KARADENİZ TEKNİK ÜNİVERSİTESİ

OF TEKNOLOJİ FAKÜLTESİ

YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM GELİŞTİRİLERİN TARTIŞTIKLARI KONULARA YÖNELİK KULLANICI SORU VE CEVAPLARININ OLASILIKSAL KONU MODELLEME YÖNTEMİ İLE BELİRLENMESİ

BİTİRME ÇALIŞMASI

Olcay ÇİFTÇİ

2019 -2020 BAHAR DÖNEMİ

KARADENİZ TEKNİK ÜNİVERSİTESİ

OF TEKNOLOJİ FAKÜLTESİ

YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

YAZILIM GELİŞTİRİLERİN TARTIŞTIKLARI KONULARA YÖNELİK KULLANICI SORU VE CEVAPLARININ OLASILIKSAL KONU MODELLEME YÖNTEMİ İLE BELİRLENMESİ

BİTİRME ÇALIŞMASI

Olcay ÇİFTÇİ

Bu projenin teslim edilmesi ve sunulması tarafımca uygundur.

Danışman : Doç. Dr. ÖZCAN ÖZYURT .....

2019 -2020 BAHAR DÖNEMİ

IEEE Etik Kuralları
IEEE Code of Ethics

Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriyi kabul etmek ve eleştiriyi yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayırımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

ÖNSÖZ

İçinde bulunduğumuz dönemde yaşanan teknolojik gelişmeler, genel anlamda geçmişe göre hız kazanıp özellikle internet kullanımını ve herkesin internete yönelmesini kolaylaştırmıştır. Artık yazılım geliştiricilerin karşılaştıkları sorunları diğer yazılım geliştiriciler ile paylaşarak, yardımlaşarak, fikir alışverişi yaparak çözümleyebilecekleri bir çok platform türemiştir. Hem bu gelişmeden dolayı hemde artık bilgiye erişimin kolay olmasından dolayı her alanda olduğu gibi yazılım geliştiricilerde yeni teknolojiler araştırıp, öğrenip kendilerini geliştirmeye, vizyonlarını genişletmeye çalışmaktadırlar. Artık bahsi geçen bu platformlarda bir çok farklı konu tartışılır, sorulur hale gelmiştir. Yazılım geliştiriciler kendilerini geliştirirken bir yandan kullandıkları teknolojileri de çok hızlı şekilde geliştirir duruma gelmiş ve populer olarak kullanılan teknoloji aydan aya değişim gösterebilir hale gelmiştir.

Bu bağlamda geliştirilen tez çalışmamda öncelikle tez çalışmamın planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, tez konusunu seçerken isteklerimi göz önünde bulundurup bana yardımcı olan, yol gösteren, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı şekillendiren sayın hocam ve aynı zamanda tez danışmanım Doç. Dr. Özcan ÖZYURT' a ve Öğr. Gör. Dr. Fatih GÜRCAN hocalarıma teşekkür ve saygılarımı sunarım.

Bütün hayatım ve eğitimim boyunca sürekli yanımda bulunan ve beni destekleyen, beni yetiştirip bu günlere getiren sevgili annem Nejla ÇİFTÇİ, babam Oktay ÇİFTÇİ ve kardeşim Alican ÇİFTÇİ' ye desteklerinden dolayı saygı ve sevgilerimi sunarım.

Olcay ÇİFTÇİ

İzmir, 2020

İÇİNDEKİLER

Sayfa No

Lisans Tezi

ÖZET

YAZILIM GELİŞTİRİLERİN TARTIŞTIKLARI KONULARA YÖNELİK KULLANICI SORU VE CEVAPLARININ OLASILIKSAL KONU MODELLEME YÖNTEMİ İLE BELİRLENMESİ

Olcay ÇİFTÇİ

Karadeniz Teknik üniversitesi

Of teknoloji Fakültesi

Yazılım Mühendisliği

Danışman: Doç. Dr. Özcan ÖZYURT

2020, 48 sayfa

İçinde bulunduğumuz dönemde yaşanan teknolojik gelişmeler, genel anlamda geçmişe göre hız kazanıp özellikle internet kullanımını ve herkesin internete yönelmesini kolaylaştırmıştır. Artık yazılım geliştiricilerin karşılaştıkları sorunları diğer yazılım geliştiriciler ile paylaşarak, yardımlaşarak, fikir alışverişi yaparak çözümleyebilecekleri bir çok platform türemiştir. Yazılım geliştiriciler bilgiye ulaşımın kolay olduğu bu dönemde kendilerini geliştirirken bir yandan kullandıkları teknolojileri de çok hızlı şekilde geliştirir duruma gelmiş ve populer olarak kullanılan teknoloji aydan aya değişim gösterebilir hale gelmiştir. Son yıllarda metin madenciliği uygulamalarında büyük önem kazanan konu modelleme yöntemleri ise bu alanda tercih edilmeye başlanmıştır. Büyük boyutlu dokümanlardan denetimsiz bir şekilde gizli yapıyı keşfeden konu modelleme güçlü bir yöntem olarak karşımıza çıkmaktadır. Bu çalışmada stack overflow soru ve cevaplarından aylık olarak tartışılan konuların eğilimini çıkarmada en popüler konu modelleme yöntemlerinden biri olan Gizli Dirichlet Tahsisi (GDT) (Latent Dirichlet Allocation (LDA)) kullanılacaktır.

Anahtar Kelimeler: Olasılıksal konu modelleme, Gizli Dirichlet tahsisi, Eğilim analizi, Metinsel veri madenciliği, Bilgi çıkarımı.

License thesis

SUMMARY

Determining Issues Discussed by Software Developers on the User Questions and Answers Using Probabilistic Topic Modeling Process

Olcay ÇİFTÇİ

Karadeniz Technical University

OF Technology Faculty

Software Engineering

Supervisor: Assoc Prof. Dr. Özcan ÖZYURT

2020, 48 pages

Technological developments in the current period have gained speed compared to the past in general and facilitated the use of the internet and everyone's heading to the internet. Now, many platforms have been developed where software developers can solve the problems they face by sharing, helping, exchanging ideas with other software developers. While software developers have been developing themselves in a time when it is easy to access information, on the other hand, they have developed the technologies they use very quickly and the technology used as a popular has changed from month to month. Topic modeling methods, which have gained great importance in text mining applications in recent years, have been preferred in this field. Topic modeling, which uncovered the hidden structure from oversized documents, is a powerful and unsupervisedly method. In this study, Latent Dirichlet Allocation (LDA), one of the most popular subject modeling methods, will be used to draw the trend of the issues discussed monthly from stack overflow questions and answers.

Key Words: Probabilistic topic modeling, Latent Dirichlet allocation, Trend analysis, Textual data mining, Knowledge extraction

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

SEMBOLLER DİZİNİ

GDT	: Gizli Dirichlet Tahsisi
LDA	: Latent Dirichlet Allocation
BT	: Bilişim Teknolojileri
CRISP-DM	: Cross Industry Standard Process Model for Data Mining
DDİ	: Doğal Dil İşleme
BoW	: Bag of Words- Kelime torbası
TF	: Terim Sıklığı(Term Frequency)
IDF	: Ters Doküman Sıklığı (Inverse Document Frequency) (IDF)
TF-IDF	: Terim sıklığı- Ters belge sıklığı (TF-IDF)
GAA	: Gizli anlamsal analiz
TDA	: Tekil değer ayrışması
GAI	: Gizli anlamsal indeksleme
OGM	: Olasılıksal grafik modelleri
DTM	: Doküman-terim matrisine

1.GENEL BİLGİLER

1.1. Giriş

Son günlerde bilgi işletmelerin ve bireylerin gelişiminde artık stratejik bir yer almaya başlamıştır. Bu kapsamda bilgi kapsamlı araştırmalar ve paralel olarak yayınlar giderek artmaktadır. Hem akademik personel hemde iş dünyasındaki insanlar yeni bilgi oluşturma, oluşmuş bilgiyi elde etme ve elde edilen bu bilgiyi bütün iş süreçlerinde kullanmak ve paylaşmak için nelerin yapılması gerektiği konusunda çeşitli arayışlar içerisindedir. [1]. Bu bakımda Bilişim Teknolojileri (BT) için ise bilginin paylaşımı ve yayılımı konusunda internet, kitaplardan daha çok ön plana çıkmaktadır. Günümüzde öğretim birikmiş bilgi ve becerileri aktarmaktansa bilgiye erişmek ve erişilen bilgiyi kullanabilme becerilei kazanma anlayışını ön plana çıkartmaktadır[2]. Yani elimizde olan bilgiye nazaran dünya üzerinde güncel olarak kabul edilen veya henüz araştırma aşamasında olan yenilikçi fikirlerin araştırılması ve bu yeni bilginin kullanılması konusunda yeni beceriler kazanılması ön plana çıkmaktadır. BT alanındaki bu gelişmeler sayesinde günümüzde yazılım teknolojileride büyük bir gelişme yaşamıştır. yazılım endüstrisi BT alanındaki yenilikçiliğin önünü açan temel unsurlardandır. Yazılıma dayalı teknolojiler, günümüzde modern ürün ve hizmetlerin çoğunda yer alan en işlevsel bileşenlerdir[3,4].

Yazılım odaklı endüstrilerdeki gelişen talep ve gereksinimlerin karşılanabilmesi, teknik zorluklarının aşılabilmesi ve yazılım odaklı ürün ve hizmetlerin günümüzün ihtiyaçlarına cevap verebilecek kalite ve işlevsellikte olabilmesi için yazılım geliştirme uzmanlık alanlarının daha etkin ve daha dinamik bir yapıya sahip olması gerekmektedir. Yazılım geliştirme uzmanlık alanlarında bu dinamik yapının sağlanabilmesi için de güncel piyasa taleplerine duyarlı yeni nesil yazılım geliştirme mimarileri, teknikleri, araç ve yöntem bilimlerine gereksinim duyulmaktadır [3,5].Bu bağlamda günümüz yazılım geliştirme ortamları internete erişimin basitleşmesi sonucunda çok kolay şekilde öğrenilebilir ve geliştirilebilir duruma gelmiştir ve gün geçtikçe yapılan geliştirmeler sonucu birbirine rakip bir çok yazılım geliştirme platformları ve yazılım geliştirme dilleri türemiştir. Teknolojinin gelişmesi ve çoğu yazılım dillerinin açık kaynak paylaşımlı olmalarından dolayı ise yazılım teknolojilerinde takip edilemeyecek gelişmeler gerçekleşmekte ve aynı işi yapan rekabet içerisinde olan yazılım dilleri ve yeni geliştirilen platformlar arasında popüler olarak kullanım çoğunluğu bakımından sürekli değişimler olmaktadır.

Bu gelişmeler ışığında sürekli değişen taleplere ayak uyduran yazılım endüstrisi hele ki internet ve bilişim çağındayken yadsınamaz bir olgudur. Bu noktada hem öğrencilerin hemde kariyerine bu alanda devam eden yazılım geliştirme uzmanlarının geleceği açısından yazılım teknolojilerindeki değişen eğilimlerin belirlenmesi, özellikle nitelikli işgücünün yetişmesi anlamında ciddi katkılar sağlayabilir [6,7]. Ne yazık ki günümüzde çoğu eğitim kurumunda ve üniversitelerde yazılım alanlarında eğitim gören öğrenciler için gerekli olan teknolojilerin anlatılması konusunda piyasada kullanılan güncel teknolojiler ile örgün eğitimde öğretilen bilgi ve beceriler arasında kopukluklar ve ciddi uyuşmazlıklar olmaktadır. Bu durum ciddi işgücü kaybına ve ciddi ekonomik durumların çıkmasına neden olabilir. Bu nedenle yazılım yazılım uzmanlarının teknik bilgi ve becerilerinin piyasadaki talepler doğrultusunda her zaman güncel olması gerekmektedir.

Sürekli olarak değişimde ve ilerlemede olan yazılım sektöründe haliyle karmaşıklık ile doğru orantılı olarak soru ve sorunlarla karşılaşma artmaktadır. Günümüzde yazılım geliştiriciler karşılaştıkları sorunların çözümünü ilk olarak çevrelerinde deneyimli yani karşılaşılan sorun ile daha önceden karşılaşması muhtemel yazılım geliştiricilere sormakta aramaktadırlar. Bu gibi imkanları olmayan veya sorunun yöneltildiği yazılımcıdan olumlu dönüt alınamayan durumlarda internet sayesinde bilgiye ulaşımın kolaylaşmasından dolayı yazılım geliştiriciler sorularının yanıtını internet üzerinde araştırır. Karşılaşılan bu gibi problemler internet aracılığı ile dünya üzerindeki tüm yazılım geliştiricilerin ulaşabileceği platformlara olan ihtiyacı doğurmuştur. Bu durum günümüzde kolaylıktan çok bir ihtiyaç haline gelmiş durumdadır ve birçok firma bünyelerine yazılım geliştirici katmak istediklerinde ön görüşme sırasında yazılım geliştiricilerden güncel teknolojileri, sorunları ne kadar takip ettiklerini ve ne kadar gelişmeye sorun çözmeye yatkın olduklarını gözlemlemek için var ise yazılım geliştiricilerin kullandığı online platformlardaki kullanıcı hesaplarının adını istemektedir.Yazılım sektöründe rakip firmalardan geri kalmamak için birçok firma bünyesinde yeni teknolojileri bilen ve kendi sistemlerini yeni teknolojilere entegre edebilecek yazılım geliştiricilerden oluşan takımlar bulundurur.

Günümüzde bu gibi platformların çoğunluğu ve yoğun kullanımından dolayı, bu gibi platformlarda paylaşılan soru ve bu soruların cevapları birikerek, paylaşılan ve depolanan bilgilerin miktarının ve çeşitliliğin artış göstermesine katkıda bulunmuştur. Bu biriken bilgiler yazılım endüstrisindeki işgücü piyasasında ortaya çıkan talep ve eğilimlerin belirlenmesinde önemli bir bilgi kaynağı olarak görülebilir.[8]

Bu amaç ve kapsam doğrultusunda yazılım geliştiricilerin interaktif soru ve cevap paylaşımı yaptığı platformlar üzerindeki paylaşımları istatistiksel modellemeye dayalı doğal dil işleme yöntemleri ile çözülebilir ve elde edilen konular piyasanın aylık olarak hangi teknolojilere ihtiyaç duyduğunu incelememize olanak sağlayabilir.

1.2. Tezin Kapsamı

Tezin birinci bölümünde yapay zeka nedir, veri madenciliği nedir ve hangi aşamalardan oluşur, doğal dil işleme hakkında genel bilgi, metin ön işleme, öznitelikler ve metinlerin vektörel temsili kavramlarının yanı sıra terim ağırlıklandırma ve bu çalışmada kullanılan yöntem GDT hakkında tanım ve bilgilere yer verilmiştir.

İkinci bölümde, yapılan çalışmanın genel mimarisi, analizde kullanılan veri setinin oluşturulması, veri seti üzerinde yapılan ön işleme adımları, kelime uzayının belirlenmesi ve doküman-terim matrisinin oluşturulması ardından son olarak GDT tabanlı konu modelleme sürecinin veri setine uygulanması aşamaları anlatılmıştır.

Üçüncü bölümde, gerçekleştirilen analiz sonucunda elde edilen bulgulara yer verilmiştir.

1.3. Yapay Zeka

Yapay zeka, makine olarak tabir edilen bir bilgisayarın veya bilgisayar tarafından kontrol edilen bir mekanik oluşumun(robot vb.) doğada hali hazırda bulunan canlıların hayatlarını sürdürme şekillerine benzer şekilde kurgulanarak çeşitli faaliyetleri icra etme kabiliyetidir[9]. Kısaca insanlara özgü olan düşünme ve karar verme işlevini taklit etmeye çalışan bilgisayar işlemlerine denebilir. Başka bir tabir ile bilgisayarın zeki bir varlık gibi düşünmesi gibi algılansada bu gibi tanımlar günümüzde hızlı bir şekilde değişmekte, gelecekte insan gibi gelişmiş bir canlının zekasından bağımsız şekilde kendini geliştirebilen bir yapay zeka kavramına doğru evrilmektedir.

Yapay zeka terimi ilk olarak 1950'lerin ortalarında kullanılmaya başlanmıştır. Yapay zeka, otomatik şekilde işlenemeyecek zeka olgularını modelleyerek makineleştirmek için çalışmaktır. [10].

Yapay zeka konusunu yalnız bilgisayar bilimiyle kısıtlamak yerine psikoloji, felsefe, eğitim gibi ve daha birçok faklı alanıda kapsayacak şekilde daha geniş bir çerçevede ele almak gerekmektedir[10]. Bu bağlamdan da anlaşılacağı üzre gelişen teknoloji ile birlikte

gelişen yazılım teknolojisi artık hayatın bir çok alanında kullanılmakta ve en basitinden evlerimizin mutfaklarına kadar girmektedir. Öyle ki gelişen yazılım sistemleri çok uzak olmayan gelecekte yapay zekaya sahip bir çok ev eşyasının olabileceğini işaret eder.

1.4. Yapay Zeka Uygulama Konuları ve Kullanım Alanları

Yapay zekanın genel amacı, insana özgü niteliklerin bilgisayar tarafından modellenmeye çalışılmasıdır. Bunun için, yapay zekanın bir çok alanı etkilediği ya da başka bir ifadeyle bir çok alanda kullanılabileceği açıktır[11,12]. Bu bağlamda yapay zeka uygulama alanları şu gruplar altında toplanabilir.

- Ses tanıma
- Görüntü işleme
- Doğal dil işleme
- Muhakeme
- Robotik
- Sinirsel ağlar
- Bulanık mantık
- Makine öğrenmesi
- Derin öğrenme
- Eğlence
- Tıp
- Siber güvenlik
- Yaşamsal görevler (yaşlı bakımında kullanılan yapay zeka teknolojileri)
- Ulaşım
- Akıllı ev sistemleri
- E-ticaret sistemleri
- Bankacılık ve finansal hizmetler
- Eğitim
- Dil çeviri sistemleri
- Müzik ve seyir öneri sistemleri

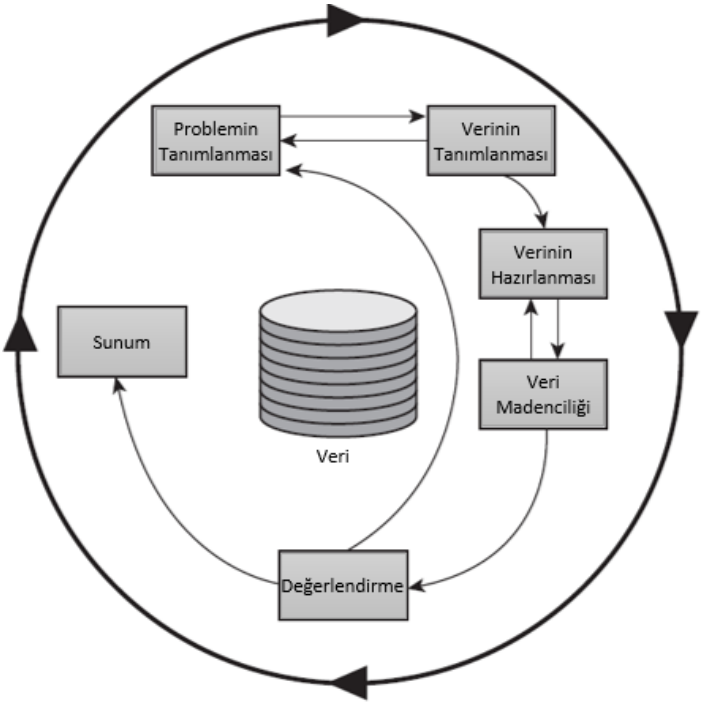
1.5. Veri Madenciliği

Veri madenciliği, mevcut veride gizli olarak yatan, veriye bakıldığı zaman anlaşılmayan ve daha önceden bilinmeyen ancak öğrenildiği zaman kullanışlı olacak bilgilerin çıkarılmasıdır. Kümeleme, verilerin özetlenmesi, yapılan değişikliklerin analizi, sapmaların tespiti gibi belirli sayıda teknik yaklaşım içermektedir. [13].

Aynı zamanda, veri madenciliği, verilerin içinde gizli olarak bulunan desen, ilişki, değişim, düzensizlik, kural, örüntü gibi yapıların yarı otomatik bir şekilde keşfedilmesidir.Yani büyük bir veri yığını içerisinden belli süreçler yardımı ile bilginin keşfidir. Bahsedilen bu süreçler istatistik ve veri tabanlı sistemlerin birbiri ile kullanılması ile oluşur.

1.6. Veri Madenciliği Aşamaları

Crisp-dm(Cross Industry Standard Process Model for Data Mining) metodolojisi, çok çeşitli iş uygulamaları ve endüstrilerde veri madenciliğinin kullanımını artırmayı ve doğru sonuçları elde etmeyi amaçlayan, veri madenciliği projelerinde başarılı bir sonuç elde etmek için altı adımdan oluşan bir süreçtir.



Şekil 1. Crisp-dm metadolojisine göre veri madenciliği aşamaları

1.6.1. Problemin Tanımlanması: Veri madenciliği projelerindeki en önemli aşama bu aşamadır, çünkü projenin amacı bu adımda tanımlanır. Bu adımdaki zorluk, proje

paydaşlarının birbiriyle ilişkili konulardaki bilgilerinin farklı olması, proje ile ilgili önyargıları ve yöntemleri olmasıdır. Tüm paydaşlar aynı konuyu aynı şekilde göremezler bu nedenle farklı yorumlar temel meseleyi anlamak ile ilgili paydaşların zorluk çekmelerine neden olabilir. Bu sebeple projenin ana hedefinin ne olduğunu tüm proje paydaşlarıyla görüşüp karar verilmelidir.

1.6.2. Verinin Tanımlanması: Eğer ki business understanding adımında herşey yolunda gider ise tespit edilen hedefe uygun verilerin toplanması işlemi bu adımda yapılır. Projenin ne istediğini ve ihtiyaçlarının anlaşılması, hangi verilerin toplanacağını, hangi kaynaklardan ve hangi yöntemlerle toplanacağını belirleyecektir. Dolayısı ile bu adımda hedefe yönelik tüm verinin toplanmasının tamamlanması gerekecektir.

1.6.3. Verinin hazırlanması, ayıklanması ve önişleme: Veriler toplandıktan sonra, daha fazla veriye ihtiyaç olmadığı belirlenmedikçe kullanılabilir bir alt kümeye dönüştürülmelidir. Dolayısı ile bir veri kümesi seçildikten sonra, şüpheli, eksik veya belirsiz durumlar kontrol edilmelidir. Kontrol sürecinde bir problem tespit edilirse bir sonraki adıma geçmeden önce bu problem giderilmelidir.

1.6.4. Veri Bütünleştirme: Bir çok farklı kaynaktan elde edilen verilerin veri setleri içerisinde analiz aşamasında oluşabilecek tutarsızlığı ortadan kaldırmak adına verilerin türlerinin tek türe dönüştürülmesi aşamasıdır. Farklı türdeki bir çok veri tek bir yapı şekline modellenerek tekdüzelik sağlanır ve veri homojen bir yapı alır.

1.6.5. Veri İndirgeme: Çok büyük veri setlerinde gerçekleştiren analizlerde verinin tamamını analiz etmek hem zor hem maliyetli olabilir. Bu yüzden yapılacak analize etkisi hiç olmayacak veya en az olan bağımsız değişkenler tespit edilip verinin miktarı veya değişken sayısı bu aşamada azaltılır.

1.6.6. Veri Dönüştürme: Kullanılacak analiz modeline uygun olacak şekilde elimizdeki verinin içeriğinin korunarak sadece biçiminin uygun formata dönüştürülmesi işlemidir.

1.6.7. Veri Madenciliği Aşaması: Önceki aşamalar sonucu uygulamaya hazır hale getirilen veri problemimize uygun olan veri madenciliği algoritması ile analiz işlemine tabi tutulur. Dikkat edilmesi gerekilen kısım problemimize en uygun veri madenciliği yönteminin seçilmesidir. Bunun için benzer yöntemler denenebilir ve en uygun yöntem bulunabilir.

1.6.8. Örüntü Değerlendirme: Bu aşamada, uygulanan veri madenciliğinden sonra oluşan model incelenen alanca uzmanlar tarafından incelenir. Eğer sonuçlar tatmin edici değilse modelleme aşamasına geri dönülür ve ideal değerler elde edilene kadar bu döngü devam eder. Eğer her aşama güzelce uygulanırsa elde edilen örüntüler değerlendirilir ve nerede ne şekilde kullanılacaklarına karar verilir.

CRISP-DM oldukça esnek ve döngüsel bir modeldir. Modelin bu özelliği her adımda bir önceki adıma tekrar dönmeyi ve değişiklik yapmayı gerekli kılabilir.

1.7. Veri Madenciliği Modelleri

Veri madenciliği modelleri genel olarak Tahmin Edici ve Tanımlayıcı modeller olacak şekilde iki başlıkta incelenebilir.

1.7.1. Tahmin Edici Modeller(Predictive Models): Bir takım olay sonunda sonuçları belli olan ve kaydedilen veriler kullanılarak bir model oluşturulup, elimizdeki sonuçları bilinmeyen veri grupları için uygun etiketler doğrultusunda sonuçların tahmin edilmesidir.

- Sınıflandırma
- Regresyon
- Zaman Serisi Analizi

1.7.2. Tanımlayıcı Modeller (Descriptive Models): Karar verme işlemini gerçekleştirmek için kullanılması mümkün olan verilerdeki örüntülerin belirlenip tanımlanmasını sağlamakta olan modellerdir.

- Kümeleme Yöntemi
- Birliktelik Kuralı

1.7.3. Birliktelik Kuralları (Association Rules): Büyük veritabanlarında birbirleri ile ilişkili değişkenlerin ve bu değişkenlerin aralarında bulunan örüntünün ve bu örüntünün büyüklüğünün tespiti için kullanılan yöntemdir. Apriori, Carma, Eclat, Sequence, GRI… birliktelik yönteminde kullanılan bazı algoritmalardır.

Örneğin Alışveriş alışkanlıkları incelenirken markette ekmek alanlar yüzde kaç oranında süt veya ekmek, süt alanlar yüzde kaç oranında peynir alıyor gibi ilişkiler tespit edilebilir.

Ekmek → | %70 (satılan ürünlerde %70 oranında ekmekte alınıyor.) Ekmek → Süt | %50 (ekmek alanlar %50 oranında sütte alıyor.) Ekmek,Süt → Peynir | %40 (ekmek ve süt alanların %40 oranında peynir alıyor)[14]

1.7.4. Sınıflandırma ve Tahmin (Classification and Prediction): Gelecekteki veri üzerinde eğilimleri gözlemlemek ve açıklamak için bir nesnenin niteliklerini inceleme ve bu nesneyi önceden yapılan eğitimler sonucu oluşturulan sınıflardan birine atama işlemidir. Tree, Random Forest, Naive Bayes, KNN… sınıflandırma yöntemi için kullanılan algoritmalardan bazılarıdır.

Örneğin kredi başvurusu yapacak bir müşteriye kredi verilebilirliği, Geçmiş bilgilerden hastalık teşhisi, Ses tanıma, kullanıcı davranışlarını belirleme.. birer sınıflandırma örnekleridir. [14]

1.7.5. Tahminleme: veri seri içerisinde eksik, bilinmeyen veya hatalı olan sayısal verilerin tahmin edilmesi işlemidir.

1.7.6. Kümeleme Analizi(Cluster Analysis): Dağınık halde duran verileri özelliklerine ve benzerliklerine göre birleştirip işlenebilir halde gruplara ayırma işlemidir. Sınıflandırma işlemine benzer ancak aradaki fark kümeleme işlemi yapılırken sınıflandırma işlemi gibi önceden kümeler belirlenmez. K-Means, K-Metoids… algoritmaları bazı kümeleme algoritmalarıdır.

Örneğin marketlerde farklı müşteri gruplarının keşfedilmesi ve bu grupların alışveriş örüntülerinin ortaya konması, biyolojide bitki ve hayvan sınıflandırmları ve işlevlerine göre genlerin sınıflandırılması, şehir planlamasında evlerin tiplerine,değerlerine ve coğrafi konumuna göre gruplara ayrılması.. kümeleme örnekleridir.[14]

1.7.7. Aykırılık Analizi (Outlier Analysis): Verilerin bazı algoritmalar ile kontrol edilerek verilerde aşırı olarak yanlış duran sapmış veya ayrıkı olanlarının bulunması sürecidir. Bu işlem sonucu bulunabilecek sıradışı olan veriler okuma,kayıt etme, ölçüm gibi fiziksel, elektronik ortamlarda oluşabilecek gürültüler sonucu elde edilen hatalı bilgilerdir. Veri madenciliği algoritmaları bu aykırı verileri en aza indirmek veya ortadan kaldırarak tahmin yöntemi ile sıradışı bilgi yerine gelebilecek en iyi veriyi tespit edip değiştirmek için kullanılır.

Örneğin kredi kartlarının olağandışı kullanımının tespiti, telekomunikasyon servislerinde olağandışı dolandırıcılık tesipiti, tıbbı tedavilerde olağandışı sonuçları bulmak.. için kullanılmaktadır.

1.8. Metin Madenciliği

Metin madenciliği çalışmaları girdi olarak metni kaynak olarak kabul eden bir tür veri madenciliği(data mining) çalışmasıdır. Diğer bir deyişle metin üzerinden yapısalleştırılmış (structured) olan veriyi elde etmeyi amaçlar. Örneğin metinlerin sınıflandırılması, kümelenmesi (clustering), varlık ilişki modellemesi (entity relationship modelling), sınıf taneciklerinin üretilmesi (production of granular taxonomy), metinlerden konu saptanması (concept/entity extraction), duygusal analiz (sentimental analysis), metin özetleme (document summarization) gibi çalışmaları hedefler. Bu hedeflere ulaşmak için metin madenciliği bağlamında yapılan çalışmalar kapsamında hece analizi (lexical analysis), enformasyon getirimi (information retrieval), örüntü tanıma (pattern recognition), etiketleme (tagging), enformasyon çıkarımı (information extraction), kelime frekans dağılımı (Word frequency distribution), veri madenciliği (data mining) ve hatta görselleştirme (visualization) gibi yöntemler kullanılır[15].

1.9. Metin Madenciliğinin Çalışma Alanları

1.9.1. Enformasyon Getirimi (Information Retrieval): Bu aşama külliyat(corpus) hakkında ön bilgi toplama aşamasıdır. Örnek olarak metin madenciliği uygulaması web üzerinden elde edilen veriler çerçevesinde gerçekleştirilecekse web adresleri, sayfaları veya dosya sistemi üzerindeyse dosya isimleri, tarihleri,dizin bilgileri ve kullanıcı bilgileri gibi bilgilerin toplandığı aşamadır.

1.9.2. Adlandırılmış Varlık Tanıma (Named Entity Recognition): Metin madenciliğinde, metin işleme aşamasında istatistiksel bazı özelliklerin çıkarımı için kullanılır. Örneğin, metnin içerisindeki yer isimleri, kişi isimleri, kısaltmalar semboller v.s. bu yöntemle bulunur. Metin madenciliği kapsamında yapılan çalışmalarda muhatap olunan metinler her zaman temiz değildir. Örneğin twitter, facebook mesajları, telefonlardan yollanan SMS mesajları gibi mesajların çoğunda yazım hataları ve kısaltmalar kullanılmaktadır. Metin madenciliği bu gibi ihtimallerinde göz önünde bulundurulması gereken bir alandır. Örneğin ‘’osmanbey’’ kelimesi, bir kişi ismi olabileceği gibi istanbulda bir semt ismi de olabilir.

Adlandırılmış varlık tanıma çalışmalarında, hedeflenen kelime gruplarının metin içerisinden çıkarılması, sayılması, yoğunluğunun bulunması, etiketlenmesi gibi işlemler yapılabilir.

1.9.3. Örüntüsü Tanımlı Varlıkların Bulunması (Pattern Identified Entities): Bazen metnin içerisindeki telefon numaraları, adresler,tarihler, e-posta adresleri gibi bilgiler metin madenciliğinde kurgulanan çalışma için önemli olabilir. Bu gibi bilgiler genelde bir örüntü içerir, içerikten bağımsız gramerler (context free grammers) ve düzenli ifadeler (regular expressions) tanımlanarak metin içerisinden tespit edilip kullanıma hazır hale getirilebilir[16].

1.9.4. Eş Atıf (Coreference): Bir olguya ve varlığa işaret eden (atıf eden) isim kelime gruplarını ve diğer terimlerin bulunması/ayrılmasını hedefler[17].

1.10. Doğal Dil İşleme(natural language processing): Dil insanlar arasında bir iletişim aracıdır. İnsan kendine ait bir özellik olan anlama ve fikir yürütme yeteneğinden dolayı, dili kolaylıkla işleyebilmektedir. Bilgisayarla dilin işlenebilmesi için, dilin tüm yönleriyle bilgisayar sistemlerine öğretilmesi gerekmektedir. Diller üzerinde yapılan çalışmalar, doğal dil işleme adı altında toplanmıştır[10]. DDİ'nin amacı, insanların kullandığı doğal dilleri çözümleyen, anlayan ve oluşturabilen sistemleri tasarlamak, uygulamak ve geliştirmektir [18,19]. Doğal dil işlemede en önemli problemlerden biri, cümle yapılarının ya da anlamların belirsizliğidir [18]. İnsanlar bu gibi belirsizliklerde daha önceki bilgi birikiminden yararlanarak çözümleme yapabilir ancak bir bilgisayarın bunu yapması beklenemez.

1.10.1. Biçimbirimsel Belirsizlik; bu belirsizlik türü, kelimelerin temel yapıları üzerinde belirsizlik olduğu durumlarda ortaya çıkmaktadır. "Koşuşturuyoruz"ifadesi ile "koşmak" mı yoksa "çok çalışmak" mı ifade edilmeye çalışılmıştır. İlk bakışta bunun anlaşılması zordur.

1.10.2. Sözdizimsel Belirsizlik; bu belirsizlik türü, kelimelerin cümle içindeki sıralanışlarından kaynaklanan belirsizlik türüdür. "Ali ile kavga ettim" cümlesi bu tür için en belirgin örneklerden biridir. Burada "Ali ile mi kavga edildiği", yoksa "Ali ile birlikte olup başkası ile mi kavga edildiği" belirsizdir.

1.10.3. Anlambilimsel Belirsizlik; kelime veya metinlerin anlamlarıyla ilgilidir. “çalmak” fiilinin öznesi bilinmiyorsa (saat veya hırsız), bunun için iki farklı anlam ortaya çıkmaktadır. [10]

1.11. Duygu Analizi (Sentimental Analysis)

2000'li yıllarda ortaya çıkan ve günümüzün önemli araştırma alanlarından birisi haline gelen duygu analizi; kişilerin varlıklar, olaylar üzerine fikirlerini, duygularını, değerlendirmelerini, değer biçmelerini, tutumlarını ve hislerini analiz etme işi olarak tanımlanmaktadır[20]. Metinlerde geçen duygusal ifadelerin çıkarılmasını amaçlar. En sık kullanılanı duygusal kutupsallıktır (sentimental polarity). Buna göre bir konu hakkında geçen mesajların veya yazıların olumlu veya olumsuz olmasına göre iki sınıfa ayrılması hedeflenir[21]

Araştırmacılar, duygu analizi problemlerini; doküman tabanlı, cümle tabanlı ve özellik tabanlı duygu analizi olacak şekilde üç ana başlığa ayırmaktadır. Belirli bir doküman üzerinde bahsi geçen olgu için dökümanı pozitif veya negatif olarak sınıflandırma işlemine doküman tabanlı duygu analizi iken bunu dokümanda bulunan her cümle için gerçekleştirme işlemine ise cümle seviyesinde duygu analizi denilir. Bir dökümanın negatif olarak sınıflandırılması o dökümanın tamamen olumsuz bir duygu içerisinde oluşturulduğu anlamına gelmez. Duygu belirten asıl hedef (özellik) belli değildir. Özellik ise dökümandaki temel olgunun özellikleridir. Yani negatif veya pozitif şekilde sınıflandırmak istediğimiz şeyler örneğin soru ve o sorular için yazılan cevaplar dökümanın özellikleridir. Tüm bunlar dikkate alındığında etkili bir duygu analizi için özelliklerin ve bu özellikleri niteleyen duygu ifadelerinin çıkartılmasını sağlayan bir modele ihtiyaç duyulmaktadır. Özellik tabanlı duygu analizinde, özellik ile kastedilen metinlerde duyguların ifade edildiği başlıklar yani; özellik üzerine yorum yapılan temel varlık, bu varlığın özellikleri, alt parçaları ve alt parçalarının özellikleri şeklinde ifade edilebilir [20,22]. "Personel çok çalışkandı." yorumunda "personel" kelimesi duygu analizindeki özelliğe karşılık gelmektedir.

1.12. Metin Madenciliği Aşamaları

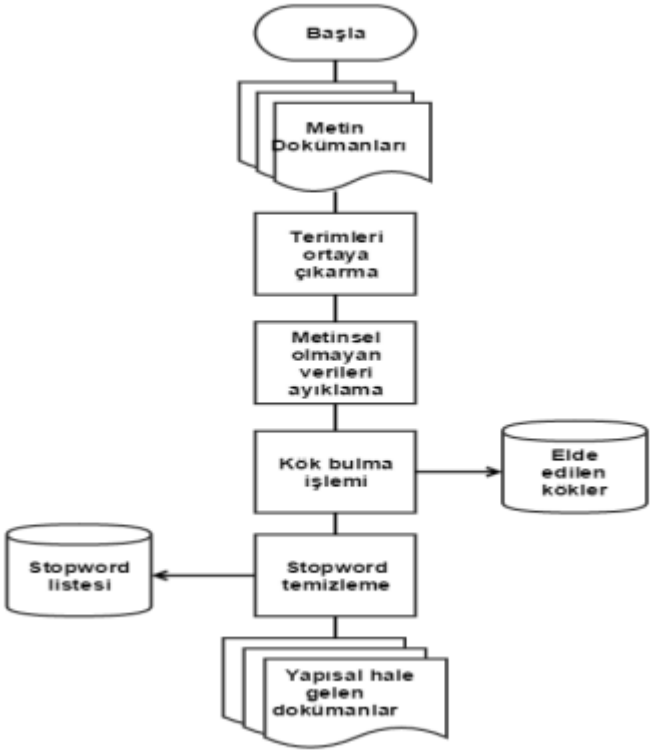
Bir metinsel veri kümesinden anlamlı bilgi edilinceye kadar gerçekleştirilen ardışık işlemler ya da daha çok bilinen tabirle ifade edilirse “metin madenciliği aşamaları” aşağıda verilen sıralı adımları içermektedir [23,24]

- Metinsel veri kümesinin oluşturulması
- Metin önişleme
- Öznitelik çıkarımı
- Metinlerin vektörel temsili
- Terim ağırlıklandırma
- Metin analizi
- Değerlendirme ve yorum

1.12.1. Metin Önişleme

Metin önişleme aşaması, metin madenciliği işleminde analiz yapmadan önce kelimeler üzerinde kelimeleri analiz için hazır hale getirmek için uygulanan işlemleri kapsayan önemli bir süreçtir. Önişleme aşamasında uygulanan süreçler yapılacak olan analiz sürecinin başısını doğrudan etkiler, gerçekleştirilecek olan deneyden önce deneyde kullanılacak metinlerin analiz sonrası yapılacak çıkarımı en iyi şekilde yansıtacak hassaslığa sahip olmalıdır. Özellikle gerçek veri analizine dayanan süreçlerde veri eksik, gürültülü veya tutarsız olabilir.

Metin önişleme adımları verinin eksik nitelik değerlerini tamamlama, metinlerdeki aykırılıkların bulunması, gürültülü verinin daha homojen hale getirilmesi, tutarsızlıkların giderilmesi, anlam ifade etmeyen verilerin silinmesi ve boyut indirgeme gibi birtakım sıralı işlemleri içermektedir. İşlenen metinlerin ve gerçekleştirilecek olan deneysel analizin türüne bağlı olarak metin önişleme adımları değişiklik gösterebilir. Eğer işlenen metinler web sayfalarından oluşuyorsa html etiketlerinin, sosyal medya mesajlarından oluşuyorsa o sosyal ağa özgü karakterlerin ve bazı özel terimlerin, elektronik postalardan oluşuyorsa kişisel bilgilerin anlamlı içeriklerden ayrıştırılması gerekir. [3,24,25].



Şekil 2. Ön işleme aşamaları

1.12.1.1. Dizge Parçalama

Dizge parçalama(tokenization) işlemi metinsel içeriği kelimelere ayırarak anlamlı öznitelikler elde edilmesi işlemidir. Böylece veri setinde yer alan her bir metinsel veri kelime olarak temsil edilir.

1.12.1.2. Gereksiz-Uygunsuz Verilerin Silinmesi

Bir sonraki aşama veri setinde mevcut ise noktalama işaretleri, özel etiketler, anlamsız karakterler ve web bağlantıları metinden silinerek metinler üzerinde veri temizleme işlemi yapılmış olunur.

1.12.1.3. Durak Kelimelerin Silinmesi

Bir diğer önemli aşama elimizde bulunan kelime sayısını küçültmek için yapılması gerekilen durak kelimelerin (stop words) silinmesi işlemidir. Durak kelimeleri bir dilde yaygın olarak kullanılan ve genellikle tek başına kullanıldığında bir anlam ifade etmeyen kelimelerdir (örneğin, İngilizce için: “and”, “or”, “with”, “there”, “she” gibi kelimeler, Türkçe için: “ve”, “ile”, “veya”, “ne”, “için” gibi kelimeler). Bu nedenle metin analizine dayalı çalışmalarda genellikle metinsel içerikler durak kelimelerinden temizlenir. Ancak bu

kelimeler çıkarılırken her dile özgü durak kelimelerini içeren bir listeye ihtiyaç duyulmaktadır. Durak kelimelerinin çıkarılması işlemi yapılan analizin türüne ve öznitelik çıkarım modeline bağlı olarak değişiklik göstermektedir [3,24,25].

1.12.1.4. Gövdeleme

Bir diğer önişleme aşaması ise gövdeleme (stemming) işlemidir. Basitçe köke indirgeme olarakta düşünülebilir. Örneğin, gözlemlemek, gözledim, gözlemler, gözlemsel gibi kelimeler gözlem kökünden türetilmiş kelimelerdir. Bu haliyle kelime uzayında beş farklı kelime olarak temsil edilecek bu kelimeler, gövdeleme işlemi sonrası tek bir kelime ("gözlem") ile temsil edilir. İngilizce metinler için Porter, Stemmer, SnowballStemmer, WordNetLemmatizer gibi gövdeleme algoritmaları yaygın olarak kullanılmakla birlikte, Türkçe gibi sondan eklemeli bir dil için Zemberek kütüphanesi düşünülebilir ancak başarı yüzdesi istenen seviyede değildir [3,24,25].

1.12.2. Öznitelik Çıkarımı

Metin içerikli veri kümeleri genel olarak içerdikleri kelimelerin sıralamaları ihmal edilerek içerdiği kelimelerle ve bu kelimelerin dizilimlerinden ortaya çıkan kelime grupları ile belirlenir. Burada metinleri temsil etmek için öznitelik olarak seçilen kelime ya da kelime grupları terim olarak adlandırılır.Kelime torbası(BoW: Bag of Words) modeli, N-gram modeli,Karım dolabı(Hashing trick) ve spam süzgeci(spam filtering) gibi bir çok öznitelik çıkartma yöntemi mevcuttur.

Öznitelik belirleme yöntemlerinden en yaygın olarak kullanılanı kelime torbası(çantası) (BoW: Bag of Words) modelidir. BoW modelinde kelimelerin elimizdeki metin uzayında bulunan tüm kelimelerin görülme sayıları hesaplanarak bir havuzda tabiri caiz ise bir çantada toplanır. Kelimelerin sıraları önemli olmayan bu modelde her kelimenin görülme frekansı belirlenmiş olur. Dile bağımlı bir model olduğu için elimizdeki metinsel verinin diline uygun şekilde önişleme uygulanmalıdır.

1.13. Metinlerin Vektörel Temsili

Metin madenciliğinde kullanılan nicel analize dayalı algoritmaların nitel veri olan metinler üzerinde uygulanabilmesi için metinlerin analize uygun sayısal formata dönüştürülmesi gerekmektedir. Nicel metin analizlerinde yaygın bir yaklaşım olan vektör uzay modelinde, dokümanlar çok boyutlu vektör uzayında bir terim vektörü olarak temsil

edilmektedirler. Dokümanlar kümesindeki ayrık terim (kelime) sayısı vektör uzayının boyutunu belirlemektedir. Bu yaklaşımda veri setindeki her bir metin bir vektör ile temsil edilir ve tüm metinler için oluşturulan her bir vektörün boyutu aynıdır. Bu vektörlerin boyutu terim uzayındaki toplam terim sayısına eşittir. Her bir metin için oluşturulan terim vektörü eşit boyutlu olmasına rağmen içerdikleri terimlere göre vektörlerin terim değerleri farklılık gösterir [3,26-28].

1.13.1. Terim Sıklığı(Term Frequency) (TF)

Bir doküman içerisinde geçen terim ağırlıklarını hesaplamak için kullanılan yöntemdir. Bir dizi İngilizce metin belgemiz olduğunu ve hangi belgenin "the brown cow" sorgusuyla en alakalı olduğunu sıralamak istediğimizi varsayalım. Başlamanın basit bir yolu, üç kelime olan "the", "brown " ve " cow" kelimelerini içermeyen belgeleri elimine etmektir, ancak bu yine de birçok belge bırakmaktadır. Bunları daha da ayırt etmek için, her bir terimin her belgede kaç kez meydana geldiğini sayabiliriz; belgede bir terimin gerçekleşme sayısına terim sıklığı denir. Ancak, belgelerin uzunluğunun büyük ölçüde değiştiği durumlarda, genellikle ayarlamalar yapılır. Ağırlıklandırmanın ilk şekli, şu şekilde özetlenebilen Hans Peter Luhn'a (1957) ait olup şu şekildedir:[29]

Belgede meydana gelen bir terimin ağırlığı, terim sıklığı ile orantılıdır.

Terim sıklığı (TF) ağırlığı değişkenleri

Ağırlıklandırma Şeması	TF ağırlığı
ikili	$0, 1$
ham sayı	$f_{t,d}$
terim sıklığı	$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$
log normalizasyon	$\log(1 + f_{t,d})$
ikili normalizasyon 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
ikili normalizasyon K	$K + (1 - K)\frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Şekil 3. TF ağırlığı değişkenleri

- İkili, doküman içerisinde terimin olup olmadığını
- Ham Frekans, terimin dokümanda geçme sayısı / dokümandaki kelime sayısı (Dokümanın uzunluğu kaliteli bir sonuç elde etmeyi engeller)
- Log normalizasyonu – logaritmik olarak normalizasyon
- Double Normalization 0,5 burada 0,5–1 arasında bir değer oluşturuyor. Raw Freq / Maksimun geçen Terim Raw Freq bölerek doküman ne kadar uzun olursa olsun terim'in diğer terimlere olan oranını bularak frekansı normalize etmektedir.
- Double Normalization K –
- Tf (t, d) frekansı söz konusu olduğunda, en basit seçim, bir belgedeki bir terimin ham sayımını kullanmaktır, yani, t teriminin d belgesinde meydana gelme sayısıdır. Ham sayıyı ft, d ile gösterirsek, en basit tf şeması tf (t, d) = ft, d'dir. Diğer olasılıklar arasında [30]
- Boole "frekansları": t d ve 0'da aksi takdirde tf (t, d) = 1;
- belge uzunluğu için ayarlanan terim frekansı: ft, d ÷ (d cinsinden kelime sayısı)
- logaritmik olarak ölçeklenmiş frekans: tf (t, d) = log (1 + ft, d); [31]

1.13.2. Ters Belge Sıklığı (Inverse Document Frequency) (IDF)

"The" Terimi çok yaygın olduğu için, terim sıklığı, "kahverengi" ve "inek" terimlerine daha fazla ağırlık vermeden, "the" kelimesini daha sık kullanan belgeleri yanlış vurgulama eğiliminde olacaktır. "Kahverengi" terimi, daha az yaygın olan "kahverengi" ve "inek" kelimelerinin aksine, alakalı ve alakasız dokümanları ve terimleri ayırt etmek için iyi bir anahtar kelime değildir. Bu nedenle, belge setinde çok sık meydana gelen terimlerin ağırlığını azaltan ve nadiren ortaya çıkan terimlerin ağırlığını arttıran ters bir belge frekans faktörü eklenir. Karen Spärck Jones (1972), terim ağırlıklandırmasının temel taşı haline gelen Ters Belge Frekansı (idf) adı verilen terim özgüllüğünün istatistiksel bir yorumunu tasarlamıştır: [32] Bir terimin özgüllüğü, meydana geldiği belge sayısının ters bir fonksiyonu olarak ölçülebilir.

Ters Belge Sıklığı(idf) Ağırlığının Değişkenleri

Ağırlıklandırma Şeması	idf ağırlığı ($n_t = \|\{d \in D : t \in d\}\|$)
tekli (birli)	1
Ters belge sıklığı	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
Ters belge sıklığı smooth	$\log\left(\frac{N}{1+n_t}\right)+1$
Ters belge sıklığı Maks	$\log\left(\frac{\max_{\{t' \in d\}} n_{t'}}{1+n_t}\right)$
Olasılıksal ters belge sıklığı	$\log \frac{N-n_t}{n_t}$

Şekil 4. Ters doküman sıklığı (idf) ağırlığı değişkenleri

Ters belge sıklığı, kelimenin ne kadar bilgi sağladığının, yani tüm belgelerde ortak veya nadir olup olmadığının bir ölçüsüdür. Kelimeyi içeren belgelerin logaritmik olarak ölçeklendirilmiş ters kısmıdır (toplam belge sayısını terimi içeren belge sayısına bölerek ve daha sonra bu bölümün logaritmasını alarak):

$$\operatorname{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Şekil 5. Idf formulü

Burada;

- N: külliyattaki toplam belge sayısı N=|D|
- $|\{d \in D : t \in d\}|$: t teriminin göründüğü belge sayısı. $\operatorname{tf}(t,d) \neq 0$ Terim corpus'ta değilse, bu sıfıra bölünmeye yol açacaktır. Bu nedenle paydayı $1 + |\{d \in D : t \in d\}|$ olarak ayarlamak yaygındır

1.13.3. Terim Sıklığı- Ters Belge Sıklığı (TF-IDF)

tf – idf şu şekilde hesaplanır:

$$\operatorname{tfidf}(t, d, D) = \operatorname{tf}(t, d) \cdot \operatorname{idf}(t, D)$$

Tf – idf cinsinden yüksek bir ağırlığa, yüksek bir belge sıklığı (verilen belgede) ve tüm belge koleksiyonunda terimin düşük belge sıklığı ile ulaşılır; ağırlıklar dolayısıyla ortak terimleri filtreleme eğilimindedir. İdf log fonksiyonunun içindeki oran her zaman 1'den büyük veya ona eşit olduğundan, idf (ve tf – idf) değeri 0'dan büyük veya ona eşittir. Bir terim daha

fazla belgede göründüğünden, logaritma içindeki oran 1'e yaklaşır idf ve tf – idf değerlerini 0'a yaklaştırır.

Önerilen Tf-Idf Ağırlıklandırma Şeması

Ağırlıklandırma şemaları	Belge Terim Ağırlığı	Sorgu Terim Ağırlığı
1	$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0.5 + 0.5\frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$
2	$1 + \log f_{t,d}$	$\log\left(1 + \frac{N}{n_t}\right)$
3	$(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$

Şekil 6. Önerilen Tf-Idf Ağırlıklandırma Şeması

1.14. Gizli Anlamsal Analiz

Gizli anlamsal analiz (GAA), doğal dil işlemede, özellikle de dağıtım anlamsallığında, bir dizi belge ile içerdikleri terimler arasındaki ilişkileri, belgeler ve terimlerle ilgili bir dizi kavram üreterek analiz etme tekniğidir. GAA, anlam bakımından yakın olan kelimelerin benzer metin parçalarında (dağılım hipotezi) olacağını varsayar. Belge başına kelime sayımlarını içeren bir matris (satırlar benzersiz kelimeleri temsil eder ve sütunlar her belgeyi temsil eder) büyük bir metin parçasından oluşturulur ve benzerlik yapısını korurken satır sayısını azaltmak için tekil değer ayrışması (TDA) adı verilen bir matematiksel teknik kullanılır sütunlar arasında. Belgeler daha sonra herhangi iki sütun tarafından oluşturulan iki vektör (veya iki vektörün normalleştirmeleri arasındaki nokta çarpımı) arasındaki açının kosinüsü alınarak karşılaştırılır. 1'e yakın değerler çok benzer belgeleri, 0'a yakın değerler çok farklı belgeleri temsil eder[33].

1.14.1. Anlamsal Hashing

Anlamsal karma belgeler, anlamsal olarak benzer belgeler yakındaki adreslerde yer alacak şekilde bir sinir ağı vasıtasıyla bellek adreslerine eşlenir. Derin sinir ağı esasen çok sayıda belgeden elde edilen kelime sayımı vektörlerinin grafik bir modelini oluşturur. Bir sorgu belgesine benzer belgeler daha sonra sorgu belgesinin adresinden sadece birkaç bit farklı olan tüm adreslere erişerek bulunabilir. Karma kodlamanın verimliliğini yaklaşık eşleştirmeye genişletmenin bu yolu, en hızlı akım yöntemi olan yere duyarlı karma işleminden çok daha hızlıdır[34] .

1.14.2. Gizli Anlamsal İndeksleme

Gizli anlamsal indeksleme (GAI), yapılandırılmamış bir metin koleksiyonunda yer alan terimler ve kavramlar arasındaki ilişkilerdeki örüntüleri tanımlamak için tekil değer ayrıştırma (TDA) kullanan bir indeksleme ve geri alma yöntemidir. GAI, aynı bağlamlarda kullanılan kelimelerin benzer anlamlara sahip olma ilkesine dayanmaktadır. GAI'nin temel bir özelliği, benzer bağlamlarda ortaya çıkan terimler arasında ilişki kurarak bir metin gövdesinin kavramsal içeriğini çıkarma yeteneğidir. [35]

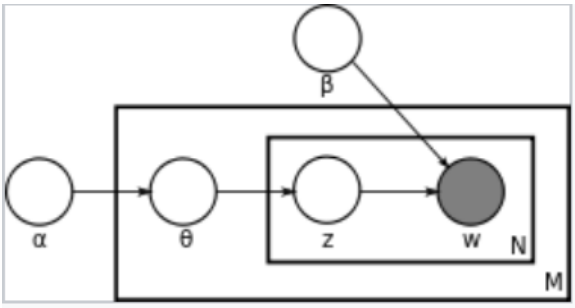
GAI, Boole anahtar kelime sorgularının ve vektör alanı modellerinin en sorunlu kısıtlamalarından biri olan hatırlamayı artırarak eş anlamlılığın üstesinden gelmeye yardımcı olur. [36] Eşanlamlılık genellikle belgelerin yazarları ve bilgi erişim sistemleri kullanıcıları tarafından kullanılan kelime dağarcığındaki uyumsuzlukların nedenidir. [37] Sonuç olarak, Boole veya anahtar kelime sorguları genellikle alakasız sonuçlar döndürür ve alakalı bilgileri kaçırır.

GAI, otomatik belge kategorizasyonu gerçekleştirmek için de kullanılır. Aslında, birkaç deney GAI ve insanların metni işleme ve kategorize etme biçimi arasında bir takım korelasyonlar olduğunu göstermiştir. [38] Belge kategorizasyonu, belgelerin, kategorilerin kavramsal içeriğine benzerliklerine dayanarak, önceden tanımlanmış bir veya daha fazla kategoriye atanmasıdır. [39] GAI, her kategori için kavramsal temeli oluşturmak amacıyla örnek belgeler kullanır. Sınıflandırma işlemi sırasında, kategorize edilen belgelerde yer alan kavramlar, örnek öğelerdeki kavramlarla karşılaştırılır ve içerdikleri kavramlar ile içerilen kavramlar arasındaki benzerliklere dayanarak dokümanlara bir kategori (veya kategoriler) atanır. örnek belgelerde.[43]

1.14.3. Gizli Dirichlet Tahsisi

Doğal dil işlemede, gizli Dirichlet tahsisi (GDT), verilerin bazı bölümlerinin neden benzer olduğunu açıklayan gözlemsiz gruplar tarafından gözlem kümelerinin açıklanmasına izin veren üretken bir istatistiksel modeldir. Örneğin, gözlemler belgelere toplanan kelimeler ise, her belgenin az sayıda konunun bir karışımı olduğunu ve her kelimenin varlığının belgenin konularından birine atfedilebileceğini öne sürer. Makine öğrenmesi ve metin madenciliği uygulamalarında büyük önem kazanan ve en temel ve en popüler konu modelleme yöntemlerinden birisi olan Gizli Dirichlet Tahsisi(Ayrımı) (Latent Dirichlet Allocation-LDA), doküman gibi ayrık verileri modellemek ve dokümanı meydana getiren

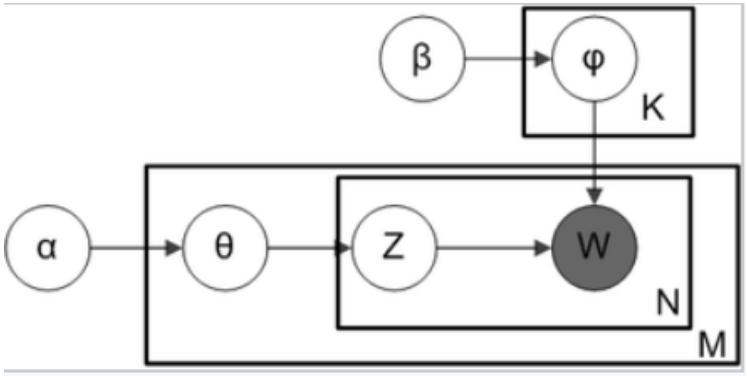
konuları ortaya çıkarmak için kullanılan üretici grafiksel modeldir [40,41]. Mühendislikte GDT 'nin bir örneği, belgeleri otomatik olarak sınıflandırmak ve çeşitli konularla ilişkilerini tahmin etmektir.



Şekil 7. Plaka gösterimi

Olasılıksal grafik modelleri (OGM'ler) temsil etmek için sıklıkla kullanılan plaka gösterimi ile birçok değişken arasındaki bağımlılıklar kısaca yakalanabilir. Kareler, tekrarlanan varlıkları olan kopyaları temsil eden "plakalar" dır. Dış plaka belgeleri temsil ederken, iç plaka belirli bir belgedeki tekrarlanan kelime pozisyonlarını temsil eder; her pozisyon bir konu ve kelime seçimi ile ilişkilendirilir. Değişken adları aşağıdaki gibi tanımlanır:

- M, belge sayısını belirtir
- N, belirli bir belgedeki kelime sayısıdır (i belgesinde N_i kelime vardır)
- α, belge başına konu dağılımından önceki Dirichlet parametresidir
- β, konu başına kelime dağıtımından önce Dirichlet'in parametresidir
- θ_i , i dokümanı için konu dağılımıdır
- φ_k, konu k için kelime dağılımıdır
- z_{ij} i belgesindeki j-th kelimesinin konusudur
- w_{ij} belirli bir kelimedir.
- K, gizli konuların sayısını belirtir.
- V, oluşturulan sözlükte bulunan toplam kelime sayısıdır.



Şekil 8. Plaka gösterimi 2

W'nin grileşmesi, w_{ij} kelimelerinin gözlemlenebilir tek değişken olduğu ve diğer değişkenlerin gizli değişken olduğu anlamına gelir. Bir konudaki kelimeler üzerindeki olasılık dağılımının çarpık olduğu sezgisini takiben, konu-kelime dağılımını modellemek için daha önce seyrek bir Dirichlet kullanılabilir, böylece sadece küçük bir kelime kümesi yüksek olasılığa sahiptir. Ortaya çıkan model, bugün GDT 'nin en yaygın uygulanan çeşididir. Bu model için plaka gösterimi Şekil 8 de gösterilmiştir, burada K konuların sayısını gösterir ve $\varphi_1, \ldots, \varphi_K$, Dirichlet tarafından dağıtılan konu-kelime dağılımlarının parametrelerini saklayan V-boyutlu vektörlerdir (V, kelime dağarcığı)[42].

$$P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j) P(W_{j,t} \mid \varphi_{Z_{j,t}})$$

Şekil 9. GDT algoritması istatistiksel formulü

2. YAPILAN ÇALIŞMALAR

2.1. Giriş

Bu bölümde bu tez çalışması kapsamında icra edilen veri setleri, kullanılan yöntemler ve çıkarımlar anlatılmıştır. İlk olarak takip edilen süreçler ve planlamaların ardından çalışılan konuya ait herhangi paylaşılan veri seti olmadığı için verilerin toplanması, toplanan verilere uygulanan ön işleme aşamaları sonucu verinin hazırlanması ve doküman-terim matrisi oluşturularak yöntem ve çıkarım konuları altında yapılan işlemlerin kavramsal açıklamaları ve GDT tabanlı olasılıksal konu modelleme yaklaşımının elde edilen veri seti üzerinde uygulanması aşamaları açıklanmıştır.

2.2. Projede Kullanılacak Süreçler

2.2.1. Çevik Süreçler

Yazılım sistemlerini etkili ve olabildiğince verimli bir şekilde modellemeye ve dökümantasyonunu yapmaya yönelik pratikliğe ve esnekliğe dayalı yöntemlere verilen genel ada çevik modelleme denir. Geleneksel diğer modellere göre prensipler, pratikler sayesinde daha esnek ve kullanışlı bir haldedir.

Çevik modellemenin başlıca özelliği veri modelleri ve ara yüzü modelleri gibi modellerin tekniklerini yani ayrıntılarını söylemektense bu tekniklerin nasıl uygulanması gerektiğini söylemesidir. Örneğin bir projenin test edileceğinin söylenmesi hususunda o projenin nasıl, hangi koşullarda veya hangi araçlar ile test edileceğini söylemez. Bu sayede bir projenin etkili, hızlı bir şekilde ortaya çıkması ve paydaşların isteklerinin olabildiğince esnek ve kolay bir şekilde elde edilmesi planlanır.

2.2.2. Hangi Durumlarda Çevik Modelleme Kullanılabilir

- Paydaşlar tarafından gelen isteklerin sürekli değişim gösterdiği, tahmin edilemeyen senaryoların bulunduğu süreçlerde
- Projenin parçalarının önceden tasarlanıp ardından geliştirlmesinin hemen gerekmesi ve önceden ne yapılacağının planlanması konusunda yeterinde zaman, detaylı yol haritası ve tasarımın ne kadar süreceğinin bilinemediği süreçlerde
- Analiz, tasarım ve test etme aşamalarının ne kadar süreceğinin önceden belirlenemediği süreçlerde

- Yazılım geliştirme ekibi paydaşlarının birbirleri arasındaki hiyerarşiye önem verdiği ve iyi iletişime sahip olduğu durumlarda

2.2.3. Başlıca Çevik Süreç Modelleri:

- Sınırsal programlama(Extreme Programming-XP)
- Çevik Birleştirilmiş Süreç(Agile Unified Process)
- Scrum
- Test Güdümlü Geliştirme(Test-driven Development)
- Çevik bilgi Metodu(Agile Data Method)
- Özellik güdümlü geliştirme(Feature-driven Programming)

2.3. Scrum

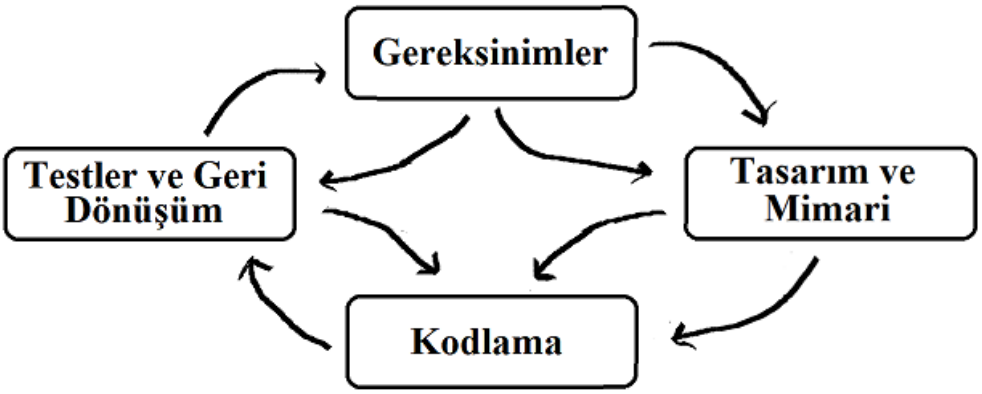
Scrum, araştırma, satış, pazarlama ve ileri teknolojiler de dahil olmak üzere diğer alanlarda kullanılmasına rağmen[44], yazılım geliştirmeye öncelikli olarak karmaşık ürünleri geliştirmek, sunmak ve sürdürmek için çevik bir çerçevedir.[45] Çalışmalarını sprint olarak adlandırılan, bir aydan fazla ve en yaygın olarak iki haftadan daha uzun süren yinelemelerde tamamlanabilecek hedeflere ayıran on veya daha az üyeli ekipler için tasarlanmıştır. Scrum Takımı, günlük scrum adı verilen 15 dakikalık zamanlı günlük toplantılardaki ilerlemeyi izler. Sprint sonunda takım, yapılan işi göstermek için sprint incelemesi ve sürekli iyileştirmek için geri dönüşlü incelenen sprint verilerini elinde tutar. Böylelikle sürekli olarak daha önce yapılan işler incelenir, yeni beliren istekler doğrultusunda süreçler ve yapılacak işler tekrar planlanır. Böylece geliştrme sırasında esneklik elde edilmiş olunup gereksiz iş yükünden kurtulunmuş olunur.

2.4. Projede Kullanılan Süreçler

Proje geliştirilirken scrum yaklaşımı kullanılacaktır. Çevik süreçlerden birisi olan scrum sayesinde danışmanlar ile haftalık yapılan kısa toplantılar sayesinde

- *Geçen hafta ne yaptım?,*
- *Bu hafta ne yapacağım?,*
- *Beni engelleyen sorunlar var mı?*
- *Var ise sorunların çözümleri konusunda ne gibi adımlar izlemeliyim*

gibi sorulara yanıtlar aranarak süreç hızlandırılabilecek ve yazılım geliştirilirken kopukluklar ortadan kalkacak.



Şekil 10. Agile Model

Ayrıca scrum bir çevik (agile) süreç modeli olduğu için sürekli bir ürün elde edilecek ve elde edilen ürün çerçevesinde testler ve gereksinim analizleri tekrarlı şekilde yapılarak proje genişletilebilecek, eksik olan veya eklenmek istenilen gereksinimler, özellikler esnek bir yapıda eklenebilecektir.

2.4.1. Gereksinim: Projede kapsanması planlanan özellikler ve yapılacak işlerin belirlenmesinin yapıldığı kısımdır. Kısaca gereksinim çıkartılan kısımdır. Bir önceki sprintte yapılan işin incelenmesi, ve bu incelemeler doğrultusunda yapılacak çıkarımlar sayesinde, ürün her sprint aşamasında özellik ve kullanım özellikleri açısından daha da gelişecektir.

2.4.2. Analiz: Gereksinim aşamasında kurgulanan gereksinimlerin raporlanarak kesin hale getirilip detaylandırılmasının yapıldığı ve bu gereksinimler hakkında ihtiyaçların belirlendiği, analizlerin yapıldığı aşamadır.

2.4.3. Tasarım: Proje geliştirilirken analiz aşamasında yapılan araştırmalar ışığında projenin içermesi gereken modüllerin, süreçlerin, işlemlerin ve mimarilerin belirlenerek bir bütün olarak tasarlanması ve projede çalışan her yazılımcının rahatça anlayacağı şekilde diyagramlar şekilde dökümante etme aşamasıdır.

2.4.4. Kodlama: Tasarım aşamasında kurgulanan diyagramlar doğrultusunda geliştirilmesi gerekilen modüllerin, arayüzlerin yazılım geliştirici takımı tarafından geliştirilmesi aşamasıdır.

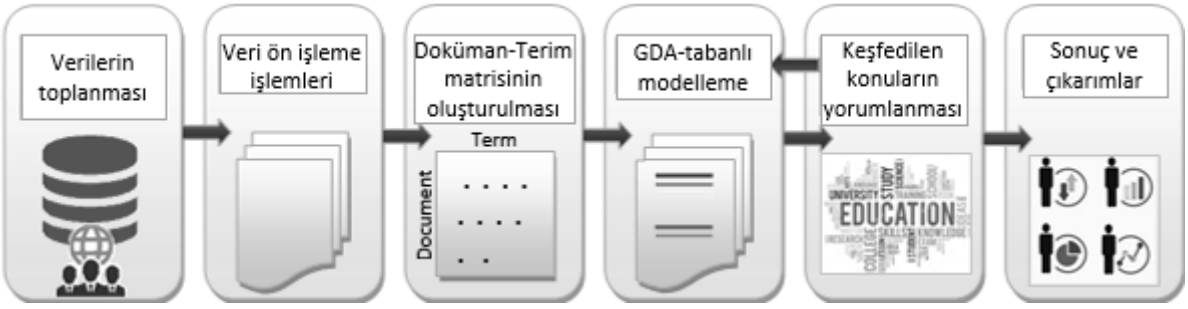
2.4.5. Test: Geliştirme sonucunda, geliştirilen kısım ve bu kısmın diğer fonksiyonlar ile birlikte doğru şekilde çalışıp çalışmadığının test edildiği kısımdır.

2.4.6. Kabul: Yapılan geliştirmelerin proje yöneticisine sunumu ve gereksinimlerin sağlandığının teyit edildiği ve onaylandığı aşamadır.

2.4.7. İnceleme: Bütün sürecin incelendiği, geliştirme sürecinin analizinin yapıldığı kısımdır. Eğer uygun görüşürse bir sonraki aşama için alınacak önlemler ve aksiyonlar belirlenir.

2.5. Çalışmanın Genel İşleyişi

Çalışmanın genel çerçevesi metin tabanlı veri seti üzerinde GDT-tabanlı konu modelleme ve yapılan konu modelleme sonucu oluşan sonuçlar üzerinde anlamsal analizin yapılması şeklinde kurgulanmış ve uygulanmıştır.

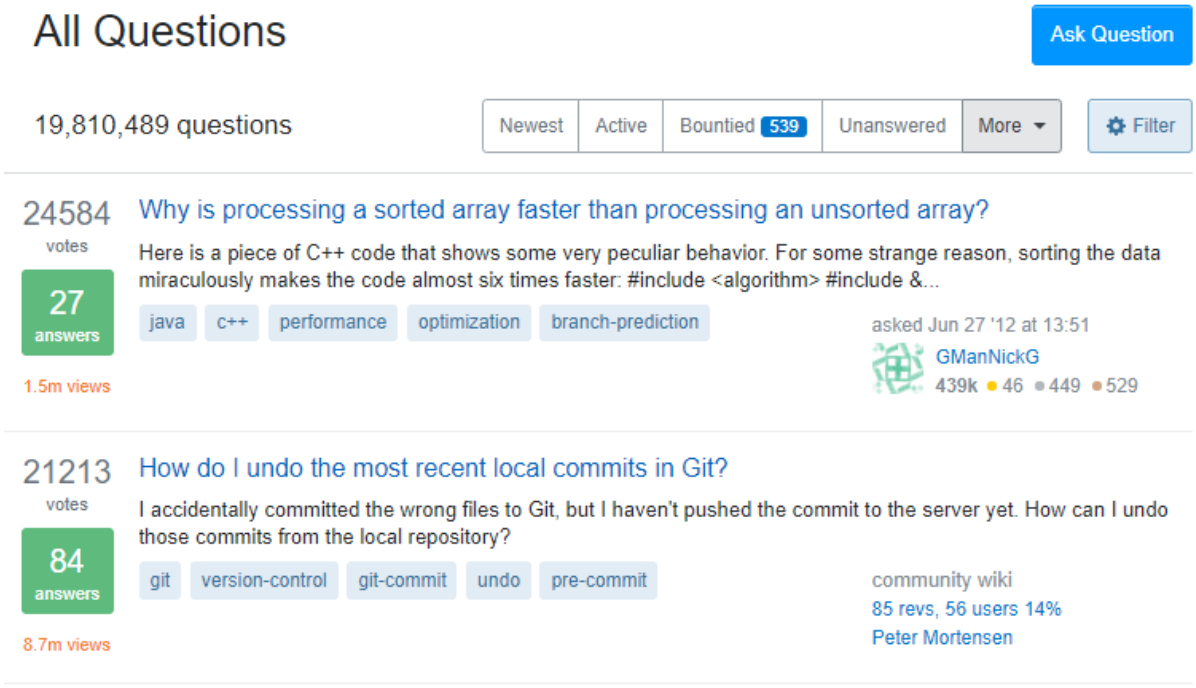


Şekil 11. Çalışmanın genel çerçevesinin akış şeması [8]

Birinci aşamada yazılım geliştiricilerin karşılaştıkları soruları ve bu sorulara diğer yazılım geliştiriciler tarafından verilen cevapları içeren stack overflow üzerinden elde edilen veriler ile veri setleri aylık bazda oluşturulmuştur. Ardından boyut azaltma ve analizin doğruluğunu ve başarısını arttırmaya yönelik veri seti üzerinde önişleme adımları uygulanmıştır. Ardından veri setleri sayısal analizin gerçekleştirilebilmesi adına doküman-terim matrisine (DTM) dönüştürülmüş ve bu işlem sonucu oluşsan DTM gizli anlamsal yapıların ve konuların keşfi için GDT tabanlı olasılıksal konu modelleme yaklaşımına dayalı anlamsal analizde kullanılmıştır. Daha sonra istenen düzeye gelene kadar anlamsal analizler değişik değişkenler ve parametreler ile tekrarlanmıştır. Son olarak analizler sonucu elde edilen sonuçlar ve bu sonuçlar doğrultusunda çıkarımlar yapılmıştır.

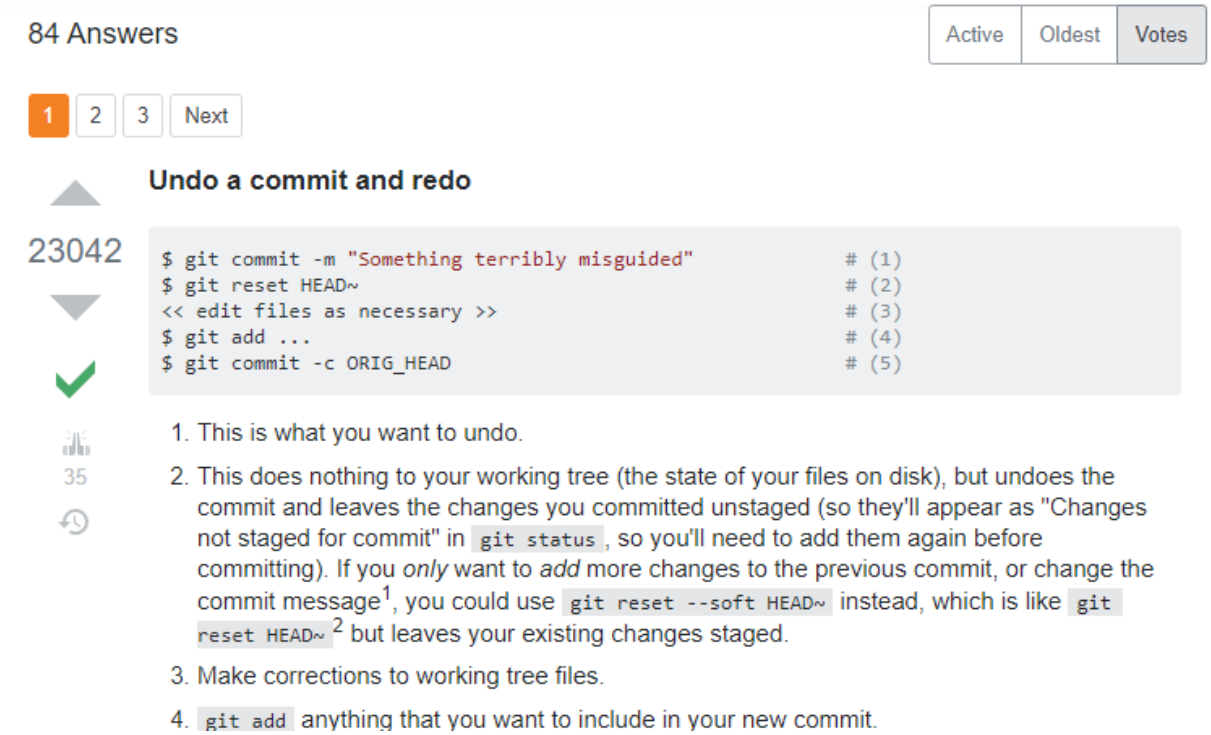
2.6. Verilerin Elde Edilmesi

Yazılım geliştiriciler teknolojinin hızlı geliştiği ve yazılım geliştirmenin giderek karmaşıklaştığı bu dönemde, yine teknolojinin hızlı gelişmesi sayesinde bilginin dünya çapında internet üzerindeki hızlı yayılımı kapsamında bu karşılaşılan karmaşıklığın çözümü olarak interneti ve internet üzerindeki yazılım geliştiricilerin karşılaştıkları soruları ve bu sorulara diğer yazılım geliştiriciler tarafından verilen cevapları içeren stack overflow' u etkin şekilde kullanmaktadırlar. Bu kapsamda bu çalışmanın veri kaynağı stack overflow üzerinde paylaşılan 2019 yılına ait bütün soru ve bu sorulara verilen kullanıcı cevaplarını içermektedir. Stack overflow üzerinde bulunan soru ve cevaplar direkt yazılım geliştirmeye yönelik olduğundan dolayı 2019 yılı boyunca aylık olarak yazılım geliştiricilerin tartıştığı konuların modellenmesinde çok etkili ve vazgeçilmez verilerdir.



Şekil 12. Örnek stack overflow soruları başlıklı görünümleri

Şekil 13. Örnek soru içeriği



Şekil 14. Örnek bir önceki şekil bağlamında sorulan soruya verilen örnek bir cevap

Stack overflow verileri stackexchange ve google ın BigQuery platformu üzerinden erişim sağlanabilmektedir. Ancak buralardan elde edilen verilen çok karışık olmakla birlikte çok büyük boyutlara ve istenilen kıstaslarda veriye erişmekten ziyade araştırma boyunca kullanılacak süreçler dışında gereksiz olarak tanımlanabilecek pek çok veriyi de yanında getirmektedir. Bu bağlamda veriye erişme konusunda bir başka yöntem olarak stack overflow un kendi altyapısında bulunan api ile token ile birlikte günlük 10.000 istek sayısına sahip bir sistem ile verilere erişmekte mümkün.

Bu kapsamda Ocak 2019 ile Aralık 2019 tarihleri arasındaki 12 aylık süreçte yayınlanan ingilizce soru ve cevapları ile oluşturulan 1880141 adet soru ve bu sorulara verilen 1962196 adet cevaptan oluşan yazılım geliştirme odaklı soru-cevap içermektedir.

Herhangi bir kısıtlama bulunmaksızın 2019 yılı boyunca yayınlanan bütün soruların soruları, cevapları, etiketleri, başlıkları ve question_id bilgileri filtrelenerek stack overflow api üzerinde bir yapı oluşturularak alınmış ve ilgili soruların cevapları soruların bitimine eklenerek veri seti 12 aylık veriyi kapsayacak şekilde 12 parçaya bölünmüştür.Stack overflow api ile stack overflow üzerinden elde edilen verilerden Ocak ayına ait olan ilk 25 kayıt aşşağıdaki şekilde gösterilmiştir.

	A	B	C	D	E
1	tags	question_id	body_markdown	title	
2	['javascript', 'gulp'	54451423	I keep getting `Did yc	Gulp did you forget to signal asyn	
3	['django', 'django-	54451422	I extended django ac	Block content on admin template	
4	['intellij-idea', 'ku	54451417	IntelliJ IDEA 2018.3 d	Is it possible to make IntelliJ recc	
5	['python', 'pyqt5']	54451416	This is second windc	QMainWindow disappears when	
6	['java', 'shutdown-	54451413	I have a ServerSocket	Java application terminating with	
7	['c++', 'matlab', 'm	54451410	I want to access the	Read custom class in C applicatio	
8	['python', 'django'	54451406	I'm using Pythor	How do I run Django unittests in P	
9	['python', 'regex']	54451403	I have a block of text	How to set a stop criteria for rege	
10	['mysql', 'sql', 'dat	54451401	I'm trying to get	SQL JAVA I'm trying to get the	
11	['docker', 'docker-(	54451393	If I specify in my docl	How can I identify what tags are a	
12	['c', 'scanf']	54451392	I am new to C progra	How do I use fscanf to capture spa	
13	['python', 'numpy',	54451391	I have a h5 file which	How to see all values of long arra	
14	['c++', 'c++11', 'lam	54451381	I am trying to find the	Using lambda function to find a m	
15	['dart', 'flutter']	54451380	When I use `routes`	Widget's didUpdateWidget m	
16	['c#', 'linq', 'datata	54451374	I have incoming data	Is it really this hard to just do a LE	
17	['vue.js', 'vuejs2']	54451373	I am trying to create	Pagination with Vue JS	
18	['python-3.x', 'line	54451371	statsmodels.sta	How to find Variance Inflation fac	
19	['c#', 'r', 'linq', 'lin	54451369	I am performing a qu	What is the equivalent of R's	
20	['excel', 'blueprisr	54451366	I am trying to create	RPA BluePrism Excel VBO Extended	
21	['javascript', 'node	54451364	- I have 2 servers (A a	How to upload file between serve	
22	['pytorch', 'ray']	54451362	When I use the Ray v	How to use GPUs with Ray in Pytor	
23	['angular', 'jestjs']	54451361	I'm new to testi	Issue I don't understand abo	
24	['angularjs']	54451358	I am wondering why	Why Angular's ng-model valu	
25	['algorithm', 'enur	54451356	I'm using an en	Enum and Switch Statement to Ma	

Şekil 15. Ocak ayının ilk 25 verisi

2.7. Metin Önişleme Aşamaları ve Vektörel Dönüştürme

Metin önişleme aşaması metin madenciğinde analiz yapmadan önce uygulanan en önemli aşamalar öbeğidir. Metin önişleme adımları, metin odaklı veri analizlerinin başarımını doğrudan etkileyen bir süreçtir [25]. Metin önişleme süreci, özellikle yapısal olmayan (düzensiz) web tabanlı metinlerin ve sosyal ağlardan elde edilen metinsel içeriklerin

analizinde mutlaka uygulanması gereken bir işlemdir [28]. Metin önişleme süreci genel olarak, dizge parçalama, metnin temizlenmesi, uyumsuz ve tamamlanmamış metinlerin veri setinden çıkarılması, durak kelimelerinin silinmesi, gövdeleme (köke indirgeme), frekans indirgeme gibi sıralı işlemleri içermektedir. İşlenen verinin ve gerçekleştirilecek olan deneysel analizin türüne bağlı olarak veri önişleme adımları değişiklik gösterebilir [3,25,28].

Bu çalışmada sadece stack overflow üzerinden elde edilen Ocak 2019 ile Aralık 2019 tarihleri arasındaki 12 aylık süreçte yayınlanan ingilizce soru ve cevapları ile oluşturulan 1880141 adet soru ve bu sorulara verilen 1962196 adet cevaptan oluşan yazılım geliştirme odaklı soru-cevaplar her veri seti kendi ayının bilgisini içerecek şekilde 12 parçaya ayrılmıştır.

Bu bağlamda aşşağıdaki tabloda her ayın veri setinde kaç adet soru ve cevap bulunduğu verilmiştir.

Tablo 1. Aylık veri setindeki toplam soru ve cevap sayıları

Ay	Toplam Soru Sayısı	Toplam Cevap Sayısı
Ocak	145863	174436
Şubat	142660	171231
Mart	126059	148951
Nisan	150936	175490
Mayıs	167260	170833
Haziran	158960	156106
Temmuz	172216	167641
Ağustos	157297	156405
Eylül	156863	155004
Ekim	177053	171436
Kasım	170100	164389
Aralık	154874	150274
Toplam	1880141	1962196

Yapılan bütün işlemler örnek olması amacı ile sadece mart ayının yani 1 veri seti üzerinden örneklendirilecektir. Fakat bu işlemlerin bütün veri setleri üzerinde aynı şekilde yapıldığı bilinmelidir.

Örneğin mart ayı verisi üzerinde bulunan 126059 adet soru ve bu sorulara karşılık gelen 148951 kullanıcı yorumu ile mart ayı için veri seti oluşturulurken 126059 soruda ilgili sorunun question_id si ile eşleşen bütün cevaplar soruların bitiminde body_markdown kısmına eklenmiştir.

Veri önişleme aşamalarından ilk önce noktalama işaretleri, web bağlantıları, özel ve anlamsız etiketler, konu modellemeyi zorlaştıracak kod blokları, konu modellemede anlamı bozacak bazı kelimeler, kelimeler arası oluşmuş olan büyük boşluklar, konu modellemeyi bozacak şekilde bulunan bütün rakamlar metinlerden silinmiştir.


```
'I&#39;m trying to make a system similar to a like/dislike system. I have it all setup, but my jquery.ajax function is not send
ing the data I need it to. The function is being called, but the data is not being set.\r\n\r\nI&#39;ve added a function to che
ck for if the isset if statement is being called, but that function isn&#39;t being called, so I know it&#39;s something wrong
with the ajax function. I&#39;ve tried looking over other posts of similar problems, but none of those solutions worked. Sorry
if this is just me being stupid, and the solution is easy.\r\n\r\nPHP - community.php - The main webpage\r\n```\r\n&lt;div id=&
quot;entries&quot;&gt;\r\n    &lt;?php foreach($posts as $post): ?&gt;\r\n    &lt;div class=&quot;entry&quot;&gt;\r\n      &lt;
div class=&quot;entryInfo&quot;&gt;\r\n        &lt;img src=&quot;IMAGES/PlaceholderAlbumCover.jpg&quot; alt=&quot;Album Cover&q
uot; class=&quot;center&quot; height=&quot;200&quot; width=&quot;200&quot;&gt;\r\n        &lt;div class=&quot;artistInfo&quot;&
gt;\r\n          &lt;h4&gt;&lt;?php echo $post[&#39;name&#39;] ?&gt;&lt;/h4&gt;\r\n          &lt;p&gt; &lt;?php echo $post[&#3
9;artist&#39;] ?&gt; &lt;/p&gt;\r\n        &lt;/div&gt;\r\n        &lt;/div&gt;\r\n        &lt;div class=&quot;votes&quot;&gt;\r\n\r\n          &lt;i &
lt;?php if(userVoted($post[&#39;id&#39;])): ?&gt;\r\n            class=&quot;fas fa-heart vote-btn&quot;\r\n            &lt;?
php else: ?&gt;\r\n            class=&quot;far fa-heart vote-btn&quot;\r\n            &lt;?php endif ?&gt;\r\n            dat
a-id=&quot;&lt;?php echo $post[&#39;id&#39;] ?&gt;&quot; style=&quot;color:#f44242&quot;&gt;\r\n          &lt;/i&gt;\r\n\r\n
&lt;span class=&quot;voteCount&quot;&gt;&lt;?php echo getVotes($post[&#39;id&#39;]); ?&gt;&lt;/span&gt;\r\n        &lt;/div&g
t;\r\n      &lt;/div&gt;\r\n    &lt;/div&gt;\r\n    &lt;?php endforeach ?&gt;\r\n\r\n  &lt;/div&gt;\r\n```\r\n\r\nJavascript -
vote.js - Handles click detection and sending the response to vote.php\r\n```\r\n$.ajax({\r\n      url: &#39;index.php?page=com
munity&#39;,\r\n      method:&#39;post&#39;,\r\n      data: {\r\n        &#39;action&#39;:action,\r\n        &#39;post_id&#39;:
post_id\r\n      },\r\n      success: function(data) {\r\n        if(action == &#39;unvote&#39;) {\r\n          $click_btn.togg
leClass(&#39;far fas&#39;);\r\n        } else if(action == &#39;vote&#39;) {\r\n          $click_btn.toggleClass(&#39;far fas&#
39;);\r\n        }\r\n      }\r\n    });\r\n```\r\n\r\nPHP - vote.php - Handles the server side of the system\r\n```\r\nif(isse
t($_POST[&#39;action&#39;])) {\r\n    consoleLog();\r\n\r\n    $post_id = $_POST[&#39;id&#39;];\r\n    $action = $_POST[&#39;ac
tion&#39;];\r\n\r\n    switch($action) {\r\n      case &#39;vote&#39;:\r\n        $query=&quot;INSERT INTO rating_info (user_i
d, post_id, rating_action)\r\n                VALUES ($user_id, $post_id, &#39;vote&#39;)\r\n                ON DUPLICATE KEY U
PDATE rating_action=&#39;vote&#39;&quot;;\r\n        break;\r\n      case &#39;unvote&#39;:\r\n        $query=&quot;DELETE FROM
rating_info WHERE user_id=$user_id AND post_id=$post_id&quot;;\r\n        break;\r\n      default:\r\n        break;\r\n    }\r
\n\r\n    mysqli_query($dbconnect, $query);\r\n    echo getRating($post_id);\r\n    exit(0);\r\n  }\r\n```\r\n\r\nWhat it&#39;s
supposed to do is increment or decrement the vote count in the mysql database by setting the action to vote or unvote, but the
js file just never sets the action data, or the PHP script is never able to catch it. You are passing the post data in the java
script as `post_id` but trying to retrieve it in the php script as `$_POST[&#39;id&#39;]`\r\n\r\nDo this instead: `$_POST[&#39;
post_id&#39;]`'
```


Şekil 16. İlk aşama öncesi veri örneği


```
'I   m trying  make a system similar  a like dislike system  I have it all setup  but my jquery ajax  is not sending the data I
need it   The  is being called  but the data is not being set I   ve added a   check for if the isset if statement is being cal
led  but   isn   t being called  so I know it   s something wrong with the ajax   I   ve tried looking over other posts of simi
lar problems  but none of those solutions worked  Sorry if this is just me being stupid  and the solution is easy PHP   munity
php   The main webpageJavascript   vote js   Handles click detection and sending the response  vote phpPHP   vote php   Handles
the server side of the systemWhat it   s supposed  do is increment or decrement the vote count in the mysql database by setting
the action  vote or unvote  but the js  just never sets the action data  or the PHP script is never able  catch it  You are pas
sing the post data in the javascript as  but trying  retrieve it in the php script as Do this instead  '
```


Şekil 17. İlk aşama sonucu temizlenen veri

Şekil 16 ve 17 den de anlaşılacağı üzre ilk aşamanın tamamlanması sonucu kirli ve temiz veri arasındaki fark gözle görülür derecede fazladır.

Bir sonraki aşama olarak stemming(gövdeleme) işlemi gerçekleştirilmiştir. Bu aşamada kelimelerin üzerlerindeki ekleri kaldırıp metinlerin köklerinin elde edilmesi amaçlanır.Örnek olarak ("denied against community broadcasting licence flies stating dies died owned") şeklinde dizgi parçalama işlemi için gönderilen kelimelerden dönecek sonuç

olarak ['deny', 'against', 'community', 'broadcast', 'licence', 'fly', 'state', 'die', 'die', 'own'] şeklinde kelimelerin kökleri biçiminde olacaktır.

İkinci aşama ile elde edilen kelime uzayını küçültmek adına bir sonraki aşama olarak durak kelimeler (stop words) metin uzayından çıkartılmıştır. Durak kelimeleri bir dilde yaygın olarak kullanılan ve genellikle tek başına kullanıldığında bir anlam ifade etmeyen kelimelerdir. Bu nedenle metin analizine dayalı çalışmalarda genellikle durak kelimeleri metinlerden çıkarılır. Ancak bu kelimeler çıkarılırken her dile özgü, durak kelimelerini içeren bir listeye ihtiyaç duyulmaktadır. Durak kelimelerinin çıkarılması işleminin yapılan analiz türüne ve öznitelik çıkarım modeline bağlı olarak yapılabilirliği değişiklik göstermektedir. Bu çalışmada oluşturulan veri seti İngilizce metinlerden oluştuğu için, İngilizce dilinde yaygın kullanılan durak kelimeleri (and, or, with, there, she, with, are, the, vb.) metinlerden çıkarılmıştır [3,25,28].

Bir sonraki aşama olarak dizge parçalama (tokenization) işlemi yapılmıştır. Elde edilen metin içeriği kelimelere ayrılmıştır. Böylelik ile veri seti içerisinde bulunan her bir metinsel veri bir kelime uzayını temsil etmiş olur.

Elde edilen kelimeler ile bir sözlük oluşturulup bütün kelimeler bir sözlükte toplanmıştır. Örnek olarak 126059 soru ve 148951 cevap verisi içeren mart ayı içerisinde bahsedilen ön işleme aşamaları tamamlandıktan sonra 545579 adet kelime kökü bulunmaktadır.

Elimizdeki sözlükte bulunan 545579 kelime BoW işlemine tabii tutularak kelime uzayında hangi kelimenin ne kadar tekrar ettiği hesaplanarak frekansları hesaplanmış olur. Ayrıca birden fazla tekrar eden kelimeler tek bir kelime olarak varsayılarak frekansları doğrultusunda konu modellenirken etki katsayıları genel işleyise etki etmiş olur. Örnek vermek gerekir ise mart ayı için bulunan 126059 soru sayısı kadar külliyat oluşturulmuş olunup bu külliyatların her biri için ayrı ayrı BoW işlemi uygulanmıştır ve aşşağıda ki örnekler gibi her külliyat kendi içerisinde frekansları belli olacak şekilde sıraları önemsiz bir şekilde BoW içerisinde bulunmaktadır.

Word 306 ("simple") appears 1 time.
Word 307 ("solution") appears 2 time.
Word 315 ("through") appears 1 time.
Word 339 ("already") appears 1 time.

Word 340 ("always") appears 1 time.
Word 341 ("assign") appears 1 time.
Word 342 ("bethis") appears 1 time.
Word 343 ("create") appears 3 time.
Word 344 ("default") appears 1 time.
Word 345 ("depend") appears 1 time.
Word 346 ("dictionary") appears 4 time.

Metin önişleme aşamalarının bitmesi ile elde edilen BoW üzerinde yapılması gerekilen sıradaki işlem elimizdeki 126059 külliyat boyutunda GDT algoritmasinin gereksinim duydugu sayısal kelime vektörü temsilinin gerçekleştirilmesi işlemidir. Özelte metinler nitel veriler olduğu için sayısal analizin yapılabilmesi için metinlerin nicel verilere yani kelime vektörüne dönüştürülmesi gerekmektedir. Çünkü aksi halde nitel veriler üzerinde nicel analizlerin yapılması mümkün değildir. Veri setini modellemek için her dökümana ait olan terim verktörlerini bir araya getirerek doküman-terim matrisi oluşturulur[3,24,26,28].

2.8. Doküman-Terim Matrisi

Metin odaklı veri analizlerinde metin belgeleri, terimlerin (kelimelerin) vektörleri olarak temsil edilir ve bu belgelerin oluşturduğu veri seti de bir DTM ile temsil edilir [46]. Ön işleme aşamaları DTM oluşturulmadan önce yapılması gerekilen aşamalardır. DTM oluşturulma süreci, metin analizi işlemlerinin başlangıcı olarak kabul edilir. Böylece ön işleme adımları sonucu elde edilen külliyatlar ile bir DTM oluşturulur.

Oluşturulan DTM bir metin koleksiyonundaki terimlerin frekansını gösteren sayısal bir matristir. Bir DTM'deki her satır, veri setindeki bir dokümanı temsil eder ve her sütun ise dokümanlarda görülen benzersiz bir kelimeyi temsil eder [47]. Oluşan DTM de her hücre aslında satır tarafında belirtilen külliyatta ona karşılık gelen sütundaki kelimenin kaç kez tekrar ettiğini gösterir. Yani DTM boyutu satırlar için mart ayı verisi için 126059 adet külliyattan ve sütunlar için ise bu külliyatlar boyunca geçen kelimelerin eşsiz yani tekrarsız olarak geçen kelime sayısı olarak ifade edilebilir.

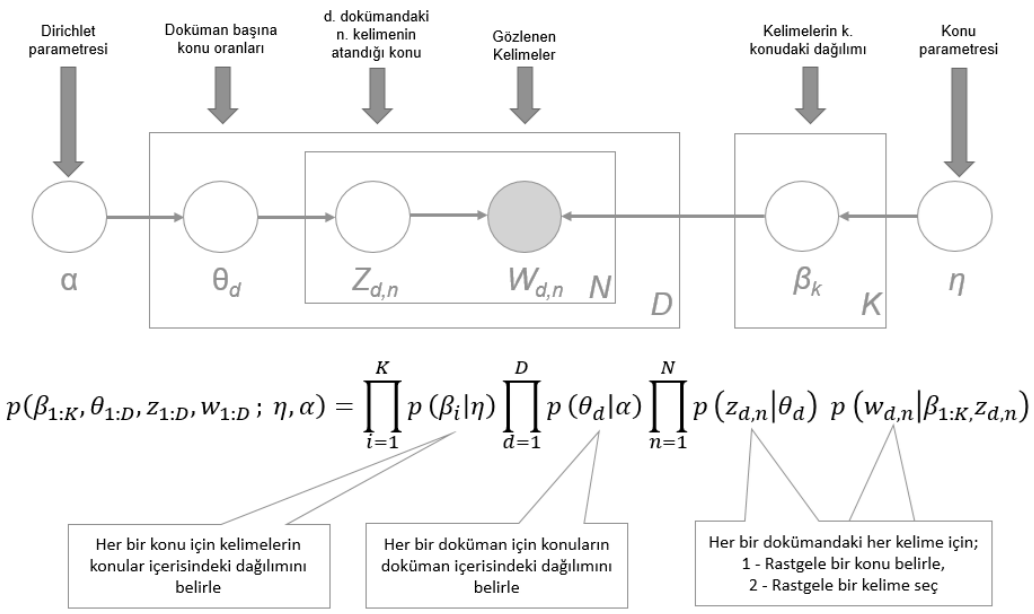
2.9. DTM-Tabanlı Konu Modelleme

Olasılıksal konu modelleme, büyük metin dokümanlarının anlamsal yapısını modellemek ve gizli anlamsal yapıları keşfetmek amacıyla kullanılan olasılıksal bir yaklaşımdır [48-50]. Metin dokümanları, konu (topic) olarak adlandırılan gizli anlamsal yapıları içerirler. Her konu sabit bir kelime kümesindeki olasılık dağılımı ile tanımlanır [49,50,51]. Olasılıksal konu modelleme yaklaşımına göre bir metin dokümanı farklı oranlara sahip birden fazla konuyu içerebilir. Bu konu oranlarının belirlenmesi olasılıksal konu modelleme yaklaşımının temelini oluşturur [49,50].

GDT konu modelleme yaklaşımı, denetimsiz öğrenmeye dayalı bir yöntem olup herhangi bir eğitim setine gerek duymadan büyük metin koleksiyonları üzerinde etkin olarak uygulanabilen yöntemdir [49]. Bu yöntem öğrenmeye dayalı bir yöntem olmadığından dolayı çok büyük metin uzayı içeren metinsel veri setleri ile işlem yapmak çok kısa sürelerde gerçekleştirilebilir.

GDT konu modelleme algoritması DTM üzerinde gerçekleştirilen bir takım matris işleminden sonra anlamsal ilişkileri ortaya çıkarmaktadır. Bu işlemin sonucunda yüksek oranda birliktelik içeren kelimeler gruplanır ve bu kelime öbeklerinin her biri “konu”(topic) olarak adlandırılır.

Bu bağlamda stack overflow ingilizce soru ve cevaplar üzerinde konu modelleme yapılırken konu sayısı olan K değişkeninin belirlenmesi çok önemlidir. K değeri bulunurken bu değerin tanımlanması tamamen araştırmacının bakış açısı ile ilgilidir ve araştırmacının daha önceden yaptığı denemeler ile belirlenebilir. Bu bağlamda konu sayısını 10 ile 20 arasında belirlenip her seferinde konu sayısını 2 arttırarak denemeler yapıldı ve bu denemeler her ay için ayrı ayrı şekilde uygulandı.



Şekil 18. GDT-tabanlı konu modelleme için akış şeması[8]

Şekil ’de gösterilen ve olasılık dağılımında kullanılan tüm parametreler ve açıklamaları Mart ayı veri seti örneği için Tablo 6’da verilmiştir

Tablo 2. GDT için önerilen model parametreleri, açıklamaları ve değerleri

Parametre	Açıklaması
D	Toplam doküman sayısı (D=126059)
K	Toplam gizli konu sayısı (K=12)
V	Sözlükte bulunan toplam kelime sayısı
Nd	d. dokümandaki kelime sayısı
α	Doküman başına düşen konu dağılımları için Dirichlet parametresi
η	Konu başına düşen kelime dağılımları için Dirichlet parametresi
θd	Konuların d. dokümandaki dağılımı
βk	Kelimelerin k. konudaki dağılımı
zd,n	d. dokümandaki n. konumda bulunan kelimenin atandığı konu
wd,n	d. dokümandaki n. konumda gözlemlenen kelime

3. BULGULAR

Bu tez kapsamında Yazılım geliştiricilerin tartıştıkları konulara yönelik stack overflow ingilizce kullanıcı soru ve cevaplarından oluşturulan veri seti üzerinde GDT-tabanlı olasılıksal konu modellemeye dayanan deneysel bir çalışma gerçekleştirilmiştir. Deneyin sonunda 2019 yılının 12 ayı, aylık olarak hangi konuların tartışıldığı yapılan konu keşfi sonunda tespit edilmeye çalışılmıştır.

3.1. Ocak Ayı Bulguları

Tablo 3. Ocak ayı bulguları

Konu	**Başlıklar**
E-ticaret sitesi geliştirme araçları	kivy,stripe,grails,tinymce,jekyll,kubelet,hbase,soup
Share point uygulamaları	sheet,worksheet,cells,sharepo,lastrow,workbook
Qt framework	qtwidgets,qtgui,qtcore,jupyter,pyqt,grpc
Python derin öğrenme	keras,plot,pandas,matplotlib,shape,layer,tensorflow,train,model
Java	java,springframework,maven,dependency,spring,boot
Android	android,layout_,wrap_content,androidx,schemas,ansible
Asistan,yönetim	ffmpeg,airflow,skill,linkedin,alexa,keycloak
Dijital ses işlemleri	audio,carousel,electron,sequelize,broker,producer,sftp
Veri işlemleri	data,list,test,request,form,script,object,create

3.2. Şubat Ayı Bulguları

Tablo 4. Şubat ayı bulguları

Konu	**Başlıklar**
Altyapı,sunucu	docker,jboss,terraform,xampp,mongo,pagination
Veri işlemleri	data,list,test,time,object,create,request,form
Api tabanlı frameworkler	nativescript,scrapy,nuxt,solr,asyncio,twilio,geolocation
Android	android,layout_,match_parent,wrap_content,linearlayout
HTML	background,border,margin,font,href,position
Python derin öğrenme	keras,activation,relu,sigmoid,dense,
Anlık döküman işlemleri	segue,fullcalendar,newdata,netty,jspdf,docusign
Veri yönetimi	graphql,flink,odoo,asyncio,mapper,ckedir

3.3. Mart Ayı Bulguları

Tablo 5. Mart ayı bulguları

Konu	**Başlıklar**
Python derin öğrenme	keras,activation,predict,layer,dense,sklearn,model,train,gulp,loss
Android	android,layout_,view,intent,recyclerview,match_parent, wrap_content,override
Java	browsersync,java,springframework,proxied,jdbc,spring, dependency,boot
Qt framework	qstring,qtcore,jupyter,pyqt
Web tasarım	navbar,gitlab,href,bootstrap,metric,dropdown
Veri işlemleri	data,response,list,result,call,object,script,test
Tweeter developer api	tweet,tweepy,elem,writerow,country
Mobil uygulama geliştirme	android,gradle,suite,flutter,java

3.4. Nisan Ayı Bulguları

Tablo 6. Nisan ayı bulguları

Konu	**Başlıklar**
Veri işlemleri	data,list,time,object,result
Android	android,layout_,intent,recyclerview,match_parent,wrap_content, textview
Platformlar arası sistemler	drupal,spyder,snmp,convolutional,cpanel
Ios	ionic,indexpath,mocki,saml
Bulut, sanal sistemler	server,request,java,service,docker,react,azure
E-ticaret para işlemleri	woomerce,captcha,loan,recaptcha
Python derin öğrenme	Keras,matplotlib,train,relu,test
Mobil platform test	appium,matcher,parenthesis,viewer,resttemplate,watcher

3.5. Mayıs Ayı Bulguları

Tablo 7. Mayıs ayı bulguları

Konu	**Başlıklar**
Frameworks	indexpath,cocoapods,tomcat,ember,sonar,coalesce

Tasarım	style,snippet,span,lang,background,hide,border,font
Paket sorgulama ve yönetimi	graphql,conda,scrapy,datagridview,charfield,queryset
Java	java,jdbc,springframework,swagger,junit
Android	android,layout_,intent,view,override,wrap_content,match_parent
Veri işlemleri	data,list,test,create,object
Python derin öğrenme	keras,sklearn,tweet,activation,dense,peer,x_train,pyspark
Excel	sheet,worksheet,cells,workbooks

3.6. Haziran Ayı Bulguları

Tablo 8. Haziran ayı bulguları

Konu	**Başlıklar**
GUI	gui,pyautogui,indexpath,svelte,sftp,lista
Entegrasyon	audiomanager,istio,modelandview,isbn,formik,palindrome,rcpp
Assembly	nifi,findelement,llvm,traefik,sitemap,aapt,wasm,gradlew
JavaScript	string,java,request,response,json,data
Android	android,layout_,wrap_content,match_parent,recyclerview, linearlayout,drawable
Text generator	dolor,discount,ipsum,amet,lorem,navbar,gsub
Data stream	minikube,reaction,pyodbc,edgeinsets,kinesis,quaternion, startinfo,kubernetes
HTML tools	twilio,model,lstm,jspdf,blazor,jquery,cloudformation
Python derin öğrenme	keras,x_train,activation,layer,dense,colab,x_test,model

3.7. Temmuz Ayı Bulguları

Tablo 9. Temmuz ayı bulguları

Konu	**Başlıklar**
Python derin öğrenme	tensorflow,train,model,keras,scipy,layer,tensor,octal, conda,carousel,dense
Android studio	data,test,android,project,server,java,build
Python destekli frameworkler	kendo,subroutine,odoo,librosa,serverless,resttemplate
Veri işlemleri	data,list,column,number,result,item,query
Apache Http Server	rewritecond,rewriterule,rewriteengine,ssrs,weblogic,request_uri,

	http_host
Android	android,layout_,androidx,fragment,match_parent,view,java, recyclerview,intent
Veri görselleştirme	hyperlink,obfuscate,ggplot,eventsource,velocity,geom_po, amcharts,dateadd,tabular
Kaynak paylaşımı	cors,firstname,postback,swagger,gpio,lastname,qdebug,tenant,claim,httpclient
Amazon servisleri	phpmailer,post_,bazel,newvalue,post_status,cloudwatch, timepicker

3.8. Ağustos Ayı Bulguları

Tablo 10. Ağustos ayı bulguları

Konu	**Başlıklar**
Excel	sheet,worksheets,cells,range,slack,workbook,worksheet
Veri işlemleri	data,result,list,test,create,response
Java	springframework,spring,bean,maven,java,dependency,boot
Tasarım	style,background,snippet,font,label,lang,center,border
Android	android,recyclerview,match_parent,wrap_content,view,layout, textview
Flutter kütüphaneleri	buildcon,onpressed,appbar,cosmos,edgeinsets,initstate,xmlhttp
Sunucu hizmetleri	spark,jenkins,kafka,server,request,devops,login,azure,schedule
Java api	keycloak,sudo,webpack,ldap,java,sass,istio
Python veri ayıklama	soup,scrapy,airflow,gender,timezone,beautifulsoup,survey,male
JavaScript	xaml,geojson,sonar,datatemplate,mutex,theta,jspdf,portfolio

3.9. Eylül Ayı Bulguları

Tablo 11. Eylül ayı bulguları

Konu	**Başlıklar**
Python dataframe	plot,dataframe,pandas,matrix,data,column,model,train,range
Android java	android,java,recyclerview,string,override,view,adapter,androidx ,javafx,void
Veri işlemleri	data,request,create,object,test

Kubernetes	cypress,appium,twilio,helm,coroutines,assemblies,dask, publisher,kubectl
Flutter	drawer,leak,edgeinsets,slave,envs,sftp,initstate,fingerpr
Ruby Gems	polygon,trait,loopback,grafana,alarmmanager,ldap,gems,poly, generics
Android	wrap_content,android,layout_,recyclerview,view,match_parent, override,java,layout
Apache java	log4j,prevstate,pageable,zookeeper,scrollviewer
JavaScript	gatsby,grpc,survey,gsub,spacy,testcafe,formik

3.10. Ekim Ayı Bulguları

Tablo 12. Ekim ayı bulguları

Konu	**Başlıklar**
Yapı otomasyonları	ansible,elastic,terraform,poser,jackson,maven,stack
Sunucu tabanlı veri işlemleri	server,request,project,service,application,data
Web geliştirme	groovy,hadoop,popover,crossorigin,combobox,istio,htdocs,bulma
Python derin öğrenme	dense,reshape,keras,x_train,strptime,activation,y_pred,spacy
Apache	rewriterule,rewritecond,htaccess,rewriteengine,dask
Alternatif geliştirme platformları	cucumber,mingw,nlog,notepad,resttemplate,webrtc,coroutines
Android UI	style,android,snippet,background,item,lang,span,hide,language
Veri işlemleri	data,list,number,column,result,output,query
Python kütüphaneleri	pyspark,hive,spark,timedelta,probuf,prer,openpyxl,cogni,swiper
Veri görselleştirme	jmeter,slack,activitythread,histogram,highcharts
Swift	swiftui,polygon,logout,twilio,openssl,solr,voice

3.11. Kasım Ayı Bulguları

Tablo 13. Kasım ayı bulguları

Konu	**Başlıklar**
Veri	scipy,pandas,snowflake,sensor,mydata,dataframe,dplyr,celery
Sunucu tabanlı veri işlemleri	test,data,request,use,server,create,service
Çoklu geliştirme platformları	elasticsearch,graphql,elastic,endpos,serverless,fabric,logstash, blazor

JavaSript modülleri	odoo,saml,cloudfront,submodule,karma,protracr,convd,athena
Android	android,view,layout_,override,recyclerview,intent
Python veri bilimi	x_test,pandas,activity,train,model
Yardımcı programlar	cron,swagger,bigquery,hadoop,nativescript,scriptlang,clipboard, cassandra,appium
Django	scrapy,tenant,migration,patient,phpmyadmin,subnet,urllib

3.12. Aralık Ayı Bulguları

Tablo 14. Aralık ayı bulguları

Konu	**Başlıklar**
Swift	indexpath,collectionview,resttemplate,uitableview,iris,beam, banner,weblogic
Python veri bilimi	pandas,column,dataframe,sheet,data,number,date,range
HTML	style,background,font,border,margin,flex,center,href,span
Web programlama	websocket,cordova,webdriver,selenium,driver,keycloak,mcat, highcharts,groovy,neo4j
Veri işlemleri	data,list,create,request,result
Android	android,layout_,intent,fragment,view,override,recyclerview, activity,androidx
Uygulama içi haberleşme ve ödeme	stripe,cart,nativescript,spotify,mqtt,virtualenv,beanstalk,county
JavaScript kütüphaneleri	cosmos,gatsby,kendo,median,akka

4. ÇIKARIMLAR

4.1. Veri Bilimi

Yapılandırılmış ve yapılandırılmamış verilerden bilgi elde etmek için bilimsel yöntem, süreç, algoritmalar ve süreçleri kullanan bir veri madenciliği ve büyük verilerle ilişkili bir alandır. Veri bilimi günümüzde yazılımın gelişmesi ile birlikte neredeyse hemen hemen her alana yayılmış, bir veya birden fazla insandan oluşan grupların zamanı ve kayrayışının kısıtlı olmasından dolayı bu işlemleri bilgisayar sistemleri gibi cansız sistemlere yaptırmak bize gerçek anlamda büyük zamanlar kazandırmaktadır. Günümüzde çok yaygın bir şekilde ilerleyen veri bilimi ile Python birbirlerini tetikleyerek gelişmektedir.

4.2. Mobil Uygulama

Mobil cihazlara(akıllı telefon ve tablet) yönelik özel kodlar ve tasarımlar ile yapılan yazılmlara mobil uygulamalar denir. Mobil uygulamalar dünya çapında IOS ve Android işletim sistemi gibi çok bilinen işletim sistemlerinin etrafında dönmektedir. Farklı işletim sistemleri için kullanılacak farklı geliştirme dilleri, platformlar ve frameworkler mevcuttur.

Mobil uygulamalar gelişen yazılım ile birlikte web sitelerine göre oldukça hızlı olduğu için daha çok tercih edilmektedir. Mobil uygulamaları sadece web sitelerinin mobil versiyonu olarak düşünmemeliyiz. Birçok otomasyon,e-ticaret uygulaması,yapay zeka tabanlı uygulamalar, oyun, kişisel gelişim, günlük gibi kişisel kullanıma yönelik uygulamalarda mevcuttur.

Yapılan çalışma sonucu elde edilen bulgulara göre, özellikle veri, veri yönetimi ve yapay zeka alanlarına büyük şekilde yönlenme olduğunu açık bir şekilde göstermektedir. Ayrıca artık günümüzde lüksten çok ihtiyaç haline gelmiş akıllı telefonların sayısının çok olması ile doğru orantılı olarak android platforma yapılan yatırımında o denli fazla olduğu bulgular sonucu çok aşikardır. Veri bilimi son yıllarda Python ile birlikte çok büyük atılımlar yapmış, özellikle python da kullanılan açık kaynaklı veri bilimi kütüphaneleri yardımı ile yapay zeka ve derin öğrenme alanlarında çalışmak çok basit hal almıştır. Aynı bağlamda Java programlama dili bir çok platformda gerek client tarafında gerekse android uygulama tabanında kullanıldığı için popüler konumdadır. Ayrıca elde edilen bulgulara göre bir diğer vazgeçilmez alan sunucular ve bu sunucuların sunmuş oldukları hizmetlerdir.

KAYNAKÇA

1. YENİÇERİ Ö., DEMİREL Y., Örgüt içi bilgi paylaşımına yönelik bireysel ve örgütsel engeller üzerine bir araştırma, Haziran 2007, Selçuk Üniversitesi Karaman İ.İ.B.F. Dergisi
2. AKKOYUNLU B., Öğretmenlerin internet kullanımı ve bu konudaki öğretmen görüşleri, Haziran 2002, Hacettepe Üniversitesi Eğitim Fakültesi Dergisi 22
3. Gurcan, F. ve Kose, C., Analysis of Software Engineering Industry Needs and Trends: Implications for Education, International Journal of Engineering Education, 33,4 (2017) 1361-1368.
4. Akman, G. ve Yilmaz, C., Innovative capability, innovation strategy and market orientation: an empirical analysis in Turkish software industry, International Journal of Innovation Management, 12,01 (2008) 69-111.
5. Harter, D. E., Krishnan, M. S. ve Slaughter, S. A., Effects of process maturity on quality, cycle time, and effort in software product development, Management Science, 46,4 (2000) 451-466.
6. Moreno, A. M., Sanchez-Segura, M. I., Medina-Dominguez, F. ve Carvajal, L., Balancing software engineering education and industrial needs, Journal of systems and software, 85,7 (2012) 1607-1620.
7. Lee, D. M., Trauth, E. M. ve Farwell, D., Critical skills and knowledge requirements of IS professionals: a joint academic/industry investigation, MIS quarterly, (1995) 313-340.
8. Gurcan F., Yeni nesil yazılım geliştirme eğilimlerine yönelik uzman bilgi ve becerilerin olasılıksal konu modelleme yordamıyla belirlenmesi, Kasım 2017, Doktora Tezi
9. Artificial Intelligence, Britannica.com, Şubat 2015

10. Özyurt Ö., Türkçe tabanlı diyalog sistemi tasarımı ve internet (chat) ortamlarından bilgi çıkarımı, Temmuz 2006, Yüksek Lisans Tezi

11. Turban, E., Expert Systems and Applied Artificial Itelligence, Prentice Hall Inc., New Jersey, 1992.

12. Russel. S. Ve Norvig. P., Artificial Intelligence, A Modern Approach, Prentice Hall, 2nd Edition,2003

13. Veri Madenciliği, https://tr.wikipedia.org/wiki/Veri_madencili%C4%9Fi

14. https://medium.com/@furkanalaybeg/veri-madencili%C4%9Fi-ve-y%C3%B6ntemleri-d0e2fd238e44

15. Sadi Evren SEKER, Cihan Mert, Khaled Al-Naami, Nuri Ozalp, Ugur Ayan (2013), Correlation between the Economy News and Stock Market in Turkey., International Journal of Business Intelligence and Review (IJBIR), vol. 4, is. 4, pp. 1-21, 2013

16. Şadi Evren ŞEKER, "Turkish Query Engine on Library Ontology", IKE12, Internet Knowledge Engineering, 2012, ISBN:1-60132-222-4, Pages:26-33

17. Sadi Evren SEKER, Banu DIRI, International Conference on Artificial Intelligence konferansı dahilinde , "Proceedings of International Conference on Artificial Intelligence", bildiri "TimeML and Turkish Temporal Logic", pp. 881-887, ICAI 2010

18. Nabiyev, V., Vasif., Yapay Zeka, Seçkin Yayıncılık, Ankara,2003

19. Allen, J., Natural Language Understanding, The Benjamin/Cummings Publishing Inc., Redwood City, California, 1995

20. Liu, B. Sentiment Analysis and Opinion Mining Synthesis Lectures on Human Language Technologies, Editör: Hirst, G. Morgan & Claypool, 2012.

21. Sadi Evren SEKER, Khaled Al-NAAMI "Sentimental Analysis on Turkish Blogs via Ensemble Classifier", PROCEEDINGS OF THE 2013 INTERNATIONAL CONFERENCE ON DATA MINING, ISBN:1- 60132-239-9, DMIN, pp. 10-16, 2013

22. Türkmen, H., İlhan Omurca, S., Ekinci, E. An Aspect Based Sentiment Analysis on Turkish Hotel Reviews, Girne American University Journal of Social and Applied Sciences, 6, 2016, pp. 9-15.

23. Weiss, S. M., Indurkhya, N., Zhang, T., ve Damerau, F., Text mining: predictive methods for analyzing unstructured information, Springer Science & Business Media, 2010.

24. Feldman, R., ve Sanger, J., The text mining handbook: advanced approaches in analyzing unstructured data, Cambridge university press, 2007.

25. Vijayarani, S., Ilamathi, M. J., ve Nithya, M., Preprocessing techniques for text mining-an overview, International Journal of Computer Science & Communication Networks, 5,1 (2015) 7-16.

26. Figueiredo, F., Rocha, L., Couto, T., Salles, T., Gonçalves, M. A., ve Meira Jr, W., Word co-occurrence features for text classification, Information Systems, 36,5 (2011) 843-858.

27. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A. ve Chanona-Hernández, L., Syntactic n-grams as machine learning features for natural language processing, Expert Systems with Applications, 41,3 (2014) 853-860.

28. Çoban Ö., Metin Sınıflandırma Teknikleri ile Türkçe Twitter Duygu Analizi, Yüksek Lisans Tezi, Atatürk Üniversitesi, Fen Bilimleri Enstitüsü, Erzurum, 2016.

29. Luhn, Hans Peter (1957). "A Statistical Approach to Mechanized Encoding and Searching of Literary Information". IBM Journal of Research and Development. 1 (4): 309–317. doi:10.1147/rd.14.0309. Retrieved 2 March 2015.
30. Manning, C.D.; Raghavan, P.; Schutze, H. (2008). "Scoring, term weighting, and the vector space model". Introduction to Information Retrieval. p. 100. doi:10.1017/CBO9780511809071.007. ISBN 978-0-511-80907-1.
31. Term Frequency- Inverse Document Frequency statistics, https://jmotif.github.io/sax-vsm_site/morea/algorithm/TFIDF.html
32. Spärck Jones, K. (1972). "A Statistical Interpretation of Term Specificity and Its Application in Retrieval". Journal of Documentation. 28: 11–21. CiteSeerX 10.1.1.115.8343. doi:10.1108/eb026526.
33. Susan T. Dumais (2005). "Latent Semantic Analysis". Annual Review of Information Science and Technology. 38: 188–230. doi:10.1002/aris.1440380105.
34. Salakhutdinov, Ruslan, and Geoffrey Hinton. "Semantic hashing." RBM 500.3 (2007): 500.
35. Deerwester, S., et al, Improving Information Retrieval with Latent Semantic Indexing, Proceedings of the 51st Annual Meeting of the American Society for Information Science 25, 1988, pp. 36–40.
36. Deerwester, Scott; Dumais, Susan T.; Furnas, George W.; Landauer, Thomas K.; Harshman, Richard (1990). "Indexing by latent semantic analysis". Journal of the American Society for Information Science. 41 (6): 391–407. CiteSeerX 10.1.1.108.8490. doi:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.
37. Furnas, G. W.; Landauer, T. K.; Gomez, L. M.; Dumais, S. T. (1987). "The vocabulary problem in human-system communication". Communications of the ACM. 30 (11): 964–971. CiteSeerX 10.1.1.118.4768. doi:10.1145/32206.32212
38. Landauer, T., et al., Learning Human-like Knowledge by Singular Value Decomposition: A Progress Report, M. I. Jordan, M. J. Kearns & S. A. Solla (Eds.), Advances in Neural Information Processing Systems 10, Cambridge: MIT Press, 1998, pp. 45–51.

39. Dumais, S.; Platt, J.; Heckerman, D.; Sahami, M. (1998). "Inductive learning algorithms and representations for text categorization". Proceedings of the seventh international conference on Information and knowledge management - CIKM '98. pp. 148. CiteSeerX 10.1.1.80.8909. doi:10.1145/288627.288651. ISBN 978-1581130614.

40. Mei, Q., Shen, X., Zhai, C. Automatic Labeling of Multinomial Topic Models, In Proceedings of ACM KDD, 2007, pp. 490-499.

41. Phan, X-H., Nguyen, C-T., Le, D-T., Nguyen, L-M., Horiguchi, S., Ha, Q-T. A Hidden Topic-Based Framework toward Building Applications with Short Web Documents, IEEE Transactions on Knowledge and Data Engineering, 23(7), 2011, pp. 961-976.

42. Blei, David M.; Ng, Andrew Y.; Jordan, Michael I (January 2003). Lafferty, John (ed.). "Latent Dirichlet Allocation". Journal of Machine Learning Research. 3 (4–5): pp. 993–1022. doi:10.1162/jmlr.2003.3.4-5.993. Archived from the original on 2012-05-01. Retrieved 2006-12-19.

43. Latent semantic analysis, https://en.wikipedia.org/wiki/Latent_semantic_analysis

44. "Lessons learned: Using Scrum in non-technical teams". *Agile Alliance*. Retrieved April 8, 2019.

45. Schwaber, Ken; Sutherland, Jeff (November 2017), *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game* , retrieved May 13, 2020

46. Lu, B., Ott, M., Cardie, C. ve Tsou, B. K., Multi-aspect sentiment analysis with topic models, In Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Aralık 2011, Vancouver, IEEE, 81-88.

47. Hong, L. ve Davison, B. D., Empirical study of topic modeling in twitter, In Proceedings of the first workshop on social media analytics, Temmuz 2010, Washington, ACM, 80-88.

48. Blei, D. M., & Lafferty, J. D., Topic models, Text mining: classification, clustering, and applications, 10,71 (2009) 34.

49. Blei, D. M., Probabilistic topic models, Communications of the ACM, 55,4 (2012) 77-84.

50. Steyvers, M. ve Griffiths, T., Probabilistic topic models, Handbook of latent semantic analysis, 427,7 (2007) 424-440.

51. Blei, D. M., Ng, A. Y. ve Jordan, M. I., Latent dirichlet allocation, Journal of machine Learning research, 3 (2003) 993-1022.

ÖZGEÇMİŞ

Ad Soyad	**Olcay ÇİFTÇİ**
Tel no	**+90 553 185 55 09**
E-posta	**olcaycft95@gmail.com**
Adres	**2712 sok. no:24 ulubatlı mahallesi. Toros/Konak İZMİR**

Özgeçmiş Bilgileri :

1995 Kars Karakaş köyü kütüklü, azeri kökenli İzmir Konak doğumlu olan Olcay; İlköğretim eğitimini 4 yıl Şehit teğmen murat arslan türk ilköğretim okulu ve 4 yıl mustafa kemal ilköğretim okulunda olmak üzere bitirdi. Daha sonra İzmir Esnaf sanatkarlar odaları birliği anadolu teknik, anadolu meslek ve endüstri meslek lisesinde: Anadolu teknik web tasarım ve programlama bölümünden mezun oldu. Lise yıllarında programlama ile tanışan Olcay tübitak yarışmalarına microsoft' un ses tanıma çalışmaları ile ilgili bazı projeler ile katıldı ve aynı zamanda teknik servis elemanı olarak çalıştı. Liseden mezun olduktan sonra 1 yıl eğitime ara verip sınavlara hazırlandı ve Karadeniz Teknik Üniversitesi Yazılım Mühendisliği bölümünü kazandı. Üniversite eğitimi sırasında Image Yazılım-İzmir, İnvenoa-İstanbul, Binovist-İstanbul gibi çeşitli yerlerde çeşitli görevler alarak staj yaptı ve aynı zamanda üniversite eğitimi sırasında Universidad Castilla la mancha-Ciudad Real-İspanya' da Erasmus programına katılmaya hak kazandı ve bir dönem eğitim aldı.