Yazılım Geliştiricilerin Tartıştıkları Konulara Yönelik Kullanıcı Soru ve Cevaplarının Olasılıksal Konu Modelleme Yöntemi İle Belirlenmesi

Olcay ÇİFTÇİ

Yazılım Mühendisliği Bölümü
Karadeniz Teknik Üniversitesi, Trabzon
olcaycft95@gmail.com

ÖZET

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.

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

SUMMARY

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

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.

2. 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[9].

2.1. 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[10]. 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[11]

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 [10,12]. "Personel çok çalışkandı." yorumunda "personel" kelimesi duygu analizindeki özelliğe karşılık gelmektedir.

2.2. 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 [3,13-15].

2.2.1. 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:[16]

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 1. 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 [17]
- 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); [18]

2.2.2. 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: [19] Bir terimin özgüllüğü, meydana geldiği belge sayısının ters bir fonksiyonu olarak ölçülebilir.

Ters Belge Sıklığı(idf) Ağırlığının Değişkenl

Ağırlıklandırma Şeması	idf ağırlığı ($n_t = \|\{d \in D$
tekli (birli)	1
Ters belge sıklığı	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
Ters belge sıklığı smoothi	$\log\left(\frac{N}{1+n_t}\right)+1$
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 2. 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):

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

Şekil 3. Idf formulü

Burada;

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

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

tf – idf şu şekilde hesaplanır:

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

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 4. Önerilen Tf-Idf Ağırlıklandırma Şeması

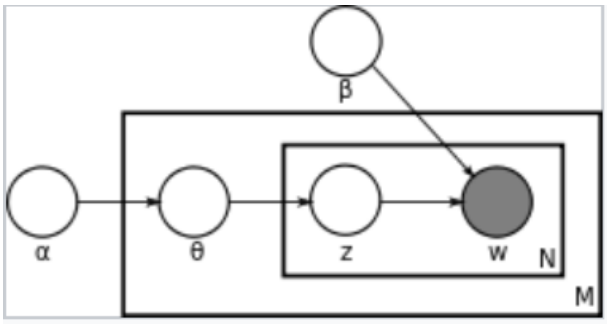
3. Gizli Anlamsal Analiz

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[20].

3.1. Gizli Dirichlet Tahsisi

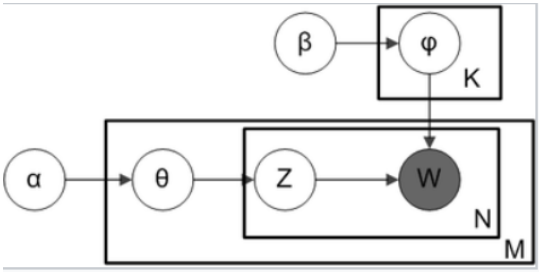
Doğal dil işlemede, gizli Dirichlet tahsisi (GDT), verilerin bazı bölümlerinin neden benzer olduğunu açıklayan gözlemsiz gruplar tarafından gözlem kümelerinin açıklanmasına izin veren üretken bir istatistiksel modeldir. Örneğin, gözlemler belgelere toplanan kelimeler ise, her belgenin az sayıda konunun 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 [21,22]. Mühendislikte GDT 'nin bir örneği, belgeleri otomatik olarak sınıflandırmak ve çeşitli konularla ilişkilerini tahmin etmektir.



Şekil 5. 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 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 6. Plaka gösterimi 2

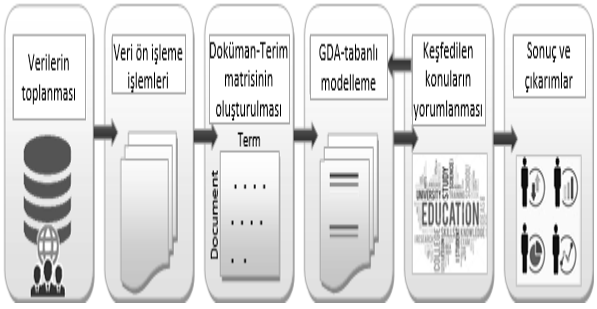
W'nin grileşmesi, w_{ij} kelimelerinin gözlemlenebilir tek değişken olduğu ve diğer değişkenlerin gizli değişken olduğu anlamına gelir. Bir konudaki kelimeler üzerindeki olasılık dağılımının çarpık olduğu sezgisini takiben, konu-kelime dağılımını modellemek için daha önce seyrek bir Dirichlet kullanılabilir, 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 , Dirichlet tarafından dağıtılan konu-kelime dağılımlarının parametrelerini saklayan V-boyutlu vektörlerdir (V, kelime dağarcığı)[23].

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

Şekil 7. GDT algoritması istatistiksel formulü

4. 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 8. 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.

4.1. Verilerin Elde Edilmesi

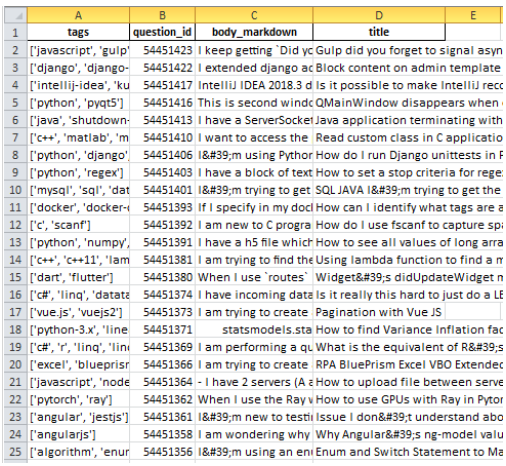
Yazılım geliştiriciler teknolojinin hızlı geliştiği ve yazılım geliştirmenin giderek karmaşıklaştığı bu dönemde, yine teknolojinin hızlı gelişmesi sayesinde bilginin dünya çapında internet üzerindeki hızlı yayılımı kapsamında bu karşılaşılan karmaşıklığın çözümü olarak interneti ve internet üzerindeki 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.

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 F	
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 doc	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 n	
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', 'blueprism	54451366	I am trying to create	RPA BluePrism Excel VBO Extendec	
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 9. Ocak ayının ilk 25 verisi

4.2. 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 [24]. Metin önişleme süreci, özellikle yapısal olmayan (düzensiz) web tabanlı metinlerin ve sosyal ağlardan elde edilen metinsel içeriklerin analizinde mutlaka uygulanması gereken bir işlemdir [15]. 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 [3,15,24].

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

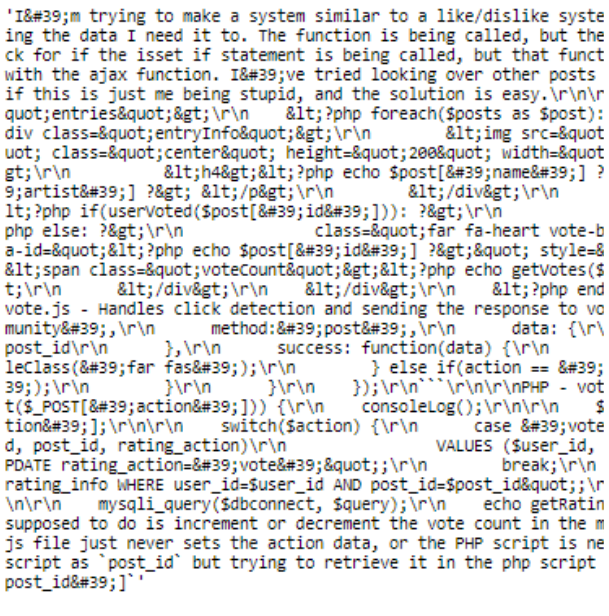
Ay	Toplam Soru Sayısı	Toplam Cevap Sayısı
Ocak	145863	174436
Şubat	142660	171231
Mart	126059	148951
Nisan	150936	175490
Mayıs	167260	170833
Haziran	158960	156106
Temmuz	172216	167641
Ağustos	157297	156405
Eylül	156863	155004
Ekim	177053	171436
Kasım	170100	164389
Aralık	154874	150274
Toplam	1880141	1962196

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.

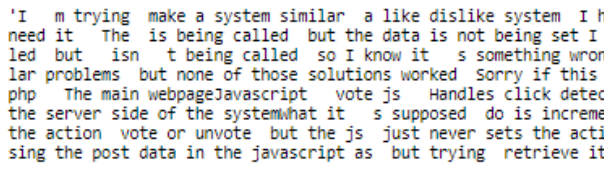
Örneğin mart ayı verisi üzerinde bulunan 126059 adet soru ve bu sorulara karşılık gelen 148951 kullanıcı yorumu ile mart ayı için veri seti oluşturulurken 126059 soruda ilgili sorunun question_id si ile

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.



Şekil 10. İlk aşama öncesi veri örneği



Şekil 11. İlk aşama sonucu temizlenen veri

Şekil 10 ve 11 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.

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. İngilizce dilinde yaygın kullanılan durak kelimeleri (and, or, with, there, she, with, are, the, vb.) metinlerden çıkarılmıştır [3,15,24].

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.

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. Veri setini modellemek için her dökümana ait olan terim verktörlerini bir araya getirerek doküman-terim matrisi oluşturulur[3,13,15,25].

4.3. DTM-Tabanlı Konu Modelleme

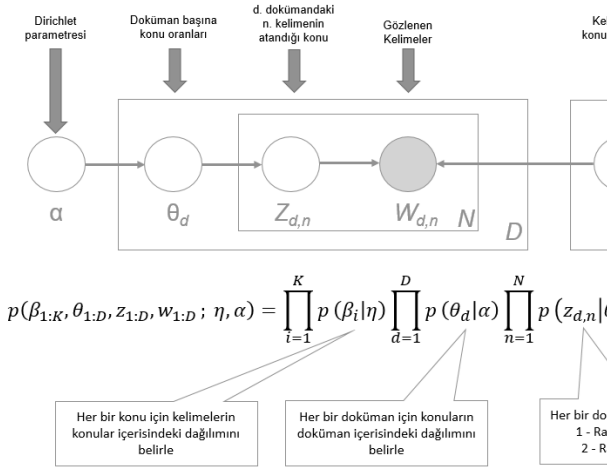
Olasılıksal konu modelleme, büyük metin dokümanlarının anlamsal yapısını modellemek ve gizli anlamsal yapıları keşfetmek amacıyla kullanılan olasılıksal bir yaklaşımdır [26-28]. Metin dokümanları, konu (topic) olarak adlandırılan gizli anlamsal yapıları içerirler. Her konu sabit bir kelime kümesindeki olasılık dağılımı ile tanımlanır [27,28,29]. 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 [27,28].

GDT konu modelleme yaklaşımı, denetimsiz öğrenmeye dayalı bir yöntem olup herhangi bir eğitim setine gerek duymadan büyük metin koleksiyonları üzerinde etkin olarak uygulanabilen yöntemdir [27]. 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 12. 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

5. 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.

5.1. Mart Ayı Bulguları

Tablo 3. 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

5.2. Nisan Ayı Bulguları

Tablo 4. 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	Python derin öğrenme
Python derin öğrenme	appium,matcher,parenthesis, viewer,resttemplate,watcher

6. ÇIKARIMLAR

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.

7. 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. 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

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

11. 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

12. 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.

13. 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.

14. 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.

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

16. 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.

17. 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.

18. Term Frequency- Inverse Document Frequency statistics, https://jmotif.github.io/sax-vsm_site/morea/algorithm/TFIDF.html

19. 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.

20. Susan T. Dumais (2005). "Latent Semantic Analysis". Annual Review of Information Science and Technology. 38: 188–230. doi:10.1002/aris.1440380105.

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

22. 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.

23. 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.

24. 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.

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

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

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

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

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