise Spyder editörü aracılığı ile 14240 adet talep bilgisi kullanılarak Python 3.8.8 sürümü kullanılarak modellenmiş ve görseller oluşturulmuştur.

Taleplere ait Tablo 4.1.'de yer alan öznitelikler, uzman kişiler tarafından belirlenmiş ve veri seti oluşturulmuştur.

Tablo 4.1. Veri setinde kullanılan öznitelikler ve açıklamaları

Öznitelik	Açıklama
Talep Numarası	İş birimi tarafından açılan talebe ait eşsiz numara
Talep Başlığı	İş birimi tarafından bildirilen talep başlığı
Talep Açıklaması	İş birimi tarafından bildirilen talep detay açıklaması
Talebi Açan Birim	Talep bildiren iş biriminin yer aldığı ekip adı
Talebi Açan Birim GMY	Talep bildiren iş biriminin yer aldığı ekibin bağlı olduğu GMY adı

Çalışmanın bundan sonraki kısımlarında her bir talep bir metin olarak kabul edilmiştir.

BÖLÜM 5. BULGULAR VE DEĞERLENDİRME

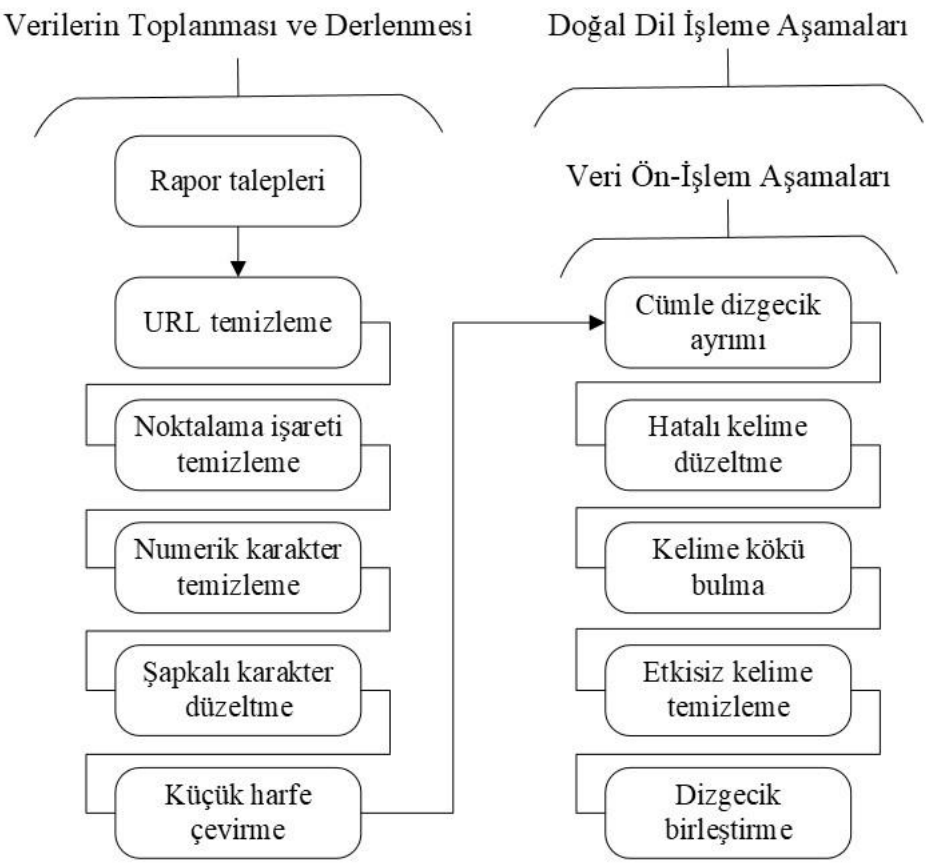
5.1. Problem Çözümü için Senaryolar

Çalışmanın bu kısmında araştırma süresince yapılan deneysel çalışmalar anlatılacak ve örnekler verilecektir.

5.1.1. Veri temizleme ve ön-işleme

Çalışmanın birinci aşamasında 14240 adet metin içeren veri seti detaylı incelenmiş olunup, ağırlıklı olarak pandas ve zemberek olmak üzere re, sys, logging, time, warnings kütüphaneleri aracılığı ile veri temizleme ve ön-işleme çalışması yapılmıştır.

Tüm senaryolarda kullanılacak olan verinin aynı olması sebebi ile bu aşama tüm senaryolarda ortak çalışılmıştır. Şekil 5.1.'de çalışmaya ait veri temizleme ve ön-işleme süreçleri gösterilmiştir.



Şekil 5.1. Veri temizleme ve ön-işlem süreçleri

Doğal dil işleme (NLP) çalışmalarında, veri temizleme aşamasında amaç, yapılan çalışma çerçevesinde veri setinin modele katkısı olmayacak değerlerden arındırılmasıdır. Bu sebeple ilgili veri seti URL'lerden, noktalama işaretlerinden, sayılardan, şapkalı harflerden arındırılırak tüm veri seti küçük harfe çevrilerek aynı formata getirilmiş ve ardından veri ön-işlem adımları uygulanmıştır.

Veri ön-işleme NLP çalışmalarının başarısını doğrudan etkileyebilecek bir aşamadır. Bu sebeple çalışmada veri ön-işlem aşamasına oldukça önem verilmiş olunup doğal dil ile yazılmış olan metinler üzerinde tokenizasyon işlemi Zemberek Kütüphanesi'nin 'TurkishTokenizer' modülü ile gerçekleştirilmiştir.

Çalışmada kullanılan Türkçe veri seti, havuz uygulamadan kaynak sisteme beslenirken İngilizce karakterler ile beslenmektedir. Örneğin; müşteri, musteri; yapılmıştır, yapilmistir vb. Yetki kısıtlamalarından dolayı kaynak sisteme müdahale edilememesi

sebebi ile bu durum, veri ön-işlem sürecinde, veri setinin normalize edilmesi ile çözümlenerek seviyede modele katkı sağlayacak en iyi duruma getirilmiştir. Çalışmada veri setinin normalize edilmesi Zemberek Kütüphanesi'nin 'TurkishSentenceNormalizer' modülü aracılığı ile gerçekleştirilmiştir.

Çalışmada kullanılan veri setini oluşturan dokümanlarda Türkçe – İngilizce kelimeler birlikte kullanılabilmektedir. Bu kullanımların sebebiyet verdiği bazı durumlar olabilmektedir. Kullanılan modülde sebebiyle, normalizasyon sırasında İngilizce kelimelere Türkçe kelime normalizasyonu yapmaktır. Sonucunda ise bankalarda "müşteri numarası" anlamında kalıplaşmış olan İngilizce "base" kelimesi, normalizasyon esnasında "başı" olarak, "login" kelimesi ise "ligin" olarak normalize edilmektedir. Base ve login kelime temsilleri ile başı ve ligin kelime temsilleri arasındaki fark göz ardı edilebilecek kadar küçük olduğu için çalışmada bu tarz durumlar göz ardı edilmiştir.

Normalize edilen veri setinde kök bulma işlemi Zemberek Kütüphanesi'nin 'TurkishSpellChecker' modülü ile gerçekleştirilmiştir. Çalışmada kök bulma ile değişken sayısı önemli ölçüde indirgenmiştir.

Çalışmada Zemberek Kütüphanesi'nin durak kelimelerine ek olarak, liste haline getirilmiş olan ilgili veri setinin içerdiği, modele katkısı olmayacak olan kelimeler de durak kelimelere eklenmiştir ve bu kelimeler veri setinden elenmiştir.

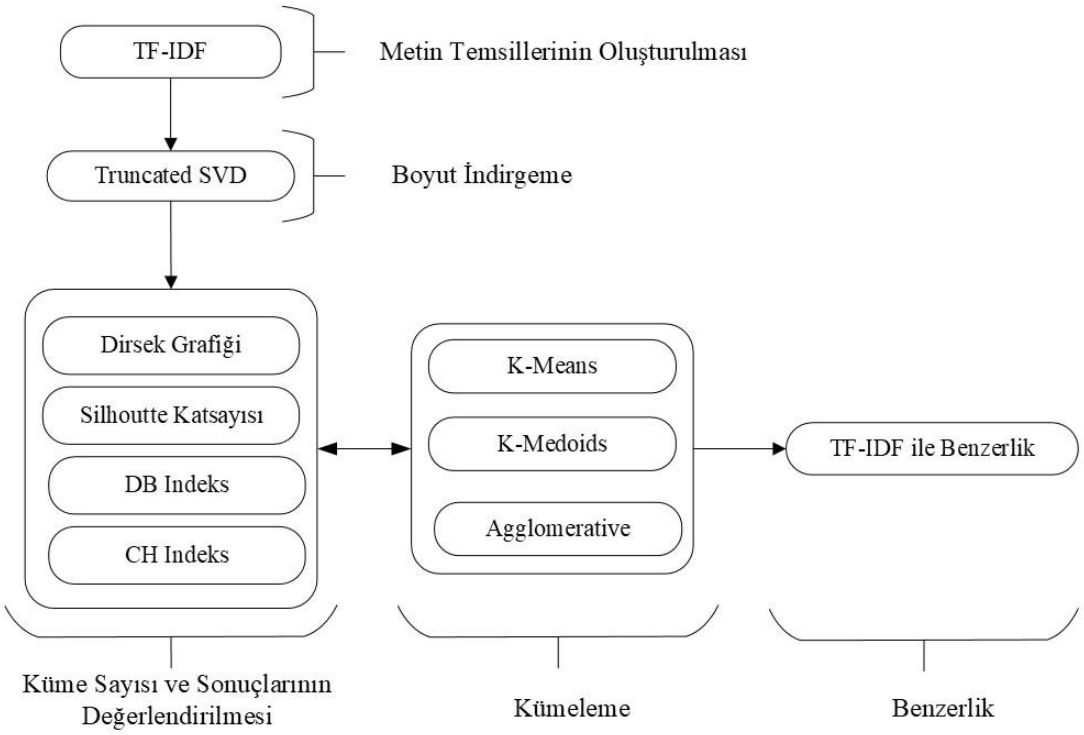
NLP çalışmalarında yüksek boyuttaki değişkenler model başarısını ve performansını olumsuz olarak etkilemektedir. Yapılan çalışmada veri temizleme ve ön-işlem aşamalarından sonra 9324 adet değişken sayısı 3504 adete düşürülmüştür. NLP çalışmalarının ilk aşaması olan veri temizleme ve ön-işlem süreçleri çok önemli olup incelenmesi ve çalışılması en önemli kısım olarak belirlenmiştir.

Çalışma, veri temizleme ve veri ön-işleme sonrası TF-IDF, Word2vec ve Doc2vec olmak üzere üç ana senaryo başlığı altında incelenmiştir.

5.1.2. TF-IDF

Kelimelerin cümle içerisinde geçme sıklıklarına bakılarak hesaplamalar yapılan TF-IDF frekans bazlı kelime temsil yöntemi ile oluşturulan senaryoda, boyut indirgeme, küme sayısı bulunması ve değerlendirilmesi, kümeleme ve benzerlik çalışması yapılmıştır.

Şekil 5.2.'de TF-IDF kelime temsil yöntemi temelinde gerçekleştirilen senaryonun süreçleri gösterilmiştir



Şekil 5.2. TF-IDF senaryo akışı

5.1.2.1. TF-IDF ile kelime kemsillerinin oluşturulması

Bu senaryoda, çalışmada kullanılan veri setini oluşturan metinler, TF-IDF kelime temsil yöntemi ile vektörel hale getirilmiştir. Kelime temsilleri, kümeleme çalışması için Sklearn Kütüphanesi'ne ait TfidfVectorizer modülü ile, benzerlik çalışması için ise Gensim Kütüphanesi'nin MatrixSimilarity modülü ile oluşturulmuştur.

5.1.2.2. Truncated SVD ile öznitelik indirgemesi

TF-IDF senaryosunda veri setine ait temsiller 14240 x 3504 boyutundadır. 3504 özniteliğe sahip veri setini girdi olarak alan modellerin sağlıklı çalışabilmesi mümkün değildir bu sebeple Sklearn kütüphanesine ait TruncatedSVD yöntemi kullanılarak 50 özniteliğe indirgenmiştir.

Öznitelik boytuna karar verme aşamasında igili veri setin için en uygun boyutu veren bir hesaplama mevcut değildir. Bu sebeple onlarca model denemesi yapılarak en uygun boyutun 50 olduğuna karar verilmiştir ve çalışmanın devamında 14240 x 50 boyutuna indirgenmiş ve matris haline getirilmiş temsiller kullanılmıştır.

5.1.2.3. Küme sayısı belirleme ve küme değerlendirme

TF-IDF ile temsilleri oluşturulan metinlere ait küme sayısı belirlenirken WCSS yöntemi ile hesaplanan değerlerin görselleştirildiği Dirsek Grafiği ile birlikte Silhoutte katsayısı, CH indeks, DB indeks kullanılmıştır.

5.1.2.4. Kümelendirme

TF-IDF ile elde edilen metin temsilleri Sklearn Kütüphenesi'ne ait K-Means, K-Medoids ve Agglomerative olmak üzere 3 farklı kümeleme algoritması ile kümelendirilerek en iyi sonuç veren kümeleme yöntemi belirlenmiştir.

Üç kümeleme algoritması hesaplamasında da uzaklık ölçütü olarak, denetimsiz öğrenme çalışmalarında en fazla kullanılan Öklid uzaklığı kullanılmıştır. Agglomerative kümeleme algoritması, kümeler arası benzerlik yöntemi olarak "ward" kullanılmıştır.

En uygun küme sayısı ve yöntemi belirlendikten sonra ilgili veri setini girdi olarak alan kümeleme yöntemi sonucunda oluşan kümeler ve bu kümelerin içerdiği metinlerin, metin numaraları bir küme matrisine beslenmiştir.

5.1.2.5. Benzerlik hesaplama fonksiyonunun oluşturulması

TF-IDF ile oluşturulan metin temsilleri arasındaki benzerliklerin bulunması için 2 aşamalı bir akış oluşturulmuştur.

1. Benzerlik matrisinin oluşturulması.
2. Aynı kümede yer alan en benzer metinlerin bulunması.

İlk aşamada benzerlik matrisi oluşturulurken Python programla dilinin sahip olduğu Gensim kütüphanesinin TF-IDF ile benzerlik matrisi hesaplayan MatrixSimilarity fonksiyonu kullanılmıştır. Bu fonksiyon, tokenize edilmiş veri setini girdi olarak alır, kosinüs benzerliği kullanarak her bir metnin diğer tüm metinler ile benzerliğini hesaplar ve 14240 x 14240 boyutunda bir benzerlik matrisi oluşturur.

Vektör uzayında yer alan vektörler yani metin temsilleri arasındaki açı ne kadar küçük ise benzerlik o kadar yüksektir.

Benzerlik matrisi, benzerlik oranlarını her bir indis bazında tutar. Çalışmada son kullanıcı için metin indisi anlamlı bir bilgi değildir. İlgili veri setindeki 0. indiste yer alan metinin diğer indislerdeki metinler ile benzerliği Tablo 5.1. TF-IDF benzerlik matirisi'nde verilmiştir. 0. indis metinin kendisidir.

Tablo 5.1. TF-IDF benzerlik matirisi

Metin indisi	Benzerlik oranı
0	1
1	0,009192
2	0,026556
..	..
14238	0,026844
14239	0,013173

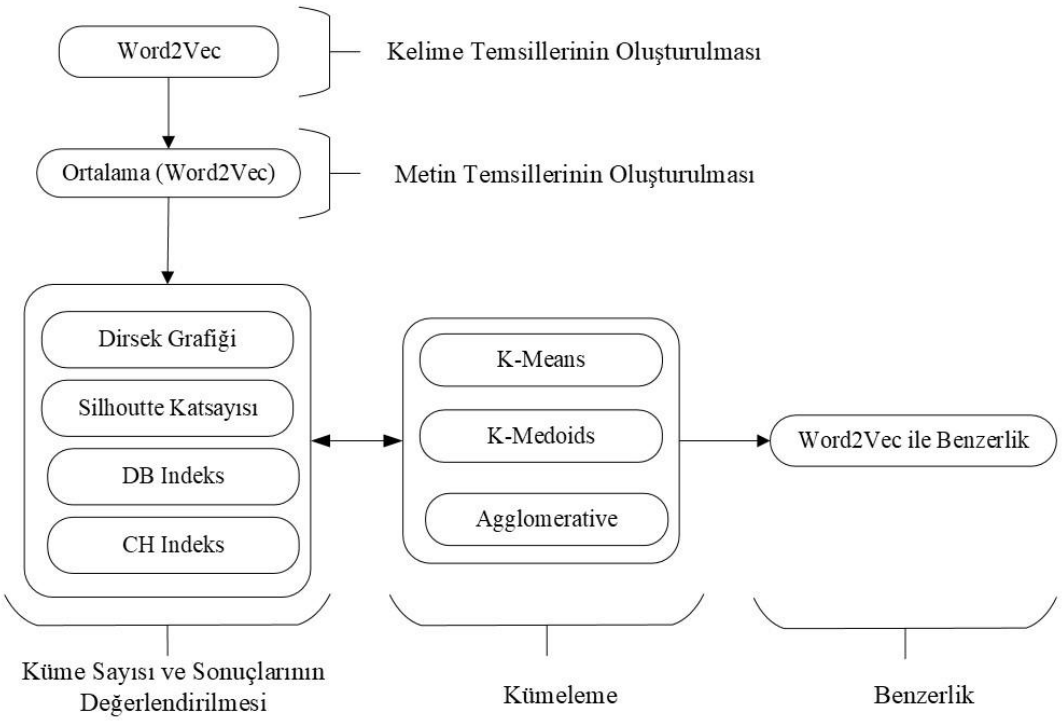
Bu sebeple ikinci aşamada hazırlanan fonksiyon, bir metine ait en benzer metinleri bulabilmek için, kullanıcıdan beş adet parametreyi girdi olarak alır. Bu parametreler, ilgili metine ait metin numarası, bir üst adımda oluşturulan benzerlik matrisi, minimum

benzer olması istenen benzerlik oranı, kümelendirme sonucunda her metnin, metin numarası ve kümesinin bulunduğu küme matrisi ve en benzer n sayıdaki metin adetidir. Bu parametreler ile benzerlik matrisinde indis bazında tutulan benzerlik oranları metin numaraları ile eşleştirilerek büyükten küçüğe doğru sıralanır ve istenen n sayıdaki aynı kümede yer alan benzer metinler, metin numaraları ve benzerlik oranları ile birlikte listelenir.

5.1.3. Word2vec

Kelimelerin cümle içerisindeki anlamsal benzerliklerine bakılarak hesaplamalar yapan Word2vec danışmasız ve tahminleme temelli kelime temsil yöntemi ile oluşturulan senaryoda, kelime vektörlerinin bulunması, her bir metine ait ortalama metin vektörlerinin bulunması, küme sayısı bulunması ve değerlendirilmesi, kümeleme ve benzerlik çalışması yapılmıştır.

Şekil 5.3.'de Word2vec kelime temsil yöntemi temelinde gerçekleştirilen senaryonun süreçleri gösterilmiştir.



Şekil 5.3. Word2vec senaryo akışı

5.1.3.1. Word2vec ile kelime ve metin temsillerinin oluşturulması

Bu senaryoda, çalışmada kullanılan veri setini oluşturan metinlerin kelime temsilleri, Gensim kütüphanesinin Word2vec modülü ile gerçekleştirilmiştir. Hiperparametrik olan bu yöntemde, ilgili veri seti için onlarca parametre denemesi yapılarak en uygun parametre değerlerine karar verilmiştir.

Word2vec ile metin vektörleri oluşturulması 2 aşamalıdır. İlk aşama, kelime vektörlerinin oluşturulması, ikinci aşama ise metin vektörlerinin oluşturulmasıdır.

Word2vec kelime temsil yönteminde, liste haline getirilmiş veri seti ve uygun parametre değerleri girdi olarak alınır ve istenen çıkış kelime veya kelimelerinin sağında ve solunda yer alan n sayıda komşu kelime ile model, anlamsal olarak eğitilir.

İlgili senaryoda, veri setinde geçen her bir kelime için Word2vec kelime temsil yönteminin CBoW algoritması ile 5 pencere boyutu parametresi ile eğitilip 5 boyutlu vektörler oluşturulmuştur.

Word2vec ile metin vektörleri oluşturmak için ise, bir üst adımda oluşturulan kelime vektörlerini girdi olarak alan, ilgili metinde geçen her bir kelimenin karşılığı vektörü bulan ve ilgili metinde yer alan kelime vektörlerinin ortalamalarını alarak, istenen vektör boyutunda (bu senaryo için 5) metin vektörü oluşturan bir fonksiyon hazırlanmıştır. Bu fonksiyon ile 14240 x 5 boyutunda metin temsillerini içeren bir matris oluşturulmuştur.

5.1.3.2. Küme sayısı belirleme ve küme değerlendirme

Word2vec ile temsilleri oluşturulan metinlere ait küme sayısı belirlenirken Dirsek grafiği ile birlikte Silhoutte katsayısı, CH indeks ve DB indeks kullanılmıştır.

5.1.3.3. Kümelendirme

Word2vec ile elde edilen metin temsilleri Sklearn Kütüphenesi'ne ait K-Means, K-Medoids ve Agglomerative olmak üzere 3 farklı kümeleme algoritması ile kümelendirilerek en iyi sonuç veren kümeleme yöntemi belirlenmiştir.

Üç kümeleme algoritması hesaplamasında da uzaklık ölçütü olarak, denetimsiz öğrenme çalışmalarında en fazla kullanılan Öklid uzaklığı kullanılmıştır. Agglomerative kümeleme algoritması, kümeler arası benzerlik yöntemi olarak "ward" kullanılmıştır.

En uygun küme sayısı ve yöntemi belirlendikten sonra ilgili veri setini girdi olarak alan kümeleme yöntemi sonucunda oluşan kümeler ve bu kümelerin içerdiği metinlerin, metin numaraları bir küme matrisine beslenir.

5.1.3.4. Benzerlik hesaplama

Word2vec ile oluşturulan metin temsilleri arasındaki benzerliklerin bulunması için 2 aşamalı bir akış oluşturulmuştur:

1. İstenen metine ait benzerlik matrisinin oluşturulması.
2. Aynı kümede yer alan en benzer metinlerin bulunması.

Benzerlik matrisinin oluşturulması için en benzer talepleri bulunması istenen metin numarası ile liste haline getirilmiş olan 14240 x 5 boyutundaki metin temsillerini girdi olarak alan bir fonksiyon oluşturulmuştur. Bu fonksiyon, Kosinüs benzerliği ile her bir metinin diğer tüm metinler ile benzerliğini hesaplayarak, benzerlik oranı bazında büyükten küçüğe doğru sıralayarak metin numarası ve benzerlik oranını tutan 14240 x 2 boyutunda bir benzerlik matrisi oluşturur. Bu matris ilgili metnin veri setinde yer alan tüm metinler ile benzerliğini tutar. Tablo 5.2.'de '155133' metin numarasına sahip metnin benzerlik matrisi verilmiştir.

Tablo 5.2. ‘155133’ numaralı metnin Word2vec benzerlik matrisi

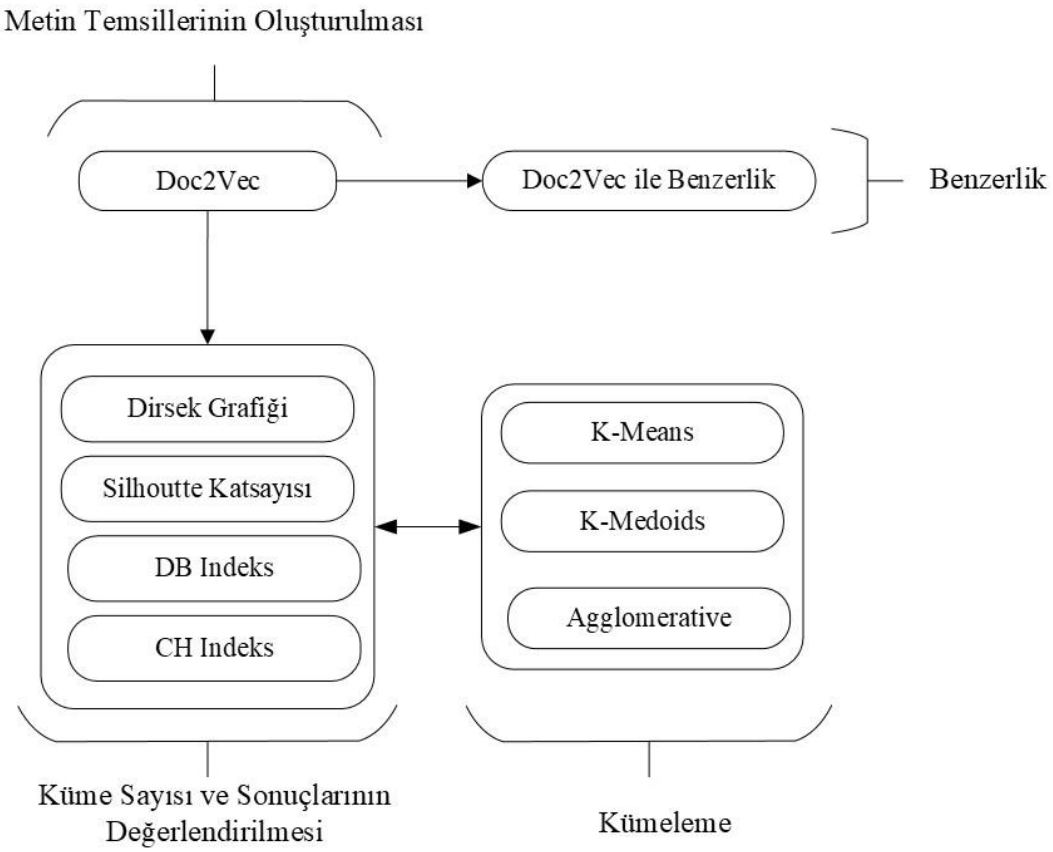
Metin numarası	Benzerlik oranı
155133	1
126821	0, 942630
91820	0, 917499
…	…
187210	-0,163494
169619	-0,179663

Aynı kümede yer alan metinlerin bulunması için benzerlik matrisi oluşturulmasının ardından kullanıcıdan alınan 5 adet parametre, ilgili fonksiyona girdi olarak verilir ve ilgili metine ait aynı kümede yer alan en benzer n adet metin, metin numarası ve benzerlik oranı ile listelenir. Bu parametreler, benzer talepleri bulunması istenen metin numarası, ilgili metine ait benzerlik matrisi, minimum benzerlik oranı, kümelendirme sonucunda her metnin, metin numarası ve kümesinin bulunduğu küme matrisi ve n sayıdaki metin adetidir.

5.1.4. Doc2vec

Kelimelerin cümle içerisindeki anlamsal benzerliklerine bakılarak hesaplamalar yapan Doc2vec danışmasız ve tahmin temelli paragraf temsil yöntemi ile oluşturulan senaryoda, kümeleme ve benzerlik metin vektörlerinin bulunması, küme sayısı bulunması ve değerlendirilmesi, kümeleme ve benzerlik çalışması yapılmıştır.

Şekil 5.4.’de Doc2vec kelime temsil yöntemi temelinde gerçekleştirilen senaryonun süreçleri gösterilmiştir.



Şekil 5.4. Doc2vec senaryo akışı

5.1.4.1. Doc2vec ile metin temsillerinin oluşturulması

Bu senaryoda, çalışmada kullanılan veri setini oluşturan metin temsilleri, Gensim kütüphanesinin Doc2vec modülü ile gerçekleştirilmiştir. Hiperparametrik olan bu yöntemde, ilgili veri seti için onlarca parametre denemesi yapılarak en uygun parametre değerlerine karar verilmiştir.

Doc2vec yönteminde her bir metin tekilliği sağlanan bir değere sahip olmalıdır. Doc2vec ile metin temsilleri oluşturulurken tekilliği sağlayan değer olarak karaktere çevrilmiş ‘metin numaraları’ kullanılmıştır. Metin numaralarının karakter formatında olması önemli bir husustur.

Bu metin temsil yönteminde, metin temsilleri oluşturulurken hedeflenen ne ise metin temsillerini o hedefe göre optimize etmek gerekmektedir. Örneğin hedef kümeleme

ise kümeleme için metin temsilleri oluşturmak, hedef benzerlik ise benzerlik için metin temsilleri oluşturmak gerekmektedir. Dolayısıyla bu teknikler veri setine ve hedeflere göre ölçeklendirilip, deneysel sonuçlar araştırmacı tarafından yorumlanmalıdır.

Çalışmada kullanılan veri seti 14240 adet metin ve 656644 adet kelime içermektedir. Model tarafından bu kelimeler arasından da nadir kullanılan kelimeler de çıkarılarak kelime adeti düşürülmektedir.

Doc2vec kelime temsil yöntemi ile ilgili literatürde küçük veri setleri üzerinde çalışmalar yapıldığı görülmüştür fakat bu yöntem esasen on milyonlarca kelime içeren veri setlerini eğitmek için tasarlanmıştır. Dolayısıyla her ne kadar küçük veri setlerinde de başarılı sonuçlar elde ediliyor olunsa da eğitim için büyük veri setleri kullanılması tavsiye edilmektedir.

Tüm bu durumlar göz önüne alındığında Doc2vec senaryosunda kümeleme çalışması esnasında ilgili veri seti ile üzerinde birçok hipermarametre denenerek en iyi kümeleme sonuçları PV-DBoW metodu 5 boyutta, en iyi benzerlik sonuçları ise PV-DBoW metodu 50 boyutta oluşturularak elde edilmiştir.

5.1.4.2. Küme sayısı belirleme ve küme değerlendirme

Doc2vec ile temsilleri oluşturulan metinlere ait küme sayısı belirlernirken Dirsek grafiği ile birlikte Silhoutte katsayısı, CH indeks ve DB indeks kullanılmıştır.

5.1.4.3. Kümelendirme

Doc2vec ile elde edilen metin temsilleri Sklearn Kütüphenesi'ne ait K-Means, K-Medoids ve Agglomerative olmak üzere 3 farklı kümeleme algoritması ile kümelendirilerek en iyi sonuç veren kümeleme yöntemi belirlenmiştir.

Üç kümeleme algoritması hesaplamasında da uzaklık ölçütü olarak, denetimsiz öğrenme çalışmalarında en fazla kullanılan Öklid uzaklığı kullanılmıştır.

Agglomerative kümeleme algoritması, kümeler arası benzerlik yöntemi olarak “ward” kullanılmıştır.

En uygun küme sayısı ve yöntemi belirlendikten sonra ilgili veri setini girdi olarak alan kümeleme yöntemi sonucunda oluşan kümeler ve bu kümelerin içerdiği metinlerin, metin numaraları bir küme matrisine beslenir.

5.1.4.4. Benzerlik hesaplama

Python programla dilinin sahip olduğu Gensim kütüphanesinin Doc2vec modülü kullanılarak metinlerin temsilleri oluşturulduktan sonra bu modülün sahip olduğu most_similar() fonksiyonu aracılığı ile girdi olarak verilen metine en benzer n adet metin listelenmektedir.

Doc2vec modülü most_similar() fonksiyonu benzerlik hesaplamasını, kosinüs benzerliğini temel alarak yapmaktadır.

Bu fonksiyon, girdi olarak ilgili metine ait etiketi (bu çalışma için metin numarası) parametre olarak alır ve istenilen n adet en benzer metini listeler. Tablo 5.3.’de ‘116421’ metin numarasına sahip metine ait en benzer 10 adet talep listelenmiştir

doc2vec_model.docvecs.most_similar(positive=['116421'], topn=10)

Tablo 5.3. ‘116421’ numaralı metnin Doc2vec benzerlik matrisi

Metin numarası	Benzerlik oranı
125137	0.944
125215	0,89
125171	0, 884
…	…
128758	0, 818
132660	0,817

5.2. Senaryo Sonuçları

Bu çalışma için belirlenen senaryolar ve senaryo sonuçları bu başlık altında paylaşılmıştır.

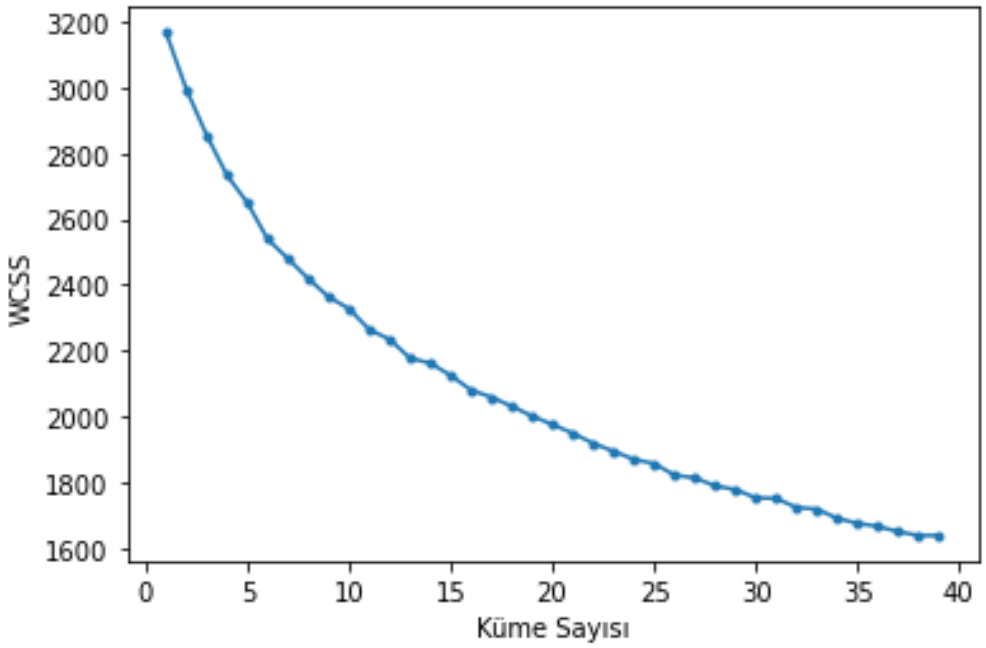
5.2.1. TF-IDF tablolar ve grafikler

TF-IDF ile oluşturulan metin temsilleri K-Means, K-Medoids ve Aglomerative kümeleme yöntemleri ile kümelenmiştir. TF-IDF ile oluşturulan metin temsilleri için en başarılı kümeleme yöntemini ve küme sayısını belirlemek için Dirsek Grafiği, Silhouette Katsayısı, Davies-Bouldin İndeksi ve Calsinki-Harabasz İndeksi kullanılmıştır.

14240 x 50 boyuta sahip TF-IDF ile oluşturulan metin temsilleri için elde edilen sonuçlara ait sayısal değerler Tablo 5.4.'de verilmiştir.

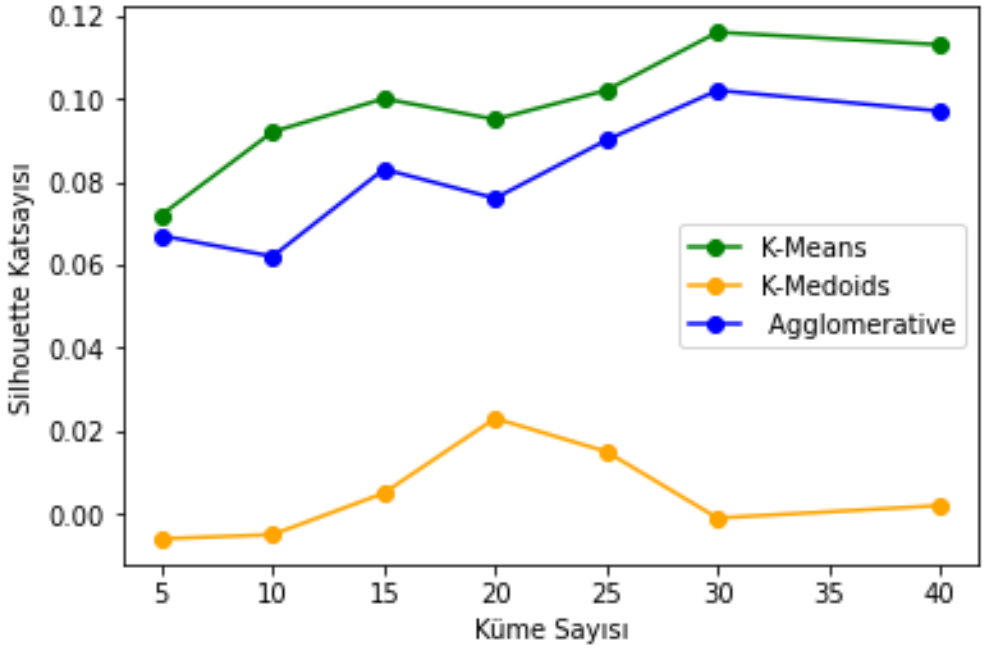
Tablo 5.4. TF-IDF kümeleme yöntemleri değerlendirme metrikleri sonuçları

	Silhoutte			DB			CH		
Küme Adeti	K-Means	K-Medoids	Agglomerative	K-Means	K-Medoids	Agglomerative	K-Means	K-Medoids	Agglomerative
5	0,072	-0,006	0,067	2,88	6,055	2,827	695,726	178,379	620,755
10	0,092	-0,005	0,062	2,569	4,925	2,823	570,996	194,935	496,642
15	0,1	0,005	0,083	2,419	4,287	2,572	498,936	217,81	427,46
20	0,095	0,023	0,076	2,232	3,772	2,365	451,789	244,269	388,402
25	0,102	0,015	0,09	2,058	3,769	2,283	418,301	201,353	361,056
30	0,116	-0,001	0,102	1,955	3,72	2,128	395,738	177,571	339,814
40	0,113	0,002	0,097	2,028	3,369	2,079	348,891	176,778	301,672



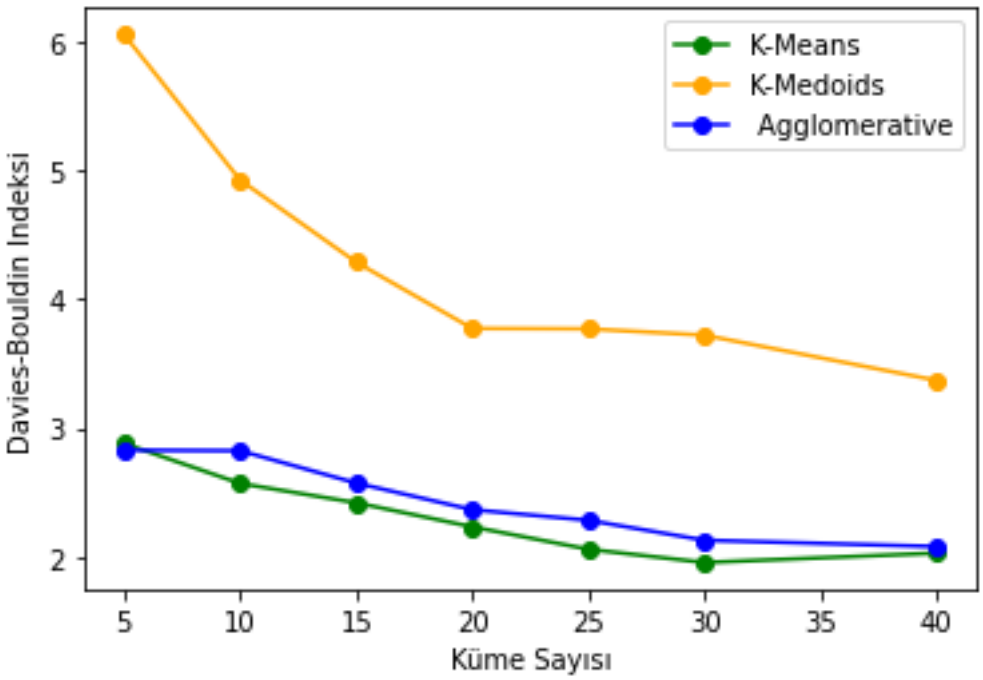
Şekil 5.5. TF-IDF Dirsek Grafiği

Dirsek Grafiği'nde eğrinin yatay eksene paralel olmaya başladığı küme sayısı en uygun küme sayısı olarak belirlenmektedir. Şekil 5.5.'de yer alan eğrinin, yatay eksene paralel hale gelmeye başladığı küme adeti aralığı 30-35 olarak görülmektedir. Dolayısıyla Dirsek grafiği bize küme sayısının 30-35 adet aralığında olduğunu göstermektedir.



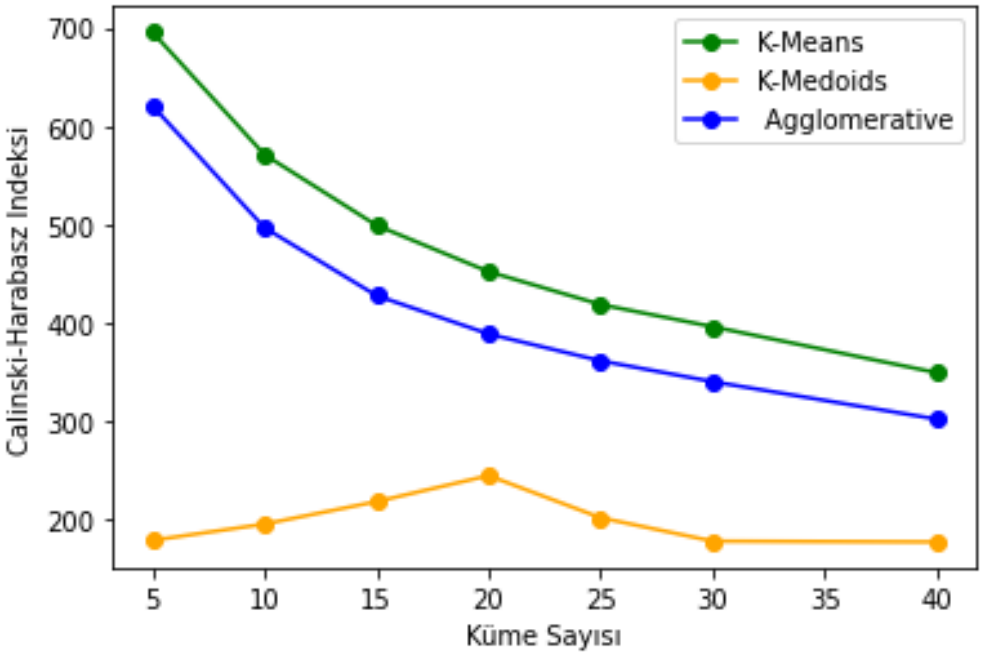
Şekil 5.6. TF-IDF Silhoutte Katsayısı

Silhoutte katsayısını maksimum değere ulaştıran kümeleme yöntemi ve küme sayısı, TF-IDF ile oluşturulan metin temsilleri için en uygun kümeleme yöntemini ve küme sayısını vermektedir. Şekil 5.6.'da görüldüğü üzere TF-IDF ile oluşturulan metin temsilleri için Silhoutte katsayısı, K-Means kümeleme yönteminde, 30 küme adetinde maksimum değere ulaşmıştır.



Şekil 5.7. TF-IDF DB Indeks

DB indeksin minimuma ulaştığı kümeleme yöntemi ve küme sayısı, ilgili veri seti için en uygun küme sayısını vermektedir. Şekil 5.7.'de görüldüğü üzere K-Means, DB indeksi 30 küme adetinde minimum değere ulaştıran kümeleme yöntemi olmuştur.



Şekil 5.8. TF-IDF CH Indeks

CH Indeksin maksimum değere ulaştığı kümeleme yöntemi ve küme sayısı en uygun küme sayısını vermektedir. Şekil 5.8.'de CH indeksi maksimuma ulaştıran kümeleme yöntemi K-Means ve 5 küme adeti olarak görülmektedir.

Literatür araştırmasında gözlemlenen sonuçlardan biri, ilgili veri seti için her kümeleme ve kümeleme değerlendirme yönteminin paralel sonuçlar vermeyebileceği dolayısyla her yöntemin her veri seti için uygun olmadığı, bu sebepten de veri setleri için birden fazla yöntemin denenerek en uygun sonuçları veren yöntemlerin çalışmada kullanılması gerektiğidir.

Bu çalışmanın TF-IDF senaryosu için elde edilen sonuçlar, olşturulan metin temsilleri için Dirsek Grafiği, Silhoutte Katsayısı ve DB Indeks'in paralel sonuçlar vererek çalışma için kullanılacak metrikler olduğu, CH Indeks'in ise uygun metrik olmadığıdır.

K-Medoids kümeleme yöntemi için grafikler incelendiğinde Silhoutte katsayısının negatif değerlere sahip olması, metinleri doğal kümelerine yerleştirmede başarısız olduğunu göstermektedir. Bu senaryo için uygun bir kümeleme yöntemi değildir.

Agglomerative kümeleme yöntemi ise K-Means kümeleme yöntemi ile paralel sonuçlar vermiştir fakat metrik değerleri incelendiğinde K-Means'in daha iyi sonuçlar verdiği görülmüştür.

TF-IDF ile oluşturulan metin temsilleri için en uygun kümeleme yönteminin 30 kümede K-Means olduğu sonucuna varılmıştır.

Tablo 5.5.'de ise K-Means, K-Medoids ve Agglomerative kümeleme yöntemlerinin, 14240 x 50 boyutlu TF-IDF ile oluşturulan metin temsilleri için kullanılan küme sayısı belirleme ve değerlendirme metriklerinin 5-40 küme arası toplam hesaplanma süreleri gösterilmiştir.

Tablo 5.5. TF-IDF ile her kümeleme yöntemi için metriklerin toplam tamamlanma süreleri

	Süre (dk)
Dirsek Grafiği	1.045
Silhoutte Katsayısı	3.124
DB Indeks	2.106
CH Indeks	2.051

Tablo 5.6.'da ilgili veri seti için her bir talebin diğer talepler ile benzerliklerinin bulunduğu matris ve en iyi kümeleme yöntemi olarak 30 kümede K-Means ile '204412' no'lu talebe ait %80 üzerinde benzerliği olan ilk 10 adet talep ve benzerlik oranı listelenmiştir. İlgili metinlerin kümeleri ve metin içerikleri incelendiğinde farklı tarihlerde oluşturulmuş aynı içeriğe sahip metinler olduğu dolayısıyla aynı kümede yer aldıkları gözlemlenmiştir.

Tablo 5.6. TF-IDF ile '204412' no'lu metine ait en benzer metinler

Metin numarası	Benzer metin numarası	Benzerlik oranı
204412	197657	1
204412	178305	1
204412	209481	1
204412	192196	1
204412	220841	1
204412	173667	1
204412	227072	1
204412	214347	1
204412	187700	1
204412	165614	0,972322

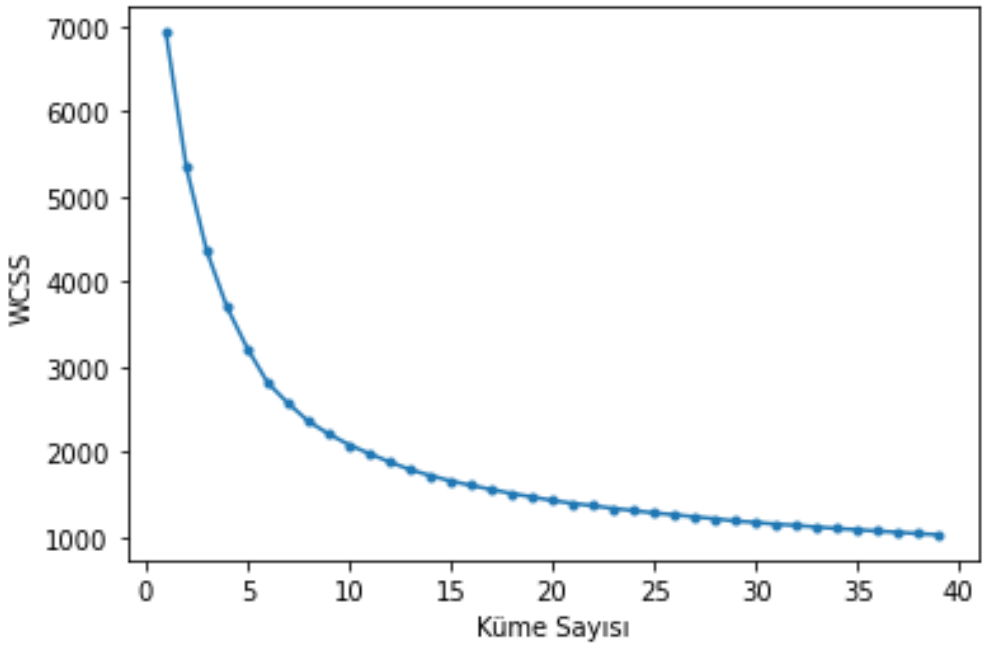
5.2.2. Word2vec tablolar ve grafikler

Word2vec ile oluşturulan metin temsilleri K-Means, K-Medoids ve Aglomerative kümeleme yöntemleri ile kümelenmiştir. Word2vec ile oluşturulan metin temsilleri için en başarılı kümeleme yöntemini ve küme sayısını belirlemek için Dirsek Grafiği, Silhouette Katsayısı, Davies-Bouldin İndeksi ve Calsinki-Harabasz İndeksi kullanılmıştır.

14240 x 5 boyuta sahip Word2vec ile oluşturulan metin temsilleri için elde edilen sonuçlara ait sayısal değerler Tablo 5.7.'de verilmiştir.

Tablo 5.7. Word2vec kümeleme yöntemleri değerlendirme metrikleri sonuçları

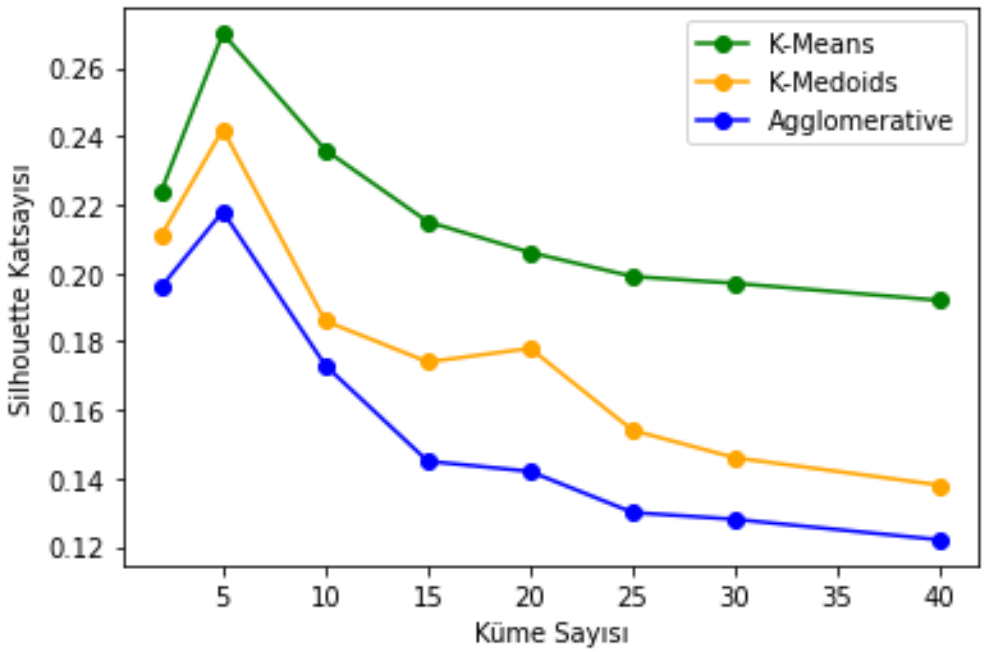
	Silhoutte			DB			CH		
Küme Adeti	K-Means	K-Medoids	Agglomerative	K-Means	K-Medoids	Agglomerative	K-Means	K-Medoids	Agglomerative
2	0,224	0,211	0,196	1,625	1,696	1,645	4236,542	3827,461	3578,215
5	0,27	0,242	0,218	1,163	1,246	1,257	4112,559	3620,331	3442,917
10	0,236	0,186	0,173	1,17	1,299	1,314	3681,912	3248,173	2935,171
15	0,215	0,174	0,145	1,22	1,46	1,373	3236,078	2782,612	2595,417
20	0,206	0,178	0,142	1,199	1,302	1,43	2886,132	2676,422	2334,909
25	0,199	0,154	0,13	1,262	1,352	1,463	2607,357	2184,555	2105,881
30	0,197	0,146	0,128	1,26	1,317	1,49	2409,685	1923,002	1940,651
40	0,192	0,138	0,122	1,207	1,337	1,449	2125,903	1690,259	1717,214



Şekil 5.9. Word2vec Dirsek Grafiği

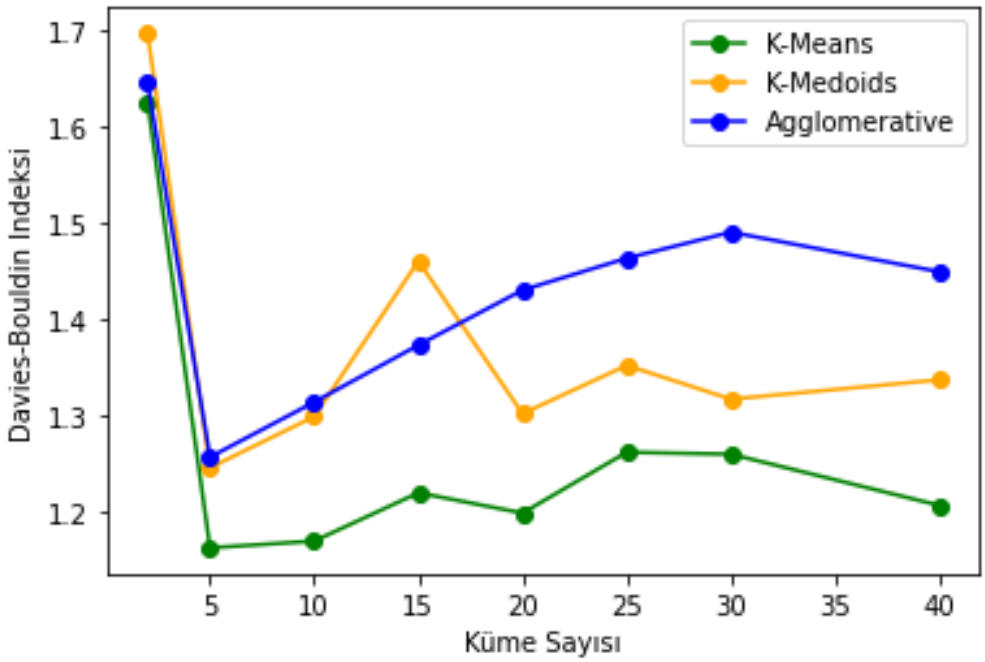
Dirsek grafiğinin yatay eksene paralel olmaya başladığı nokta 'dirsek noktası' olarak adlandırılır. Word2vec senaryosu için Şekil 5.9.'de 5 küme sayısında yatay eksene paralel hale gelmeye başladığı görülmektedir. Bu senaryo için küme sayısını Dirsek

grafiği 5–15 arasında göstermektedir. 15 küme sayısından sonra artan küme sayılarında eğrinin yatay eksene paralel olması durumundaki değişimler çok küçüktür dolayısıyla küme sayısı bu noktadan sonra değişmemektedir.



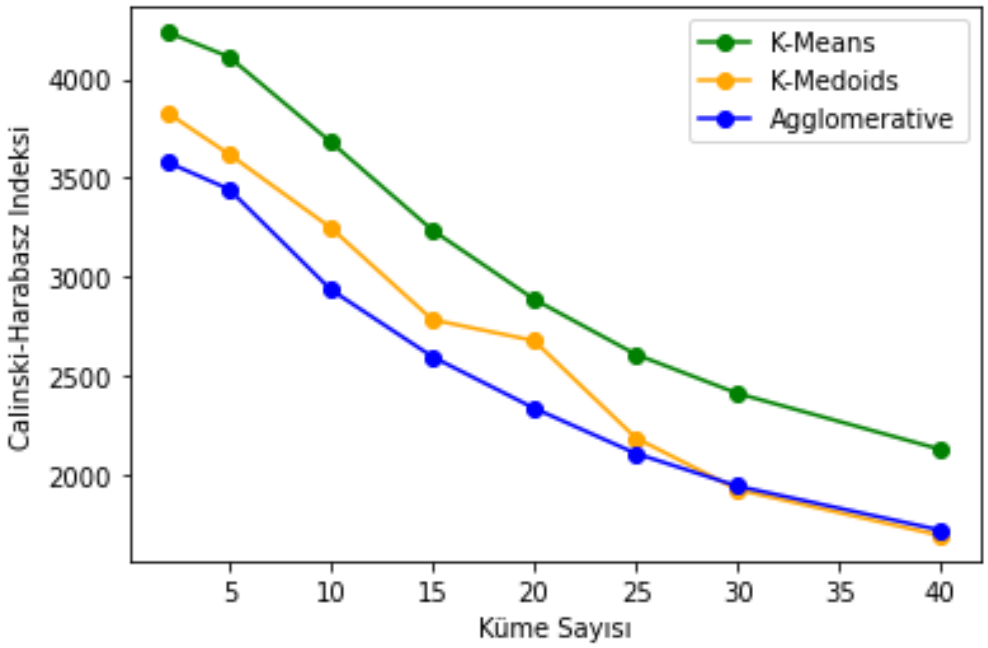
Şekil 5.10. Word2vec Silhoutte Katsayısı

Silhoutte katsayısını maksimuma ulaştıran kümeleme yöntemi ve küme sayısı, en uygun yöntem ve küme sayısını vermektedir. Şekil 5.10.'da Word2vec ile oluşturulan metin temsilleri için en iyi kümeleme yöntemi K-Means, küme sayısı ise 5'tir.



Şekil 5.11. Word2vec DB Indeks

DB indeks değerini minimuma ulaştıran kümeleme yöntemi ve küme sayısı ise en uygun kümeleme yöntemi ve küme sayısını vermektedir. Şekil 5.11.'de Word2vec ile oluşturulan metin temsilleri için DB indeks değerini minimuma ulaştıran kümeleme yöntemi K-Means, küme sayısı ise 5'tir.



Şekil 5.12. Word2vec CH Indeks

CH indeks değerini maksimuma ulaştıran kümeleme yöntemi ve küme sayısı en uygun kümeleme yöntemi ve sayısını vermektedir. Şekil 5.12.'de Word2vec ile oluşturulan metin temsilleri için en iyi kümeleme yöntemi K-Means, küme sayısı ise 2'dir.

Word2vec ile oluşturulan metin temsilleri için en uygun kümeleme yöntemi K-Means, küme sayısı ise 5 olarak değerlendirilmiştir.

Tablo 5.8.'de ise K-Means, K-Medoids ve Agglomerative kümeleme yöntemlerinin, 14240 x 5 boyutlu Word2vec ile oluşturulan metin temsilleri için kullanılan küme sayısı belirleme ve değerlendirme metriklerinin 2-40 küme arası toplam hesaplanma süreleri gösterilmiştir.

Tablo 5.8. Word2vec ile her kümeleme yöntemi için metriklerin toplam tamamlanma süreleri

	Süre (dk)
Dirsek Grafiği	2.655
Silhoutte Katsayısı	3.656
DB Indeks	1.923
CH Indeks	2.005

Tablo 5.9.'da ilgili veri seti için her bir talebin diğer talepler ile benzerliklerinin bulunduğu matris ve en iyi kümeleme yöntemi olarak 5 kümede K-Means ile '204412' no'lu talebe ait %80 üzerinde benzerliği olan ilk 10 adet talep ve benzerlik oranı listelenmiştir. İlgili metinlerin kümeleri ve metin içerikleri incelendiğinde farklı tarihlerde oluşturulmuş aynı içeriğe sahip metinler olduğu dolayısıyla aynı kümede yer aldıkları gözlemlenmiştir.

Tablo 5.9. Word2vec ile '204412' no'lu metine ait en benzer metinler

Metin numarası	Benzer metin numarası	Benzerlik oranı
204412	197657	1
204412	173667	1
204412	178305	1
204412	209481	1
204412	220841	1
204412	187700	1
204412	214347	1
204412	192196	1
204412	227072	1
204412	183177	0,999842

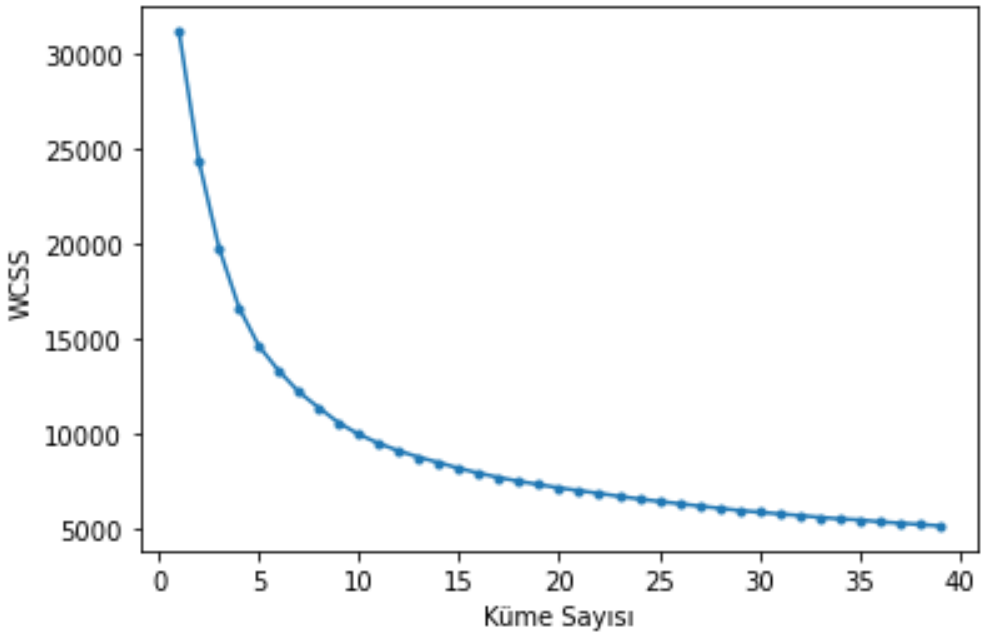
5.2.3. Doc2vec tablolar ve grafikler

Doc2vec ile kümeleme için oluşturulan metin temsilleri K-Means, K-Medoids ve Aglomerative kümeleme yöntemleri ile kümelenmiştir. Doc2vec ile oluşturulan metin temsilleri için en başarılı kümeleme yöntemini ve küme sayısını belirlemek için Dirsek Grafiği, Silhouette Katsayısı, Davies-Bouldin İndeksi ve Calsinki-Harabasz İndeksi kullanılmıştır.

14240 x 5 boyuta sahip Doc2vec ile oluşturulan metin temsilleri için elde edilen sonuçlara ait sayısal değerler Tablo 5.10.'de verilmiştir.

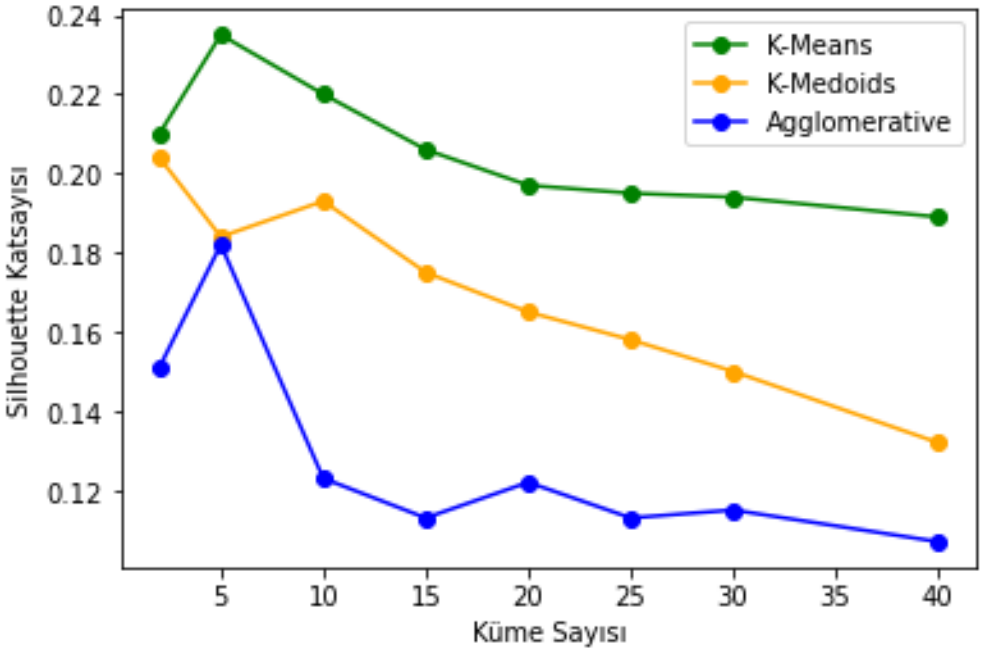
Tablo 5.10. Doc2vec kümeleme yöntemleri değerlendirme metrikleri sonuçları

	Silhoutte			DB			CH		
Küme Adeti	K-Means	K-Medoids	Agglomerative	K-Means	K-Medoids	Agglomerative	K-Means	K-Medoids	Agglomerative
2	0,21	0,204	0,151	1,735	1,744	1,773	4016,871	3887,629	2761,279
5	0,235	0,184	0,182	1,218	1,418	1,414	4049,802	3329,718	3036,694
10	0,22	0,193	0,123	1,346	1,437	1,45	3380,444	3167,793	2496,937
15	0,206	0,175	0,113	1,307	1,301	1,531	2872,673	2628,034	2145,998
20	0,197	0,165	0,122	1,245	1345	1,509	2529,066	2307,8	1934,716
25	0,195	0,158	0,113	1,259	1,355	1,477	2291,062	2025,626	1762,579
30	0,194	0,15	0,115	1223	1,374	1,468	2126,745	1873,763	1637,412
40	0,189	0,132	0,107	1,239	1,367	1,508	1884,823	1587,577	1455,918



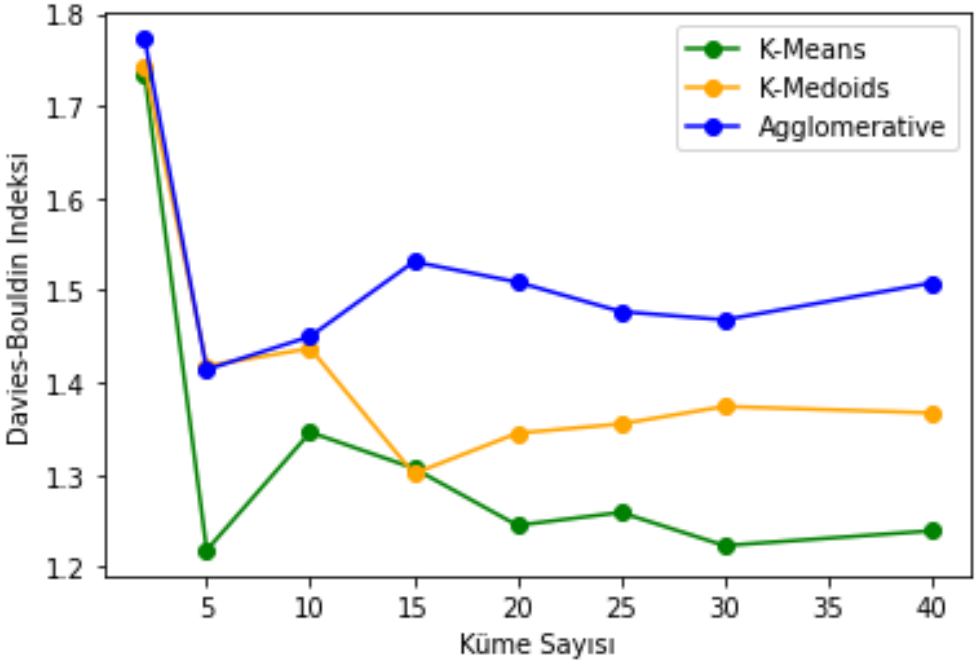
Şekil 5.13. Doc2vec Dirsek Grafiği

Dirsek grafiğinin yatay eksene paralel olmaya başladığı nokta ‘dirsek noktası’ olarak adlandırılır. Doc2vec senaryosu için Şekil 5.13.’de 5 küme sayısında yatay eksene paralel hale gelmeye başladığı görülmektedir. Bu senaryo için küme sayısını Dirsek grafiği 5–15 arasında göstermektedir. 15 küme sayısından sonra artan küme sayılarında eğrinin yatay eksene paralel olması durumundaki değişimler çok küçüktür dolayısıyla küme sayısı bu noktadan sonra değişmemektedir.



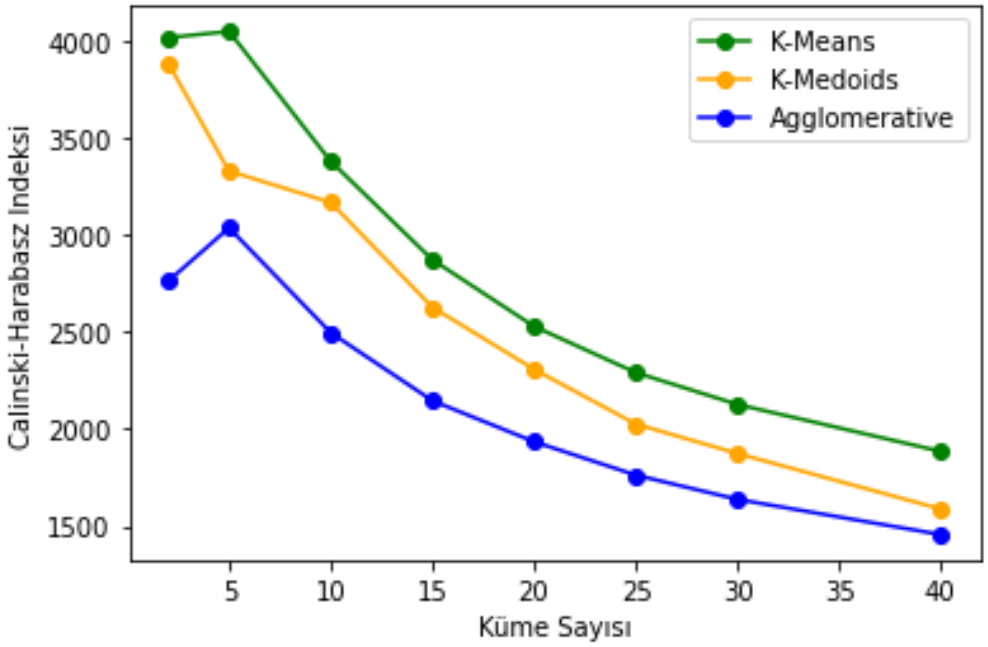
Şekil 5.14. Doc2vec Silhoutte Katsayısı

Silhoutte katsayısını maksimuma ulaştıran kümeleme yöntemi ve küme sayısı, en uygun yöntem ve küme sayısını vermektedir. Şekil 5.14.'da Doc2vec ile oluşturulan metin temsilleri için en iyi kümeleme yöntemi K-Means, küme sayısı ise 5'tir.



Şekil 5.15. Doc2vec DB Indeks

DB indeks değerini minimuma ulaştıran kümeleme yöntemi ve küme sayısı ise en uygun kümeleme yöntemi ve küme sayısını vermektedir. Şekil 5.15.'da Doc2vec ile oluşturulan metin temsilleri için DB indeks değerini minimuma ulaştıran kümeleme yöntemi K-Means, küme sayısı ise 5'tir.



Şekil 5.16. Doc2vec CH Indeks

CH indeks değerini maksimuma ulaştıran kümeleme yöntemi ve küme sayısı en uygun kümeleme yöntemi ve sayısını vermektedir. Şekil 5.16.'de Doc2vec ile oluşturulan metin temsilleri için en iyi kümeleme yöntemi K-Means, küme sayısı ise 5'tir.

Doc2vec ile oluşturulan metin temsilleri için en uygun kümeleme yöntemi K-Means, küme sayısı ise 5 olarak değerlendirilmiştir.

Tablo 5.11.'de ise K-Means, K-Medoids ve Agglomerative kümeleme yöntemlerinin, 14240 x 5 boyutlu Doc2vec ile oluşturulan metin temsilleri için kullanılan küme sayısı belirleme ve değerlendirme metriklerinin 2-40 küme arası toplam hesaplanma süreleri gösterilmiştir.

Tablo 5.11. Doc2vec ile her kümeleme yöntemi için metriklerin toplam tamamlanma süreleri

	Süre (dk)
Dirsek Grafiği	1.606
Silhoutte	3.484
DB	1.936
CH	1.983

Tablo 5.12.'de 14240 x 50 boyutlu Doc2vec ile benzerlik için oluşturulmuş metin temsillerinden '204412' no lu metine en benzer 10 adet metin listlenmiştir.

Tablo 5.12. Doc2vec ile '204412' no'lu metine ait en benzer metinler

Metin numarası	Benzer metin numarası	Benzerlik oranı
204412	197657	0,995
204412	220841	0,995
204412	214347	0,995
204412	173667	0,994
204412	160749	0,994
204412	192196	0,994
204412	169790	0,994
204412	178305	0,993
204412	209481	0,993
204412	165614	0,993

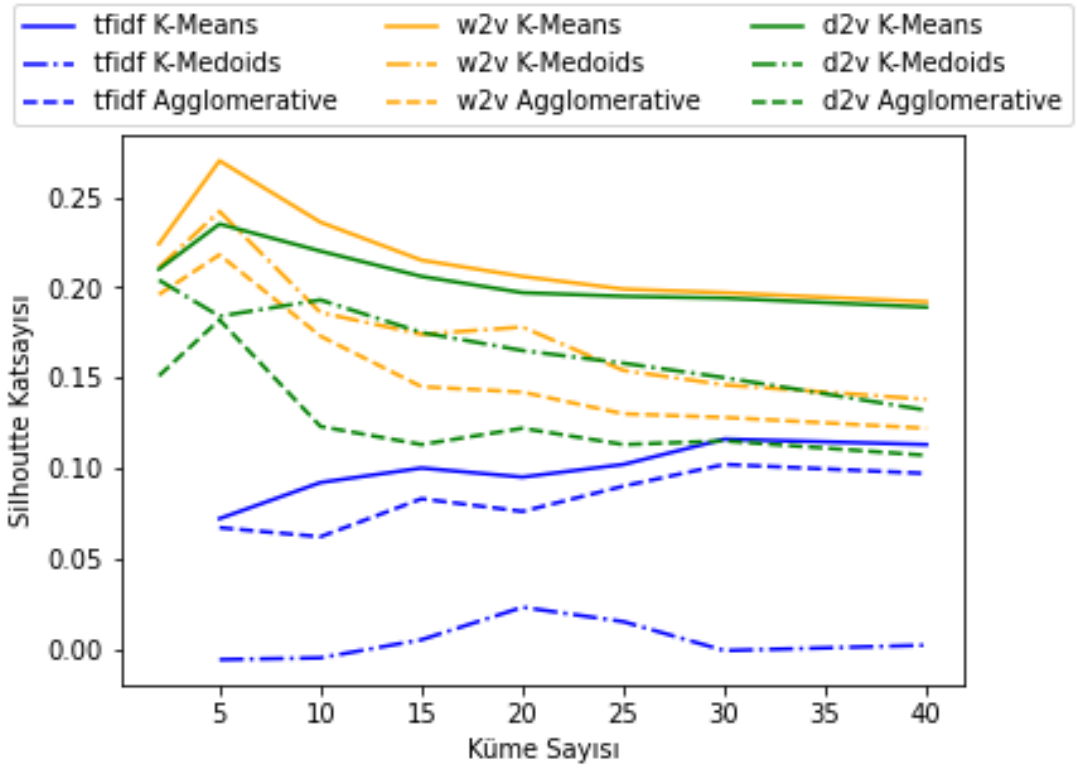
5.3. Senaryo Sonuçlarının Değerlendirilmesi

Çalışmada kullanılan yöntemler 3 farklı NLP yöntemi ile elde edilen kümeleme ve benzerlik çalışmalarının yakın sonuçlar verdiği görülmektedir. Değerlendirme metrikleri sonucunda elde edilen değerler aynı olmasa da her bir NLP yöntemi için en uygun kümeleme yöntemlerinın aynı olduğu sonucuna varılmıştır.

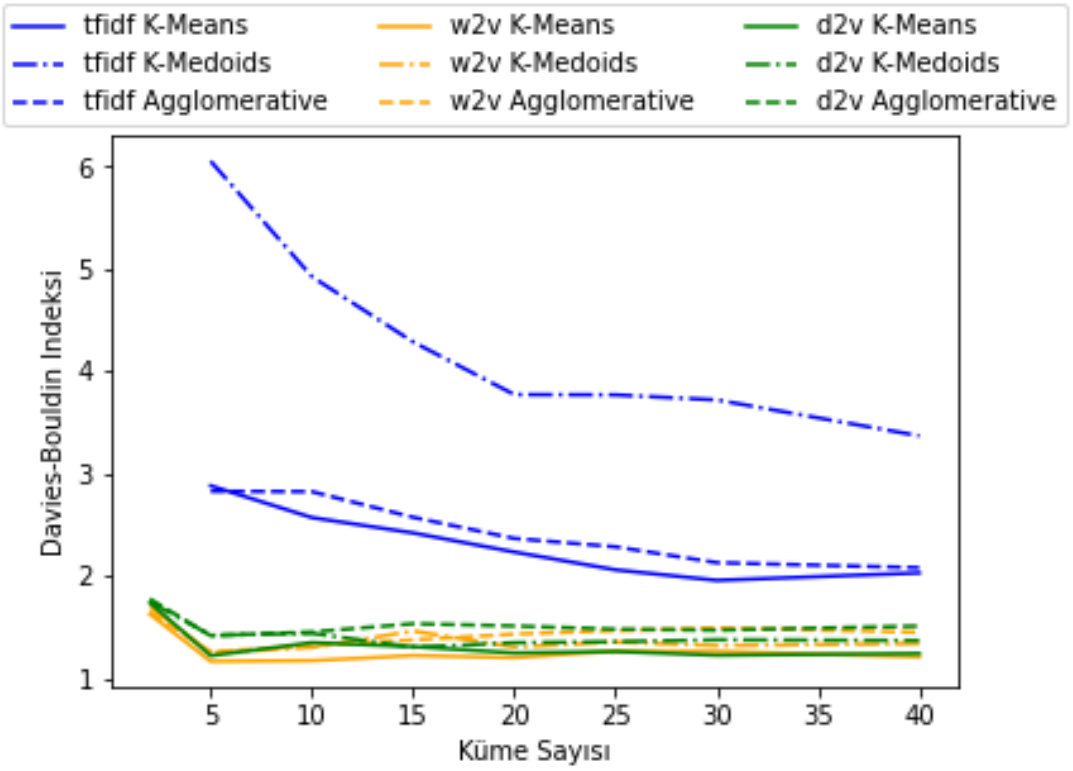
Frekans bazlı kelime temsil yöntemi ile elde edilen küme sayısının fazla olması boyut sayısının fazla olmasına ilişkindir. Danışmansız tahmin bazlı kelime temsil yöntemleri on milyonlarca kelimeyi eğitmek için tasarlanmış yöntemlerdir dolayısıyla çalışılan veri setinin küçük olması durumunda oluşturulan metin temsillerinin boyutları da küçüktür.

TF-IDF, Word2vec ve Doc2vec NLP yöntemleri ile oluşturulan metin temsilleri aracılığıyla yapılan hesaplamalar sonucunda, oluşturulan Şekil 5.17., Şekil 5.18., Şekil 5.19.'da yer alan grafikler incelendiğinde en iyi sonucu veren kümeleme algoritmasının ilgili her metrik için de K-Means olduğu görülmektedir.

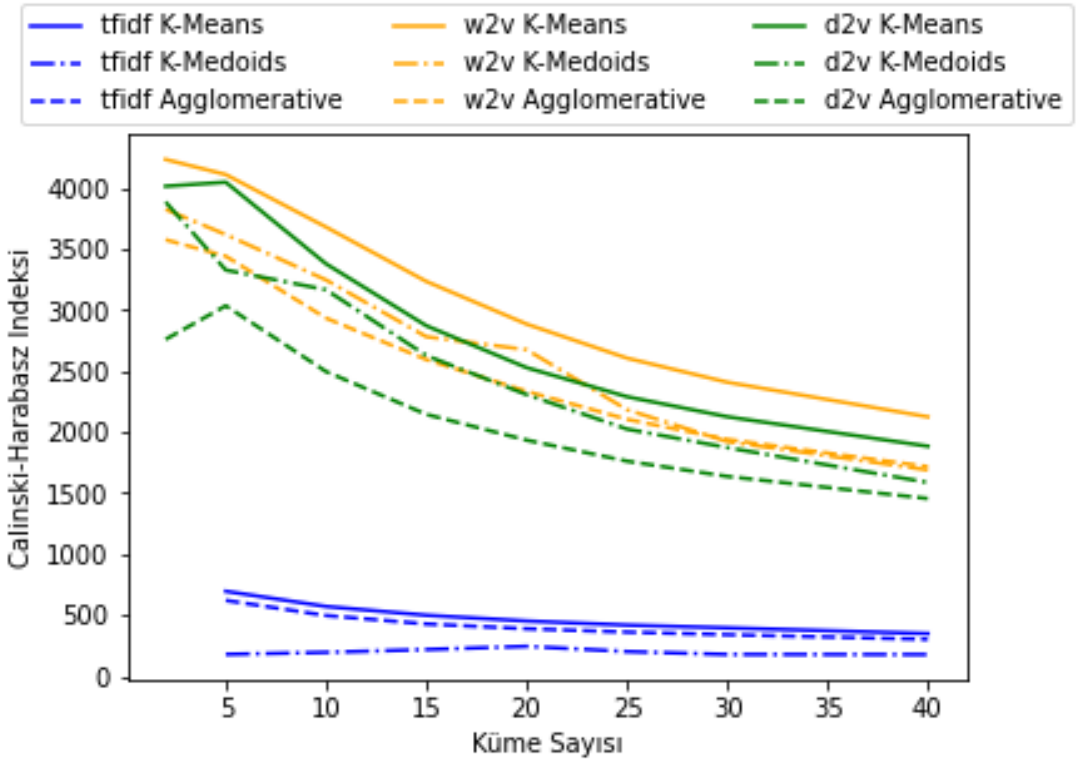
TF-IDF frekans bazlı kelime temsil yöntemi ve Word2vec, Doc2vec tahmin bazlı kelime temsil yöntemleri sonuçlarına bakıldığında ise frekans bazlı kelime temsil yöntemleri ile elde edilen sonuçlarda, boyut sayısının fazla olması sebebi ile ilgili metrikler değerlerinin, tahmin bazlı kelime temsil yöntemlerinden daha başarısız olduğu görülmektedir.



Şekil 5.17. Silhoutte katsayısı ile TF-IDF, Word2vec, Doc2vec senaryolarına ait kümeleme yöntemleri değerlendirme



Şekil 5.18. DB indeks ile TF-IDF, Word2vec, Doc2vec senaryolarına ait kümeleme yöntemleri değerlendirme



Şekil 5.19. CH indeks ile TF-IDF, Word2vec, Doc2vec senaryolarına ait kümeleme yöntemleri değerlendirme

Metinler arası benzerlik ölçümlerinde ise 3 farklı DDİ yöntemi sonucunda aynı metine ait önerilen metinlerin Tablo 5.6, Tablo 5.9 ve Tablo 5.12'de de görüldüğü üzere benzer olduğu sonucu elde edilmiştir. İlgili talep ve önerilen benzer talepler kontrol edildiğinde ise her yıl belirli periyotlarla istenen (tarihleri değiştirilmiş) içerikleri aynı olan talepler olduğu görülmüştür.

5.4. Gerçekleme

Bu çalışmada 3 farklı DDİ modeli ile veri seti eğitilerek, metinler arası benzerliğin bulunması için fonksiyonlar hazırlanmış ve 3 farklı kümeleme yöntemi ile de kümeleme çalışması yapılmıştır. Çalışmalar sonucunda elde edilen sonuçlar ve uzman kişiler değerlendirmeleri ile gerçekleme aşamasında kullanılacak olan NLP yönteminin Doc2vec, kümeleme yönteminin ise K-Means olmasına karar verilmiştir.

Gerçekleme esnasında kullanılacak olan veri seti ilgili kurumun Microsoft SQL Server veritabanında tutulmaktadır ve buradan Oracle ODI 11g sürümü ile veri ambarı sisteminin tutulduğu Oracle Exadata veritabanında yer alan XXX.ML_xxx_REQUEST isimli tabloya (Tablo 5.13.) aktarılmaktadır.

Tablo 5.13. XXX.ML_xxx_REQUEST tablo alan bilgileri

XXX.ML_xxx_REQUEST	
Kolon Adı	Açıklama
REQ_NO	Talep/metin numarası
REQ_HEADER	Talep/metin başlığı
EXPLANATION	Telep/metin açıklaması
BUSINESS_UNIT_NM	Talep/metini açan ekip adı
GMY_NM	Talep/metini açan genel müdürlük birim adı
DOMAIN_NM	Talep/metinin açıldığı ekip adı
CROSS_CHECK_DT	Talep/metnin onaylandığı tarih
REQ_TP_ID	Talep/metin tip numarası
REQ_TP_NM	Talep/metin tip adı
REQ_ST_ID	Talep/metin durum numarası
REQ_ST_NM	Talep/metin durum adı

Çalışmanın gerçekleme aşamasında ilgili kurumun veri ambarı ekibine ait taleplerin ve talep detaylarının yer aldığı havuz sistemin, kurulduğu günden çalışmanın yapıldığı güne kadar olan n adet talep metni, veritabanında yer alan XXX.ML_xxx_REQUEST isimli tablodan Spyder editörü aracılığıyla cx_Oracle kütüphanesi kullanılarak Python ortamına aktarılır.

Python ortamında, talep içeriğinde yer alan metinler, veri temizleme ve ön işlem aşamalarından geçirilerek veritabanında yer alan XXX.NORM_ML_xxx_REQUEST isimli tabloya (Tablo 5.14.) aktarılır.

Tablo 5.14. XXX.NORM_ML_xxx_REQUEST tablo alan bilgileri

XXX.NORM_ML_xxx_REQUEST	
Kolon Adı	Açıklama
REQ_NO	Talep/metin numarası
REQ_EXP	REQ_HEADER + EXPLANATION + DOMAIN_NM + BUSINESS_UNIT_NM + GMY_NM
REQ_TP_ID	Talep/metin tip numarası
REQ_TP_NM	Talep/metin tip adı
REQ_ST_ID	Talep/metin durum numarası
REQ_ST_NM	Talep/metin durum adı
DOMAIN_NM	Talep/metinin açıldığı ekip adı
CROSS_CHECK_DT	Talep/metnin onaylandığı tarih
NORMALIZE	REQ_EXP alanının normalize edilmiş hali

Normalize edilen metinler python ortamından veritabanına erişilerek XXX.NORM_ML_xxx_REQUEST isimli tablodan okunur ve Doc2vec doküman temsil yöntemi ile vektörel temsilleri elde edilir. Temsilleri elde edilen metinlerin arasından tipi ‘rapor’, talep durumu ‘açık’ olan metinler bir listeye beslenir ve bu listeden tek tek okunarak, her bir metin için en benzer 5 adet metin, metin numarası, en benzer metin numarası ve benzerlik oranı ile bir objeye yazılır. Akabinde bu obje veritabanında yer alan XXX.SIMILARITY_ML_xxx_REQUEST isimli tabloya (Tablo 5.15.) aktarılır.

Tablo 5.15. XXX.SIMILARITY_ML_xxx_REQUEST tablo alan bilgileri

XXX.SIMILARITY_ML_xxx_REQUEST	
Kolon Adı	Açıklama
REQ_NO	Talep/metin numarası
SIM_REQ_NO	En benzer talep/metin numarası
SIM_RATE	Benzerlik oranı

Temsilleri elde edilen metinlerin arasından tipi ‘rapor’, talep durumu ‘açık’ olan ve bir listeye beslenen metinler K-Means kümeleme yöntemi ile ilgili ekibin sahip olduğu çalışan sayısı kadar kümeye bölünerek XXX.CLUSTER_ML_xxx_REQUEST isimli tabloya (Tablo 5.16.) beslenir. Bu tablo, talep/metin numarası ve küme numarası alanlarına ve talep/metin sayısı kadar satıra sahiptir.

Tablo 5.16. XXX.CLUSTER_ML_xxx_REQUEST tablo alan bilgileri

XXX.CLUSTER_ML_xxx_REQUEST	
Kolon Adı	Açıklama
REQ_NO	Talep/metin numarası
REQ_CLUSTER	İlgili talebin ait olduğu küme numarası

İlgili tablo beslendiğinde her küme bir çalışana atanır, çalışan ise ilgili kümedeki rapor taleplerini XXX.SIMILARITY_ML_xxx_REQUEST tablosunda benzerleri var mı diye sorgular ve aksiyon alır.

Gerçekleme aşaması, ilk çalışmasında yukarıda aktarılan şekilde çalışır, sonrasında ise tüm akışın her seferinde en baştan çalıştırılması yerine sadece yeni gelen talepler için çalışacak şekilde dizayn edilir ve maliyet kaybı önlenir.

Çalışma ilgili kurumun periyodik iş çalıştırma uygulamasında günlük çalışacak şekilde düzenlenir. Kişiler ise sorumlu oldukları rapor taleplerini yaparken ilk önce XXX.SIMILARITY_ML_xxx_REQUEST tablosunu, ellerindeki talep/metin numarası ile sorgular akabinde gelen en benzer talep içeriklerine ve sorgularına bakarak sorumluluklarında olan işi en kısa sürede tamamlayarak teslim ederler.

XXX.CLUSTER_ML_xxx_REQUEST tablosu gerekli durumlarda beslenir, günlük beslenmez.

İlgili çalışma özel bir bankanın veri ambarı ekibi için yapılmıştır ve kurum kuralları gereği katı yetkilendirme kuralları mevcuttur. Dolayısıyla hazırlanan çalışmaya kurumun veri ambarı ekibi çalışanlarının yetkileri dahilinde ulaşabilecekleri şekilde entegre edilmesi gerekmiştir. Bu sebeple ilgili ekip çalışanlarının yetkilendirme konusunda sorun yaşamayacağı yer olan, bankanın veri ambarının tutulduğu veritabanına periyodik olarak çalışacak şekilde entegre edilmiştir.

Çalışmanın tamamlandığı tarihte ilgili veri ambarı ekibinin X konuları ile ilgilenen alt ekibinde 91 adet açık olan rapor talebi yer almaktadır. Bu çalışma öncesinde ilgili ekip lideri, 91 adet raporu göz ile tarayarak, ilgili ekip çalışanlarına atamaktaydı. İlgili ekip çalışanı ise atanan taleplerin her biri için manuel olarak tablolardan ve klasörden daha önce benzeri yapılmış mı diye gözle arayarak konu hakkında fikir edinebileceği bir sorgu elde etmeye çalışmaktaydı. Bu manuel süreç hem ekip lideri için hem de çalışan için ciddi zaman kaybına yol açmaktaydı. Çalışma tamamlandıktan sonra ise ekip lideri hazırlanan XXX.CLUSTER_ML_xxx_REQUEST tablosunda, kümeleri yer alan benzer talepleri, küme bazında ilgili ekip çalışanlarına atamakta ve çalışan ilgili kümedeki rapor taleplerinin numaralarını alarak XXX.SIMILARITY_ML_xxx_REQUEST tablosunda ilgili talebi aratarak daha önceden ilgili talebe benzer talep yapılmış ise o talebe ait detay bilgilieri XXX. ML_xxx_REQUEST tablosundan öğrenmekte ve talep sorgusuna ulaşabileceği tabloda veya klasörde benzer talep numarasını aratarak, talebin yapılması aşamasında yol gösterecek bilgilere ulaşarak talep tamamlama sürecini hızlandırmaktadır.

Çalışma ile gözlemlenen sonuçlardan bir diğeri ise iş birimleri tarafında hatalı açılan taleplerin tespit edilmesinin kolaylaşmasıdır. Örneğin; farkedilmeden aynı başlığa ve içeriğe ait olan birden fazla açılmaktadır. Bu çalışma ile XXX.SIMILARITY_ML_xxx_REQUEST tablosundan neredeyse %100 oranında benzer olan diğer talep numarası elde edilir ve XXX. ML_xxx_REQUEST tablosu sorgulanıp detay bilgiler elde edildiğinde ise aynı konuya sahip birden fazla talep

açıldığı farkedilir, taleplerden biri iptal edilir ve gereksiz yere aynı talep için maliyet kaybı yaşanmaz.

BÖLÜM 6. SONUÇ VE ÖNERİLER

Bu çalışma ile literatüre, kurumsal bir şirkette yer alan verilerin doğal dil işleme (NLP) ve makine öğrenmesi yöntemleri ile modellenerek, atılda bırakılmadan nasıl değerlendirilebileceğini gösterecek bir çalışma kazandırılması hedeflenmiştir.

Çalışma ile hedeflenen diğer konular ise çalışmanın gerek ilgili kurumun bilgi teknolojileri ekipleri gerekse de iş birimleri tarafından kullanılarak talep maliyetlerinin düşürülmesi, talep tekrarının engellenmesi ve hızlı aksiyon alınmasını sağlamaktır.

İlgili kurumun 14240 adet metin verisi 3 farklı NLP tekniği ile vektörel hale getirilerek her biri için 3 farklı kümeleme yöntemi ile kümelendirilmiştir. Her bir NLP yöntemi bazında en iyi kümeleme yöntemine karar vermek için küme sayısı belirleme ve değerlendirme yöntem & metrikleri kullanılmış olup, yorumlanarak ilgili NLP yöntemi için en uygun kümeleme yöntemi belirlenmiş ve en uygun küme sayısı için kümeleme yapılmıştır. Çalışmada kullanılan veri seti etiketli olmadığı için kümeleme başarısı ancak yorumlanabilir niteliktedir. Kullanıcı tarafından belirlenen X metnine en benzer n adet metin bulunurken aynı kümede olan metinlerin adreslenmesi sağlanmıştır. Buradaki amaç ise farklı kümelerde yer alan metinlerin kullanıcıyı yanıltmasının engellenmek istenmesidir.

Çalışmada kullanılan veri seti Türkçe ve kişilerin konuşma dili ile yazdığı metinlerdir. Dolayısıyla metinler dil bilgisi kurallarına uygunluk açısından zayıftır. Bu durum ise çalışmada kullanılan verinin yapısal hale getirilmesi aşamasında ciddi sorunlar ortaya çıkarmıştır ve kimi sorunlar düzeltilebilirken kimi sorunlar göz ardı edilmek zorunda kalınmıştır.

Çalışmadaki en büyük kısıtlardan biri, verinin etiketli olmaması ve değerlendirme yöntemlerinin ancak yorumlanabilir nitelikte olmasıdır. Etiketli verinin olmaması

sebebi ile çalışmanın her katmanında birden fazla yöntem çalışılmış olunup en uygun yöntem için yorumlar yapılmıştır.

Veri setinin beslendiği kaynak sistemde verilerin tarihsel olarak tutulmaması sebebi ile çalışmaya olumlu katkısı olacağı düşünülen öznitelikler kullanılamamıştır. İlgili havuz uygulamasında talepler, yöneticiler tarafından talebi karşılayacak kişiye atandığında 'Talep Cevaplayan Kişi' ve 'Talep Cevaplayan Birim' öznitelikleri ilgili kişi adı ve ilgili kişinin o tarihte bağlı olduğu birim olarak güncellenmektedir. İlgili kişi talebi o tarihte bağlı olduğu A biriminde karşıladıktan belirli bir zaman sonra B birimine geçtiğinde, ilgili talebi A biriminde iken karşılamış olmasına rağmen talep detaylarının tutulduğu tabloda B birimi olarak görünmektedir. Bu durum talep açıklaması ve cevaplayan birimin tutarsız olması sebebi ve modeli yanıltması sebebi ile kullanılamamıştır.

Doğal dil işleme çalışmalarındaki en önemli kısıtlardan biri de belirgin hesaplamaları olmayan tamamen çalışmada kullanılan veri seti özellikleri ve boyutuna bağlı belirlenmesi gereken modellere ait parametreleridir. Veri setine uygun model kurgulanırken parametre değerlerinin belirlenmesi için onlarca parametre çalışması yapılmıştır. Bu da zaman kaybına sebebiyet vermiştir. Bu tarz durumlar için çalışmaya ait en uygun parametreleri bulabilecek otomatize sistemler kurulmalıdır. Girdi olarak modeli alan bu sistemler onlarca, yüzlerce parametre denemesi yaparak her bir sonucu tutmakta ve en iyi sonucu veren parametreleri kullanıcıya geri beslemektedir.

Çalışmada kullanılan kütüphaneler ile ilgili soru veya sorun olduğunda sanal ortamda yer alan kirli bilgilere karşı dikkatli olunmalı ve gerektiğinde ilgili kişilerle iletişime geçilmelidir.

NLP uygulamalarında kullanılan veri seti ile model başarıları doğru orantılıdır. Veri seti ne kadar büyük ise modellerin öğrenmeleri de o kadar fazladır. Çalışmada 14240 adet metin verisi kullanılmıştır ve model başarıları yorumlandığında bu adetin düşük olduğu görülmektedir.

Çalışmalarda kullanılacak veri seti küçük ise ilgili veri seti içeriğine uygun önceden eğitilmiş modeller kullanılabilmektedir. Önceden eğitilmiş modellerde İngilizce için çok fazla model var iken Türkçe için yok denecek kadar azdır. Bu sebeple veri setinin küçük olmasına çözüm olarak görülen bu yöntem uygulanamamıştır.

NLP uygulamalarında veri manipülasyonu en önemli adımlardan biridir. Kullanılan model geliştirme ortamlarında veri manipülasyonu için çeşitli kütüphanler mevcuttur. Türkçe ile geliştirilen NLP uygulamalarındaki bir diğer kısıtta uygun kütüphane sayısının az olmasıdır. Türkçe'nin morfolojik yapısının çeşitliliği, sondan eklemeli bir dil olması hususu Türkçe için geliştirilen kütüphanelerin az olmasına sebebiyet vermiştir.

Kurumsal firmalarda, kurum içi oluşan veri setinin tarihsel, etiketli, parametrik ve formatlı şekilde oluşmasının sağlanması NLP uygulamaları için daha başarılı sonuçlar elde edilmesini sağlayacaktır. Bu hususta kurum içi kullanılan uygulamalarda geliştirmeler yapılabilir. Çalışmada kullanılan veri setinin tarihsel olarak tutulması önerilmektedir.

Farklı bir çalışma konusu olarak ise çalışmada kullanılan veri seti ile talep açılma ve talep kapanma tarihleri kullanılarak danışmanlı / denetimli (supervised) öğrenme yöntemleri ile yeni gelen bir talebin maliyet tahmini yapılabilir.

KAYNAKLAR

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. Information Processing and Management. 39 (1): 45–65. DOI: 10.1016/S0306-4573(02)00021-3.

Akın, M.D., & Akın, A.A. (2007) Türk Dilleri İçin Açık Kaynaklı Doğal Dil İşleme Kütüphanesi : Zemberek. Elektrik Mühendisliği, 431: 38-44. https://www.emo.org.tr/ekler/c7a625d5077d3ba_ek.pdf?dergi=483

Altıntaş, T. (2006). Veri madenciliği metotlarından olan kümeleme algoritmalarının uygulamalı etkinlik analizi. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Bölümü, Yüksek Lisans Tezi.

Anderberg, M.R. (1973). Cluster Analysis for applications. Academic Press, New York, 553–555.

Aslanyürek, M., & Mesut, A. (2021). Kümeleme Performansını Ölçmek için Yeni Bir Yöntem ve Metin Kümeleme için Değerlendirmesi. Avrupa Bilim ve Teknoloji Dergisi, 27: 53-65. DOI: 10.31590/ejosat.932938

Aydın, N., & Seven, A.N. (2015). İl Nüfus ve Vatandaşlık Müdürlüklerinin İş Yoğunluğuna Göre Hibrid Kümeleme İle Sınıflandırılması. Yönetim ve Ekonomi Araştırmaları Dergisi, 13 (2): 181-201.

Brock, G., Pihur, V., Datta, S., & Datta, S. (2008) clValid: An R Package for Cluster Validation Journal of Statistical Software 25(4). DOI: 10.18637/jss.v025.i04

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:135–146. https://doi.org/10.48550/arXiv.1607.04606

Calinski, T., & J. (1974). A Dendrite Method for Cluster Analysis. Communications in Statistics – Theory and Methods, 3(1): 1–27.

Charrad M., Ghazzali N., Boiteau V., & Niknafs A. (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. Journal of Statistical Software, 61(6): 1-36. DOI: 10.18637/jss.v061.i06

Çelik, Ö., & Koç, B.C. (2021). TF-IDF, Word2vec ve Fasttext Vektör Model Yöntemleri ile Türkçe Haber Metinlerinin Sınıflandırılması. DEUFMD, 23(67): 121-127.

Davies, D.L. & Bouldin, D.W. (1979). A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1(2): 224–227.

Demirkale, Ö. & Özarı, Ç. (2020). K-Ortalamalar Kümeleme Yönteme ile Temel Makroekonomik ve Finansal Göstergeler ile Değerlendirilmesi: Kırılgan Beşli Ülkelerinin Örneği. Finans Ekonomi ve Sosyal Araştırmalar Dergisi, 5 (1): 22-32. DOI: 10.29106/fesa.649176

Djellali, C. (2013). Enhancing text Clustering model based on Truncated Singular Value Decomposition, Fuzzy ART and Cross Validation. 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013), Niagara, Ontario. DOI: 10.1145/2492517.2500317

Hacıoğlu, H. K. (2016). Kümeleme analizinde kullanılan bazı benzerlik indekslerinin karşılaştırılması. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, İstatistik Ana Bilim Dalı, Yüksek Lisans Tezi.

Hansen, P. C. (1987). The truncatedSVD as a method for regularization. Bit, vol. 27, no. 4, pp. 534-553, 1987. https://doi.org/10.1007/BF01937276

Hartigan, J.A. (1975). Clustering Algorithms, Wiley New York, http://statsoft.com/textbook/esc.html., 01.01.2008.

Huang, A. (2008). Similarity measures for text document clustering. Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), New Zealand, 49-56.

Jain, V., & Kashyap, K.L. (2021). Optimal K-Means Clustering Algorithm For Weblog Mining. 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT). DOI: 10.1109/CSNT51715.2021.9509644

Jones, K. S. (1972). A Statistical Interpretation of Term Specificity and Its Application in Retrieval. Journal of Documentation. 28(1): 11–21. DOI: 10.1108/eb026526

Jurafsky, D., & Martin, H.J. (2000). Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition (PDF). Upper Saddle River, N.J.: Prentice Hall. DOI: https://doi.org/10.1515/zfsw.2002.21.1.134

Kassambara, A. (2017). Practical guide to cluster analysis in R: Unsupervised machine learning. 1. Baskı, STHDA. 138.

Kaufman, L., & Rousseeuw, P. J. (1987). Clustering by Means of Medoids, Statistical Data Analysis Based on The L1–Norm and Related Methods, edited by Y. Dodge, North-Holland, 405–416.

Kaufman, L., & Rousseeuw, P. J. (1990), Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley and Sons.

Ketchen, D. J., & Shook, C. L. (1996). The application of cluster analysis in strategic management research: an analysis and critique. Strategic management journal, 17(6): 441-458.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. 31th International Conference on Machine Learning, China. https://doi.org/10.48550/arXiv.1405.4053

Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information (PDF). IBM Journal of Research and Development, 1 (4): 309–317. DOI: 10.1147/rd.14.0309

Ma, H., Wang, X., Hou, J., & Lu, Y. (2017). Course Recommendation Based on Semantic Similarity Analysis. International Conference on Control Science and Systems Engineering, Beijing, China, 638- 641. DO: 10.1109/CCSSE.2017.8088011

MacQueen, J. B. (1967), MacQueen, Some Methods for Classification and Analysis of Multivariate Observations, Proc. Symp. Math. Statist. and Probability (5): 281–297.

Marriott, F. H. C. (1971). Practical Problems in a Method of Cluster Analysis. Biometrics. 27(3): 501-514.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficientestimation of Word Representations in Vectorspace. https://doi.org/10.48550/arXiv.1301.3781

Peker, N. & Kubat, C. (2021). Boyut Azaltmanın Bulanık C-Ortalama Kümeleme Teknikleri Üzerindeki Etkisi. Veri Bilimi, 4 (1): 1-7. Retrieved from https://dergipark.org.tr/tr/pub/veri/issue/59505/769371

Rousseeuw, P. (1987). Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of Computational and Applied Mathematics, 20, 53–65.

Silahtaroğlu, G. (2016). Veri madenciliği: Kavram ve algoritmaları. Papatya.

Subba, B., & Gupta, P. (2020). A tfidfvectorizer and singular value decomposition based host intrusion detection system framework for detecting anomalous system processes. Computers & Security. DOI: 10.1016/j.cose.2020.102084

Taşçı, A.E., & Onan, A. (2016). K En Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi. Akademik Bilişim, Aydın, Türkiye, 1-8.

Theodoridis, S., & Koutroubas, K. (2008). Pattern Recognition. 4th edition. Academic Press.

Tunalı, T., & Bilgin T.T. (2012). Türkçe Metinlerin Kümelenmesinde Farklı Kök Bulma Yöntemlerinin Etkisinin Araştırılması. ELECO 2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu, Bursa.

Wang, G., & Kwok, S.W.H., (2021). Using K-Means Clustering Method with Doc2vec to Understand the Twitter Users' Opinions on COVID-19 Vaccination. 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI). DOI: 10.1109/BHI50953.2021.9508578

Qi, Z. (2020). The Text Classification of Theft Crime Based on TF-IDF and XGBoost Model. 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). DOI: 10.1109/ICAICA50127.2020.9182555

ÖZGEÇMİŞ

Adı Soyadı **:** Seda Aydin Tuzcuay

ÖĞRENİM DURUMU

Derece	Eğitim Birimi	Mezuniyet Yılı
Yüksek Lisans	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü / Bilişim Sistemleri Mühendisliği	2022
Lisans	Sakarya Üniversitesi / Bilgisayar ve Bilişim Bilimleri Fakültesi / Bilişim Sistemleri Mühendisliği	2017
Lise	Kemal Hasoğlu Lisesi	2012

İŞ DENEYİMİ

Yıl	Yer	Görev
2018-Halen	IBTECH Uluslararası Bilişim ve İletişim Teknolojileri A.Ş.	Senior Data Designer
2017-2018	Türk Ekonomi Bankası A.Ş	Junior Datawarehouse Developer

YABANCI DİL

İngilizce

ESERLER (makale, bildiri, proje vb.)

1. Bankacılık Sistemlerinde Veri Ambarı Uygulamaları Gelişim Sürecinin İncelenmesi. IMASCONGREES Uluslararası Marmara Fen ve Sosyal Bilimler

Kongresi, 23 Kasım 2018. Sayfa: 1068-1074 http://www.imascon.com/dosyalar/imascon2018/imascon2018_tam_metin.pdf