Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 24, Sayı 2, 455-*463, 2020

Süleyman Demirel University
Journal of Natural and Applied Sciences
Volume 24, Issue 2, 455-463, 2020

DOI: 10.19113/sdufenbed.645579

Türkçe Otel Yorumlarıyla Eğitilen Kelime Vektörü Modellerinin Duygu Analizi ile İncelenmesi

Hüseyin AHMETOĞLU[*1], **Resul DAŞ**[2]

[1]Mardin Artuklu Üniversitesi, Midyat Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, 47500, Mardin, Türkiye
[1](ORCID: https://orcid.org/0000-0002-4320-0198)
[2]Fırat Üniversitesi, Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü, 23119, Elazığ, Türkiye
[2](ORCID: https://orcid.org/0000-0002-6113-4649)

(Alınış / Received: 11.11.2020, Kabul / Accepted: 19.05.2020, Online Yayınlanma / Published Online: 20.08.2020)

Anahtar Kelimeler
Doğal dil işleme,
Veri kazıma,
Duygu analizi,
Yinelenen yapay sinir ağı,
Word2Vec,
Kelime gömme

Özet: Doğal dil işlemenin ve metin sınıflandırmanın önemli araştırma alanlarından biri de duygu analizidir. Bu alanda çalışmalar hızla büyümektedir. Bu teknik dijital yaşamın her çeşit uygulama alanında kendini göstermektedir. Duygu analizi için geliştirilen birçok teknik vardır ancak son zamanlarda doğal dil işlemenin kelime vektör modeli metotları duygu analizinde yaygın olarak kullanılmaya başlamıştır. Word2Vec kelimeleri anlamlı vektörlere dönüştürebilen en kullanışlı kelime vektör modeli yöntemleri arasındadır. Bu yöntem ile kelime vektörleri oluşturabilmek için büyük kelime havuzlarına ihtiyaç vardır. Önceden eğitilmiş modeller duygu analizinde daha doğru sonuçlara ulaşabilmeyi mümkün kılarlar. Bu çalışmada duygu analizinde incelenmek üzere, onaylanmış kullanıcıların Türkçe otel yorumları veri kazıma yöntemleri ile toplanmıştır. Elde edilen bu özgün veriler Word2Vec ile eğitilerek kelime vektörleri oluşturulmuştur. Bu vektörler ile tekrarlanan yapay sinir ağının bir çeşidi olan geçitli tekrarlayan birimler ile bir sınıflandırma modeli geliştirilmiştir. Daha geniş kelime torbalarıyla eğitilmiş kelime vektörleri ile rastgele değerler atanarak oluşturulan vektörler, aynı derin öğrenme yöntemiyle yeniden incelenmiş ve elde edilen sınıflandırma başarıları karşılaştırılmıştır. Elde edilen sonuçlara göre özel alandan bağımsız, daha geniş kapsamlı kelime torbalarının sınıflandırma başarısını arttırdığı gözlemlenmiştir.

Investigation of Word Vector Models Trained with Turkish Hotel Comments by Sentiment Analysis

Keywords
Natural language processing,
Data scraping,
Sentiment analysis,
Recurrent neural networks,
Word2Vec,
Word embeddings,

Abstract: One of the important research areas of Natural Language Processing and text classification is sentiment analysis. Studies in this area are growing rapidly. This technique manifests itself in all kinds of applications of digital life. There are many techniques developed for sentiment analysis, but recently, word embedding methods of natural language processing have become widely used in sentiment analysis. Word2Vec is one of the most useful word embedding methods that can convert words into meaningful vectors. In order to create word vectors with this method, large word pools are needed. Pre-trained models make it possible to achieve more accurate results in sentiment analysis. In this study, Turkish hotel reviews of approved users were collected by data scraping methods for examination of sentiment analysis. Obtained from the original data by training with Word2Vec word vectors were created. With these vectors, a classification model has been developed with Gated Recurrent Unit which is a kind of Recurrent Neural Networks. The vectors formed by assigning random values to wider corpus-trained word vectors were re-examined with the same deep learning method and the obtained classification successes were compared. According to the results, it was observed that the broader corpus independent of the private area increased the success of classification.

1. Giriş

Duygu analizi, duygusal durumları ve subjektif bilgileri sistematik olarak tanımlamak, çıkarmak, ölçmek ve incelemek için doğal dil işleme, metin analizi, hesaplama, dilbilimi ve biyometri kullanımı anlamına gelir. Duygu analizi işletmelerin reklam ve pazarlama süreçlerinden sosyal medyadaki alışkanlık analizlerine, bir markanın duyarlılık analizinden, hükümetlerin politik karar alma

* *Hüseyin AHMETOĞLU: huseyinahmetoglu@artuklu.edu.tr*

süreçlerine kadar birçok alanda önemli rol oynayan pratik bir tekniktir. Duygu analizinde temel görev, verilen bir metnin hangi kutupta yer aldığını belge veya cümle düzeyinde sınıflandırmaktır. Bir belgede ifade edilen görüşün veya bir cümlenin; olumlu, olumsuz veya nötr olup olmadığı duygu analizi işlemleri ile belirlenebilir. Gelişmiş düzeydeki duygu analizleri metinlerdeki duygu durumlarını kızgın, üzgün ve mutlu gibi duygusal durumlara ayırabilir. Metinden duygu analizi yapılırken; veri seti içerisindeki etkisiz kelimeler filtrelenir, her kelimenin sayısal temsili için gösterimler gerçekleşir, sözcük temsilleri ile kelime modeli eğitilir, bu model yardımıyla her kelimenin vektör karşılıkları belirlenir, bu vektörler kullanılarak sınıflandırma modeli eğitilir ve son olarak bu model yardımıyla duygu sınıflandırması gerçekleştirilir.

Duygu analizinde kullanılmak üzere çok çeşitli teknolojiler geliştirilmiştir. Makine öğrenmesi ve özellikle son yıllarda derin öğrenme tekniklerine dayalı sistemler duygu analizinde etkili olarak kullanılmaktadır. Metinden duygu analizi işlemlerinde her kelimenin bir vektör şeklinde gösterimi ile matematiksel temsil elde edilebilir. Kelime vektör modeli uygulamaları sözcüklerin ve belgelerin vektör şeklinde gösterimini sağlayan gözetimsiz derin öğrenme teknikleridir. Bu yöntemler, kelimeler arasındaki sözdizimsel ve anlamsal ilişkileri yakalayabilme yeteneklerinden dolayı metin sınıflandırmada ve duygu analizinde büyük ilgi görmüştür. Derin öğrenmenin en başarılı kelime vektör modeli yöntemlerinden biri Word2Vec'tir [1]. Araştırmacılar bu yöntemi kullanarak duygu analizinde çeşitli uygulamalar geliştirmişlerdir.

Duygu sınıflandırma teknikleri sözlük temelli yöntemler ve makine öğrenimi yöntemleri olmak üzere ikiye ayrılabilir. Makine öğrenmesi yöntemlerinin temelini de derin öğrenme oluşturmaktadır [2], [3]. Sözlük temelli duygu analizi yöntemleri pozitif ve negatif çağrışımlara sahip sözcük ve cümlelerin listesine dayanmaktadır. Bu yöntemde her bir kelimenin negatif ve pozitif duyarlılık değerleriyle elde edilmiş bir sözlüğe ihtiyaç vardır. Bu yaklaşımla geliştirilen yöntemler basit, ölçeklenebilir ve hesaplama açısından verimlidirler [4], [5], [6]. Derin öğrenme yöntemlerinin doğal dil işlemede aldığı roller son zamanlarda büyük bir artış göstermiştir. Doğal dil işlemedeki (Natural Language Processing-NLP) derin öğrenme yöntemlerinde kelime vektörü kullanımı büyük bir artış göstermiştir [7]. Word2Vec ve GloVe gibi kelime vektör modeli algoritmaları, kelimeleri anlamlı vektörlere dönüştürebilen derin öğrenme teknikleridir. Bu algoritmalar kelimeleri sürekli vektörler şeklinde gösterebilirler. Kelimelerin vektör gösterimleri metin sınıflandırma, kümeleme ve bilgi almada çok kullanışlıdır. Kelime vektör modeli teknikleri, sözcük torbaları gösterimine kıyasla bazı avantajlara sahiptir. Örneğin, anlamı yakın olan kelimeler bu vektörel gösterim sayesinde birbirlerine matematiksel olarak daha yakın hale gelmişlerdir. Bu vektörel gösterim aynı zamanda kelimelerin daha küçük boyutlarda temsil edilmesini ve böylece işlem kabiliyeti kazanmalarına olanak sağlamaktadır [1].

Kelime vektör modellerinin doğruluğu, model eğitilirken kullanılan kelime torbasının büyüklüğüyle doğru orantılıdır. Tang ve ark. [8] Twitter için bir duygu analizi çalışması yapmışlar ve özel bir kelime vektör modeli yöntemi önermişlerdir. Severyn ve Moschitti [9] 50 Milyon tweet ile bir kelime torbası oluşturup Word2Vec ile eğitmişlerdir. Önceden eğitilmiş bu kelime vektörlerini evrişimli bir sinir ağına (Convolutional Neural Network-CNN) girdi olarak uygulamışlardır. Fu ve ark. [10] duygu analizi için özyinelemeli otomatik kodlayıcıyı kullanmışlardır. Kelime vektörleri için İngilizce ve Çince Wikipedia kelime torbasını Word2Vec ile eğitmişlerdir. Qin, Xu ve Guo [11], Word2Vec algoritmasını kullanarak 408 milyon kelimeye sahip olan Wikipedia kelime torbasını eğiterek kelime vektörleri oluşturmuşlardır. Önceden eğitilmiş olan vektörleri kullanarak CNN için girdi katmanı olarak kullandılar. Kim, [12] yine önceden eğitilmiş Word2Vec vektörlerini CNN için girdi olarak kullanmış ve duygu analizi başarısını arttırmıştır. Rezaeinia ve ark. [13] kelime vektörleri için yeni bir yöntem önermişlerdir. Önerilen bu yöntemin başarı sonuçlarını diğer yöntemlerle karşılaştırmışlardır. Wang ve ark. [14] ise önceden eğitilmiş vektörleri, en-boy düzeyindeki duygu analizi için dikkat temelli Uzun kısa süreli bellek (Long short-term memory-LSTM) modelinin girdileri olarak uygulamışlardır. LSTM ve GRU gibi tekrarlanan yapay sinir ağı (Recurrent Neural Networks-RNN) tiplerinin kullanılmasındaki amaçlardan biri de RNN'nin değişken uzunluktaki girdilerde öğrenmenin son derece zor ve uzun gerçekleşmesidir. Değişken uzunluklu girdiler, sabit boyutlu bağlam pencereleri ile çözülebilir ancak bağlam pencereleri uzun süreli anlam ilişkilerini yakalayamaz. [15]

Bu çalışmada ise web ortamından toplanmış onaylı Türkçe otel yorumları, yorumların olumlu-olumsuz olma durumlarına göre etiketlenmiştir. Kelime torbası oluşturulmuş ve bu kelime torbası kullanılarak Word2Vec kelime vektörleri elde edilmiştir. Bu kelime vektörleri geçitli tekrarlayan birimler (Gated Recurrent Unit-GRU) için girdi katmanı olarak kullanılmış ve yorumları duygu durumlarına göre sınıflandıran bir model geliştirilmiştir. Sonra Türkçe Wikipedia makaleleri kullanılarak eğitilmiş Word2Vec vektörleri kullanılarak vektör modeli işlemi yeniden yapılmış ve bu sefer modele girdi katmanı olarak bu vektörler sunulmuştur. Rastgele oluşturulan kelime vektörleri de modelde kullanılmış ve sonrasında tüm sınıflandırma sonuçları karşılaştırılmıştır.

2. Materyal ve Metot

Bu çalışmada kullanılan veri seti, veri kazıma yöntemleri kullanılarak elde edilmiştir. Bu veri seti Python'a ait veri işleme kütüphaneleri kullanılarak ön işlemden geçirilmiştir. Word2Vec ile kelime vektörleri oluşturulmuş ve vektörler RNN altındaki GRU ile oluşturulan modele girdi katmanı olarak verilmiştir. Aşağıdaki başlıklarda bu yöntemler açıklanmıştır.

2.1. Veri kazıma

Veri kazıma (Data Scraping) özel araçlar kullanılarak web ortamından veri çıkarım işidir. Veri kazıma ile web üzerinde özel amaçlı verilerin toplanması otomatikleşir. Veri çıkarılıp saklandıktan sonra dağıtık halden daha düzenli bir hale geçer. Veri kazıma işlemleri verilerden sonuçlar elde etmek için kullanılır. Bu sonuçlar genelde ticari sonuçlardır. Örneğin bir ürüne ait en düşük fiyat bilgisi web sitelerinden çekilen otomatik fiyat bilgilerinin karşılaştırılması ile bulunabilir. Firmalar bu bilgiler ışığında sattıkları ürünler için değişken fiyat stratejileri belirleyebilirler. Web sitelerinin standart yapısı istenilen verilere ulaşmada kolaylık sağlar. En yaygın metin tabanlı işaretleme dili olan Hiper Metin İşaretleme Dili'nin (HyperText Markup Language-HTML) etiket yapısına hakim olmak bu süreçte önemlidir. Web ortamında veri kazıma işlemleri çeşitli yöntemlerle gerçekleştirilebilir. Kullanıcı olay denetimini gerektiren durumlarda devreye bot yazılımları girer. Bu çalışmada gerçekleştirilen daha dar kapsamlı veri kazıma işlemlerinde ise sitenin HTML içeriği indirilmiş ve bu içerik parçalanarak istenilen bilgiler seçilip alınmıştır. Python'un BeautifulSoup kütüphanesi bu iş için kullanılan en güçlü ve en hızlı kütüphanedir [16].

2.2. Word2Vec

Word2Vec, kelimeleri vektör uzayında ifade etmeye yarayan gözetimsiz ve tahmin temelli bir derin öğrenme yöntemidir. Google'da araştırmacı Tomas Mikolov ve arkadaşları tarafından 2013 yılında geliştirilmiştir [1]. Word2Vec ile oluşturulan vektörel yapılar üzerine yazılmış araçlar ile anlamsal yönden bir kelimeye en yakın veya en uzak kelimeler listelenebilir ve böylece kelimeler arası anoloji kurulabilir.

Tahmini bir kelime vektörü $\hat{r}$ ve hedef bir kelime vektörü w_t olarak düşünüldüğünde tahmini vektör ile hedef vektörün olasılık hesabı softmax fonksiyonuna göre Denklem 1 ile hesaplanır. W tüm hedef vektörlerin kümesidir.

$$P(w_t|\hat{r}) = \frac{exp(w_t^T \hat{r})}{\sum_{w \in W} exp(w^T \hat{r})} \tag{1}$$

Word2Vec modellerinin maliyet hesabının minimumlaştırılması Deklem 2 ile gerçekleştirilir.

$$\mathscr{L}(w_t, \hat{r}) = -\log P(w_t|\hat{r}) = \log\left(\sum_{w \in W} exp(w^T \hat{r})\right) - w_i^T \hat{r} \tag{2}$$

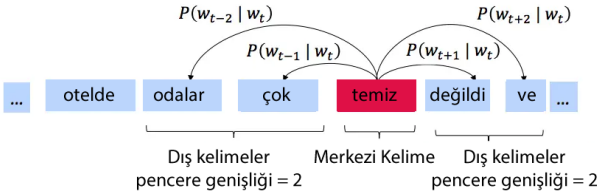
$\mathscr{L}(w_t, \hat{r})$'nin w'a göre gradyan hesabı Denklem 3 ile bulunur.

$$g_1(w, w_t, W, \hat{r}) = \frac{\partial}{\partial w}\mathscr{L}(w_t, \hat{r}) = \hat{r}(P(w|\hat{r}) - I\{w = w_t\}) \tag{3}$$

$\hat{r}$'ye göre gradyan hesabı ise Denklem 4 ile hesaplanır.

$$g_2(w_t, W, \hat{r}) = \frac{\partial}{\partial \hat{r}}\mathscr{L}(w_t, \hat{r}) = \sum_{w \in W}[P(w|\hat{r})w] - w_t \tag{4}$$

Word2Vec için modelin yapısını belirleyen en önemli üst parametre pencere genişliği (window size)'dur. Bu parametre bize, metinde vektörel ifadesi belirlenecek merkezi kelimenin sağında ve solunda kaçar kelime olabileceğini ifade eder. Şekil 1'de pencere boyutu 2 seçildiğinde merkezi kelimenin diğer çevre kelimelere göre olasılık hesaplamalarının nasıl gerçekleştiği gösterilmiştir. Word2Vec'in iki farklı alt algoritması vardır. Bunlar CBOW(Continous Bag of Words) ve Skip-Gram'dır. CBOW ve Skip-Gram modellerinin yapıları birbirlerinden farklıdır. Bu farklılığın ana nedeni girdi ve çıktı katmanlarına verilen verilerdir.



Şekil 1. Pencere genişliği 2 seçildiğinde "temiz" kelimesinin diğer çevre kelimelere göre olasılık durumları

CBOW modelinde pencere genişliğine göre merkez seçilen kelimenin çevresindeki kelimeler girdi katmanını oluşturur. Buna göre bu kelimelerin merkezindeki kelime çıktı katmanında tahmin edilmeye çalışılır. Bu durum Şekil 2'de verilmiştir. CBOW için hedef kelimenin vektörü tüm çevre kelimelerin vektör toplamları alınarak Denklem 5 ile hesaplanır.

$$\hat{r} = \sum_{i-c \le j \le i+c, i \ne j} r_j \tag{5}$$

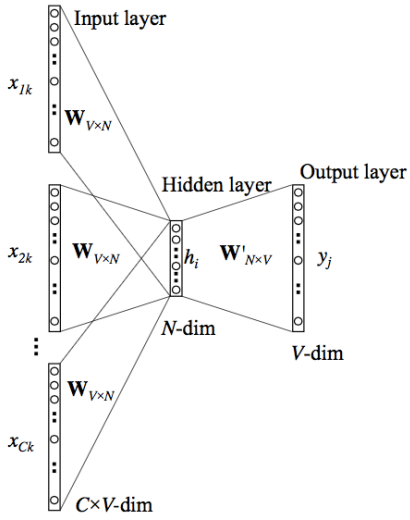
CBOW'un maliyet hesabı Denklem 6 ile yapılır.

$$\mathscr{L}_{CBOW}(c, i) = -\log P(w_i|\hat{r}) \tag{6}$$

Gradyan hesapları ise Denklem 7 ve Denklem 8 ile bulunur.

$$\frac{\partial}{\partial w}\mathscr{L}_{CBOW}(c, i) = g_1(w, w_i, W, \hat{r}) \tag{7}$$

$$\frac{\partial}{\partial r_j}\mathscr{L}_{CBOW}(c, i) = g_2(w_i, W, \hat{r}) \tag{8}$$



Şekil 2. CBOW modeli [17]

Skip-Gram modelinde ise yine pencere genişliğine göre merkezdeki kelime girdi katmanını oluşturur ve bu girdi bilgisine göre merkez kelimenin çevresindeki kelimeler çıktı katmanında tahmin edilmeye çalışılır. Bu işlem Şekil 3'te verilmiştir. Skip-Gram için maliyet fonksiyonu Denklem 9 ile hesaplanır.

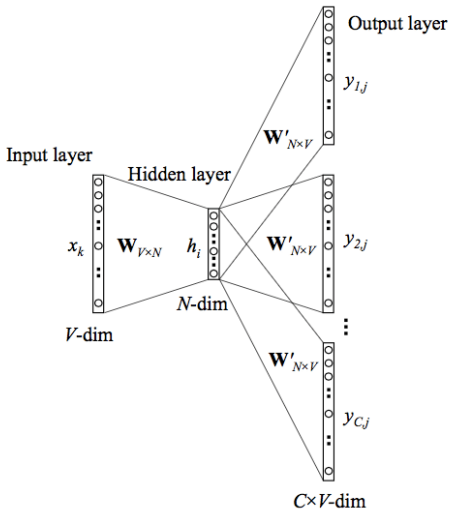
$$\mathscr{L}_{skipgram}(c,i) = \sum_{i-c\leq j\leq i+c, i\neq j} -\log P(w_j|r_i) \quad (9)$$

Gradyan hesapları ise Denklem 10 ve Denklem 11 ile bulunur.

$$\frac{\partial}{\partial w}\mathscr{L}_{skipgram}(c,i) = r_i \sum_{i-c\leq j\leq i+c, i\neq j} g_1(w,w_j,W,r_i) \quad (10)$$

$$\frac{\partial}{\partial r_i}\mathscr{L}_{skipgram}(c,i) = \sum_{i-c\leq j\leq i+c, i\neq j} g_2(w_j,W,r_i) \quad (11)$$

Her iki yöntemde de tüm işlemler cümle bitene kadar tekrar edilir. Bir cümleye uygulanan bu işlemler tüm veri setindeki cümlelere de uygulanır. Böylece başlangıçta etiketsiz olarak bulunan veri vektörel olarak haritalanmış olur. Bu model eğitim için kullanmaya hazırdır. CBOW modelleri genel olarak küçük veri setlerinde daha iyi çalışırken, büyük veri setlerinde Skip-gram daha iyi sonuçlar vermektedir. CBOW daha az işlem gücü gerektirirken, Skip-Gram daha fazla işlem gücüne ihtiyaç duyar. CBOW iki veya daha çok anlamlı kelimeleri anlamakta iyi değilken Skip-Gram iki veya daha çok anlamlı kelimeleri daha iyi öğrenebilmektedir.



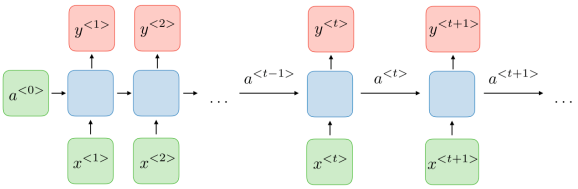
Şekil 3. Skip-gram Modeli [17]

2.3. Derin öğrenme ve tekrarlayan sinir ağları

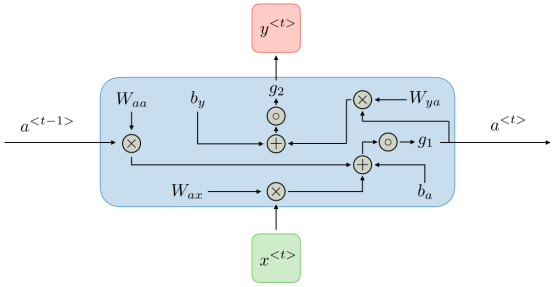
Derin öğrenme; denetimli veya denetimsiz özellik çıkarma, dönüştürme, desen analizi ve sınıflandırma için birçok doğrusal olmayan gizli katmandan yararlanan ve birbirini takip eden bu katmanlarda veriler işlenirken giderek artan şekilde daha kullanışlı gösterimler elde edebilen alt makine öğrenme tekniklerinden biridir. Bu metodun "derinliği" ile birbirini takip eden gösterim katmanlarının sayısı kastedilmektedir. Günümüzde derin öğrenme bir çok disiplin için önemli bir araştırma konusudur [18]. Tekrarlanan yapay sinir ağları (Recurrent Neural Networks-RNN), birimler arasındaki bağlantıların yönlendirilmiş bir döngü oluşturduğu ağlardır. RNN ile dinamik zamansal davranışlar sergilenebilir. İleri beslemeli sinir ağlarından farklı olarak, RNN'ler kendi giriş belleklerini, girdileri işlemek için kullanabilirler. Bu özellik RNN'leri, el yazısı tanıma ve konuşma tanımada, kullanışlı bir yöntem haline getirmektedir. RNN diğer derin öğrenme modellerine göre gizli durumlara sahiptir ve önceki çıktıların girdi olarak kullanılmasına izin veren bir yapıları vardır. Genel yapıları Şekil 4'te verilmiştir. Her bir t zamanı için $a^{<t>}$ aktivasyonu ve $y^{<t>}$ çıktısı Denklem 12 ve Denklem 13 ile ifade edilir. $W_{aa}, W_{ax}, W_{ya}, b_a, b_y$ geçici olarak paylaşılan katsayılardır ve g_1, g_2 de aktivasyon fonksiyonlarıdır. Şekil 5'te RNN'in iç bloğu verilmiştir.

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^t + b_a) \quad (12)$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y) \quad (13)$$



Şekil 4. RNN mimarisi [19]



Şekil 5. RNN blok yapısı[19]

RNN'de girdilerinizin uzunlukları önemli değildir. Girdi büyüklüğünün fazla olması modelin boyutunu etkilemez. Geçmiş bilgiler hesaba alındığı için modelin bir hafızası vardır. Bu şekilde zaman içinde paylaşılan ağırlıklar ortaya çıkar. Buna karşın RNN'de hesaplama yavaştır. Modelin bir hafızası olsa da uzun zaman önceki bilgiye erişim zordur. Mevcut durumda gelecekteki herhangi bir girdinin hesaba katılamaması durumu vardır. RNN'de tüm zaman dilimlerindeki kayıp fonksiyonu $\mathscr{L}$ Denklem 14 ile hesaplanır.

$$\mathscr{L}(\hat{y},y) = \sum_{t=1}^{T_y} \mathscr{L}(\hat{y}^{<t>}, y^{<t>}) \quad (14)$$

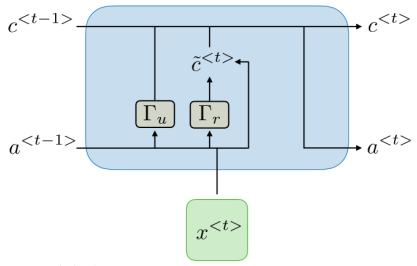
Geriye yayılım, zamanın her noktasında yapılır. T zaman diliminde, ağırlık matrisi W'ye göre $\mathscr{L}$ kaybının türevi Denklem 15 ile ifade edilir.

$$\frac{\partial \mathscr{L}^{(T)}}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathscr{L}^{(T)}}{\partial W}\bigg|_{(t)} \quad (15)$$

RNN'ler, önceki bilgiler ile bağlantı kurup anlamlandırma özelliğinden dolayı, zaman bazlı problemlerde başarılı sonuçlar vermektedir. Ancak RNN'lerde, hangi aktiviteler ne kadar süre hatırlanacak bilinmemektedir. Bütün bilgiler, modelin içerisinde tutulmaktadır. Aktiviteler için bazı bilgiler önemliyken, bazı bilgiler gereksizdir. Bu yüzden bazı aktivitelerin sınıflandırılmasında, tüm geçmişin saklanmasına gerek yoktur. Aktivite sınıflandırılmasında, gerekli bilgi çok önceden oluşmuş ise, bu bilgiye ulaşılamayabilir. RNN'lerin çok önceden olan olayları tahmin edebilmesi için farklı bir mimari yapıya ihtiyaçları vardır. Kaybolan gradyan problemini çözmek için farklı RNN türleri geliştirilmiştir. Bu tür problemlerde daha iyi çalışan RNN'lerin, bir çeşidi olan Uzun Kısa-Süreli Bellek (Long Short Term Memory - LSTM) ağları kullanılmaktadır. Bu RNN türlerinde belirli kapılar kullanılır, bu kapılar iyi tanımlanmış belirli bir amaca sahiptir. Γ olarak ifade edilen kapı değerleri Denklem 16 ile hesaplanır. W, U, b kapıya özgü katsayılardır ve σ ise sigmoid fonksiyondur.

$$\Gamma = \sigma(Wx^{<t>} + Ua^{<t-1>} + b) \tag{16}$$

Geçitli tekrarlayan birimler (Gated Recurrent Unit-GRU) ise 2014 yılında Cho ve ark. [20] tarafından tanıtılan unutma ve giriş kapıları tek bir kapı üzerinde birleştirilerek bir güncelleme kapısının oluşturulduğu bir modeldir. Elde edilen model, geleneksel LSTM modelinden, daha basit olduğundan gün geçtikçe GRU daha popüler hale gelmektedir. Şekil 6'da GRU'nun yapısı verilmiştir. Şekil 6'ya göre GRU mimarisinin denklemleri Denklem 17, Denklem 18 ve Denklem 19'da verilmiştir. Γ_u güncelleme kapısı, Γ_r uygunluk kapısıdır.



Şekil 6. GRU blok yapısı [19]

$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c) \tag{17}$$

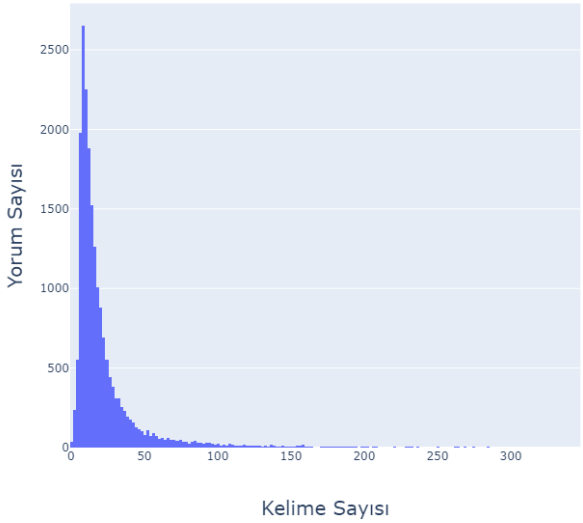
$$c^{<t>} = \Gamma_u \star \tilde{c}^{<t>} + (1 - \Gamma_u) \star c^{<t-1>} \tag{18}$$

$$a^{<t>} = c^{<t>} \tag{19}$$

3. Bulgular

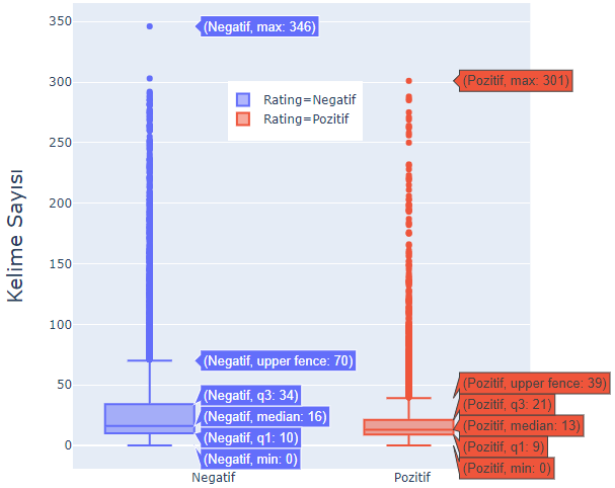
Bu çalışma toplamda üç aşamadan oluşmaktadır. İlk aşamada web ortamından veri kazıma yöntemleri ile toplanan otel yorumları ve puanlarıyla bir veri seti oluşturulmuş ve bu veri setindeki her bir örnek pozitif ve negatif olarak etiketlenmiştir. İkinci aşamada veri setinin kelime torbası kullanılarak Word2Vec algoritmasıyla kelime gömme modeli oluşturulmuştur. Bu aşamada aynı zamanda Türkçe Wikipedia makaleleri ile geniş bir kelime torbası kullanılarak başka bir kelime gömme modeli daha oluşturulmuştur. Üçüncü aşamada ise veri seti test ve eğitim verisi olarak ikiye ayrılmış ve kelime temsilleri için farklı kelime gömme modelleri yardımıyla farklı vektörler kullanılarak sınıflandırma modellerinin eğitimi gerçekleştirilmiştir.

Web ortamından toplanan otel yorumalarıyla bir veri seti oluşturulmuştur. Veri seti toplamda 20179 adet farklı yorumdan oluşmaktadır. Bu yorumların her biri olumlu ya da olumsuz olarak etiketlenmiştir. Web ortamından toplanmış otel yorumları, kullanıcıların oteller hakkındaki yazıya dökdükleri yorumlar ve otele 10 üzerinden verdikleri yıldız sayılarından meydana gelir. Her yorum işlenirken yıldız sayılarına göre etiketlenmiştir. Etiketleme işlemi yorumların yıldız sayıları üzerinden gerçekleştirilmiştir. 4 ve altı yıldız barından yorumlar negatif, 6 ve üstü yıldızı olan yorumlar pozitif olarak etiketlenmiştir. Negatif etiketli yorumların sayısı 7996 (%40), pozitif etiketli yorumların sayısı 12183 (%60) 'tür. Veri setinde yer alan her yorumun kelime uzunlukları farklıdır. Tüm veri seti incelendiğinde yorumların kelime sayılarına göre dağılımları Şekil 7'de verilmiştir. Farklı etiketlere ait yorumların kelime sayılarına göre detaylı durumları Şekil 8'de kutu grafiğinde verilmiştir.Veri setinde tekrar eden kelimeler incelenmiştir. Buna göre veri setinde en çok tekrar eden 20 kelime geçme frekanslarına göre Şekil 9'da verilmiştir. Tüm bu grafikler oluşturulmadan önce veri seti NLP'nin ön işlem adımlarından geçirilmiştir. Buna göre veri setinde yer alan tüm noktalama işaretleri, HTML etiketleri, emoji karakterleri, Türkçe etkisiz kelimeler çıkarılmış ve tüm kelimeler küçük harf formatına getirilmiştir. Şekil 10'da veri setinde yer alan kelimeler geçme sıklıklarına göre kelime bulutunda gösterilmiştir. Bu işlem sırasında veri setindeki noktalama işaretleri kaldırılmış, veri seti tek bir String haline getirilmiş, veri setindeki etkisiz kelimeler atılarak tüm kelimeler bir listeye aktarılmıştır.

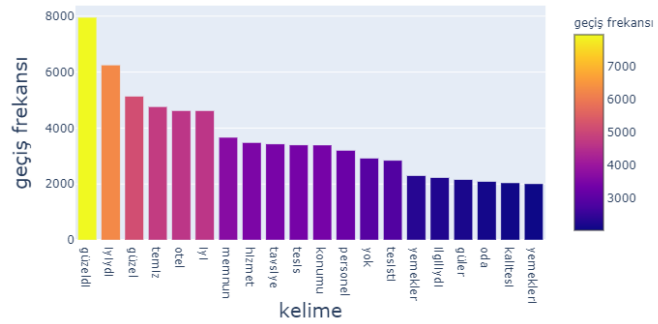


Şekil 7. Kelime sayılarına göre yorum yoğunluğu

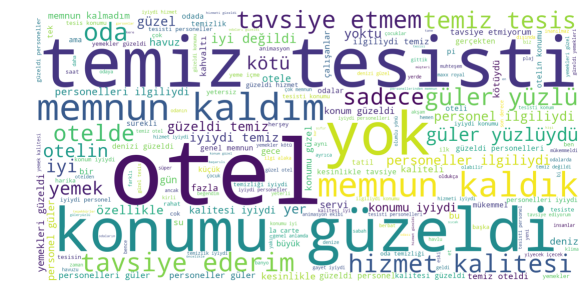
Listeye aktarılmış kelimelerin vektörel karşılıklarını bulabilmek için *Gensim* kütüphanesi [21] kullanılarak Word2Vec ile bir model eğitilmiştir. Bu model eğitilirken her bir kelimenin vektörel gösterim boyutu olarak 400 seçilmiştir. Kullanılacak alt algoritma olarak CBOW



Şekil 8. Kategorilere göre kelime sayıları



Şekil 9. Yorumlarda en çok geçen 20 kelimenin frekansı



Şekil 10. Kelime bulutu

modeli için pencere genişliği 5 olarak seçilmiştir. Eğitilen modelin t-SNE [22] ile 2 boyutlu düzleme aktarılması Şekil 15'te verilmiştir.

Veri setinin sözcük temsilinde *Keras Tokenizer* [23] kullanılmıştır. Bu işlem sırasında toplam 39276 farklı sözcük olduğu görülmüştür. Toplam 20179 farklı yorum, giriş matrisinin satır sayısını verecektir. Sütun sayısının belirlenmesi performans-maliyet açısından kritiktir. Bu sayı belirlenirken Denklem 20'den yararlanılmıştır. Bu formül oluşturulurken her yorumdaki kelimelerin sayısı bir listeye aktarılmıştır. s bu listenin standart sapması, m bu listenin ortalama değeridir. T_{max} ise maksimum sözcük sayısıdır.

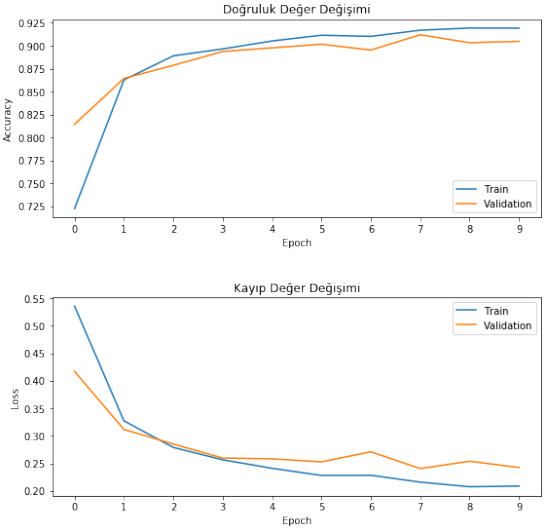
$$T_{max} = m + 2 * s \tag{20}$$

Denklem 20 ile giriş katmanı için sözcük sayısı olarak 86 elde edilmiştir. Bu sayının yorumlardaki tüm kelimeler düşünüldüğünde temsil değeri %96'dır. Buna göre yorumlardaki kelime sayısı 86 dan az olan yorumlar bu değer tamamlanıncaya kadar sıfır ile doldurulmuştur. Kelime sayısı 86 dan fazla olan yorumlardan rasgele kelimeler atılmıştır. Word2Vec ile eğitilmiş model kullanılarak veri setindeki her bir kelimenin temsil karşılığına göre modelden o kelimenin vektörel gösterimi seçilmiş ve giriş matrisi elde edilmiştir.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 86, 400)           15710800
_________________________________________________________________
cu_dnngru_1 (CuDNNGRU)       (None, 86, 32)            41664
_________________________________________________________________
cu_dnngru_2 (CuDNNGRU)       (None, 16)                2400
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 17
=================================================================
Total params: 15,754,881
Trainable params: 44,081
Non-trainable params: 15,710,800
_________________________________________________________________
```

Şekil 11. GRU modeli özeti

Sınıflandırma için oluşturulan GRU modelinin özeti Şekil 11'de verilmiştir. Eğitim için veri setinin %75'i alınmış, bu verilerin de %25'i doğrulama verileri olarak kullanılmıştır. Bu modelin eğitim sırasında gerçekleşen doğruluk ve kayıp değişimleri Şekil 12'de verilmiştir. Eğitim sonun da elde edilen modele test verileri sunulmuş ve sınıflandırma modeli için alınmış başarı sonuçları Tablo 1'de verilmiştir.



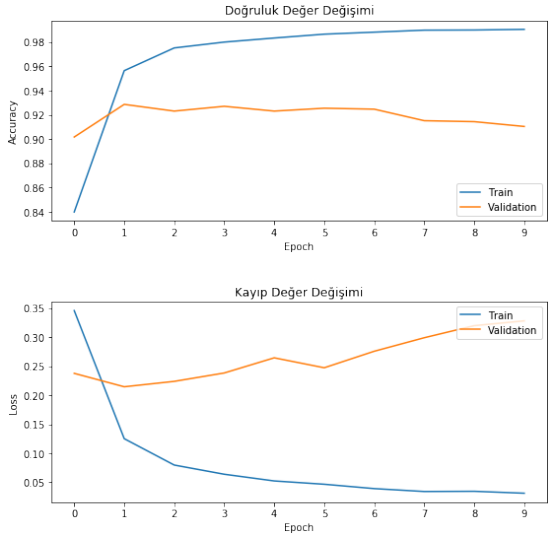
Şekil 12. Otel kelime vektör modeli ile GRU modeli doğruluk-kayıp değişimi

Tablo 1. Özel Kelime vektör modeli ile Test Sonuçları

Sınıf	Tutarlılık	Anma	f1-skor
Negatif	0.89	0.89	0.89
Pozitif	0.93	0.92	0.92

İkinci modelin giriş katmanına her kelime için rasgele atanmış sayılarla oluşturulmuş matrisler verilmiştir. Bu modelin diğerlerinden farkı kelimeler arasındaki anlam ilişkileri, modelin eğitimi gerçekleşirken belirlenmesidir, yani model eğitilirken kelime vektörleri de aynı anda eğitilmektedir. Dolayısıyla eğitim süresi diğer modellere kıyasla

daha uzun sürmüştür. Bu modelin eğitim sırasında gerçekleşen doğruluk ve kayıp değişimleri Şekil 13'te verilmiştir. Eğitim sonun da elde edilen modele test verileri sunulmuştur. Sınıflandırma modeli için başarı sonuçları Tablo 2'de verilmiştir.



Şekil 13. Rasgele kelime vektör modeli ile GRU modeli doğruluk-kayıp değişimi

Tablo 2. Rasgele Kelime vektör modeli ile Test Sonuçları

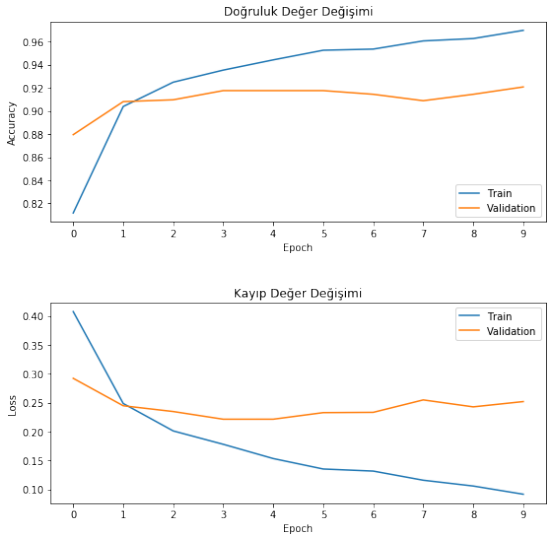
Sınıf	Tutarlılık	Anma	f1-skor
Negatif	0.91	0.88	0.89
Pozitif	0.92	0.94	0.93

Eğitilen üçüncü modele giriş matrisi olarak farklı bir kelime vektör modelinden elde edilmiş vektörel değerler sunulmuştur. Bu kelime vektör modeli Türkçe Wikipedia makalelerinin [24] işlenmesiyle oluşturulmuştur. Word2Vec ile oluşturulan modelde 412457 benzersiz kelimenin 400 sütundan oluşan vektörel karşılıkları bulunmaktadır. Tüm veri setindeki kelimeler bu modelde taranmış ve vektörel karşılıkları bulunarak giriş matrisi elde edilmiştir. Bu modelin eğitim sırasında gerçekleşen doğruluk ve kayıp değişimleri Şekil 14'te verilmiştir. Bulunan ağırlıklara göre eğitim sonun da elde edilen modele test verileri sunulmuştur. Sınıflandırma modeli için başarı sonuçları Tablo 3'te verilmiştir.

Tablo 3. Genel Kelime vektör modeli ile Test Sonuçları

Sınıf	Tutarlılık	Anma	f1-skor
Negatif	0.90	0.91	0.90
Pozitif	0.94	0.93	0.94

Her üç GRU modelinde de girdi ve çıktı katmanları dahil toplam 4 katman vardır. Kayıp değer için *Binary Crossentropy* ve optimizasyon için *Adam* algoritması kullanılmıştır. Eğitim toplam 10 epok sürmüş ve *Batch Size* olarak 256 değeri seçilmiştir. Birinci Gizli katmanda birim sayısı 32, ikinci gizli katmanda birim sayısı 16'dır. İkili sınıflandırma



Şekil 14. Wikipedia kelime vektör modeli ile GRU modeli doğruluk-kayıp değişimi

yapıldığı için çıktı katmanında aktivasyon fonksiyonu olarak *Sigmoid* kullanılmıştır. Eğitimler 15135 veri üzerinde gerçekleşirken, 1261 doğrulama verisi kullanılmıştır. Test verileri ise 3784 örnekten oluşmaktadır. Her üç modele sunulan test verilerinin doğruluk ve kayıp değerleri Tablo 4'te verilmiştir.

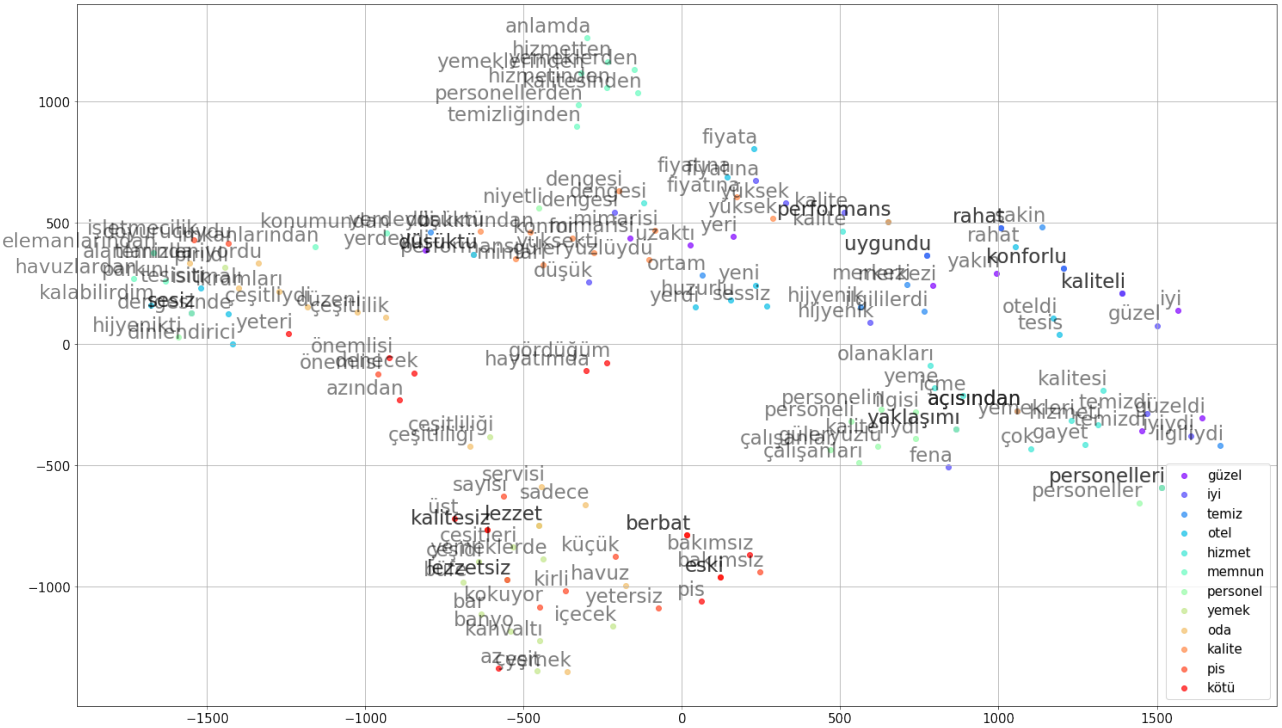
Tablo 4. Test Doğruluk ve Kayıp Sonuçları

Kelime vektör modeli Türü	Doğruluk	Kayıp
Özel	0.91	0.22
Genel	0.92	0.24
Rasgele	0.91	0.34

4. Sonuç

Dil insanların kendilerini ifade edebilmelerini sağlayan güçlü bir mekanizmadır. Makinelerin insanlarla etkileşimi, bu mekanızmanın anlamlandırılması ile gerçekleşir. NLP ile makineler, metinlerin kendilerince daha anlamlı sayısal karşılıklarını kullanarak ilişkisel çıkarımlar elde ederler. Bu görevlere hizmet edebilmek için gerçekleştirilen bu çalışmada ham verinin elde edilerek işlenmesi ve analizi, bu verinin sınıflandırılmaya uygun hale getirilmesi ve makine öğrenmesi ile sınıflandırılması işlemleri gerçekleştirilmiştir.

Word2Vec ile eğitilen modellerin etkin kelime torbalarına ihtiyaçları vardır. Bu çalışmada farklı kapsamlı kelime torbaları ile eğitilmiş modellerin sınıflandırma başarısına etkileri incelenmiştir. Sınıflandırmada kullanılan veri seti kendi içinde eğitilerek kelime vektörleri için bir model oluşturulmuştur. Bu modelin sınıflandırmadaki başarısı ile rasgele oluşturulmuş ve başka bir kelime torbası kullanılarak oluşturulmuş modeller karşılaştırılmıştır. Geniş konu yelpazesine sahip Türkçe makaleler kullanılarak oluşturulmuş kelime gösterim modelinin sınıflandırma başarısının daha yüksek olduğu gözlemlenmiştir. Rasgele oluşturulmuş kelime vektörleri ile gerçekleştirilen sınıflandırma modelinin başarımı diğer modellere göre



Şekil 15. Word2Vec modelinin t-SNE ile 2 boyutlu düzleme aktarılması

daha azdır. Bu farkın sebebi eğitim sırasında kelimeler arasındaki anlam ilişkilerinin çıkarılmasıdır. Bu durumun kayıp değere etkisi eğitim sırasında gözlemlenmiş ve rasgele kelime vektörleri ile oluşturulmuş modelin kayıp değerinin daha yüksek olduğu görülmüştür. Önceden eğitilmiş kelime vektör gösterimlerinin kayıp değer etkisi daha azdır. En düşük kayıp değer veri setinin kendisi kelime modeli olarak kullanıldığında elde edilmiştir. Bu duruma, sınıflandırma modelinin kullandığı kelimeler arasındaki anlam ilişkisini daha kolay çözebilmesi sebep olmuştur.

Doğal dil işlemede, kelimelerin sayısal temsillerinin olşturulması aşaması en kritik bölümlerden biridir. Bu sürecin başarısı tüm uygulamayı etkileyecektir. Bu çalışma ile literatüre sözcüklerin sayısal temsilleri noktasında bir bakış açısı kazandırılmak istenmiştir. Kullanılacak kelime torbalarının dar ve geniş kapsamdaki uygulamalarının sonuçları karşılaştırmalı olarak gözlemlenmiştir. Elde edilen sonuçlar, gözetimsiz kelime vektörü çıkarımı işlemleri için bir ön bilgi niteliğindedir.

Bu çalışmanın farklı alanlarda özel geliştirilmiş karar verme sistemleri için örnek teşkil edeceği düşünülmektedir. Çalışmada gerçekleştirilen; verinin toplanması ve analizi süreci, veriler üzerinde gözetimsiz öğrenme algoritmaları ile sözcük temsillerinin oluşturulması süreci ve geliştirilen karar destek modelleri ile bu çalışmanın farklı disiplinlere ilham vermesi beklenmektedir.

Kaynakça

[1] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations ofwords and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013.

[2] Z. Hailong, G. Wenyan, and J. Bo, "Machine learning and lexicon based methods for sentiment classification: A survey," in *Proceedings - 11th Web Information System and Application Conference, WISA 2014*, pp. 262–265, 2014.

[3] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[4] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *KDD-2004 - Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177, 2004.

[5] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *WSDM'08 - Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 231–239, 2008.

[6] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-basedmethods for sentiment analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[7] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Systems with Applications*, vol. 77, pp. 236–246, 2017.

[8] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *52nd Annual Meeting of the Association for Computational*

Linguistics, ACL 2014 - Proceedings of the Conference, vol. 1, pp. 1555–1565, 2014.

[9] A. Severyn and A. Moschitti, "UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification," pp. 464–469, the 38th International ACM SIGIR Conference, 2015.

[10] X. Fu, W. Liu, Y. Xu, and L. Cui, "Combine HowNet lexicon to train phrase recursive autoencoder for sentence-level sentiment analysis," *Neurocomputing*, vol. 241, pp. 18–27, 2017.

[11] P. Qin, W. Xu, and J. Guo, "An empirical convolutional neural network approach for semantic relation classification," *Neurocomputing*, vol. 190, pp. 1–9, 2016.

[12] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1746–1751, 2014.

[13] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, "Sentiment analysis based on improved pre-trained word embeddings," *Expert Systems with Applications*, vol. 117, pp. 139–147, 2019.

[14] Y. Wang, M. Huang, xiaoyan Zhu, and L. Zhao, "Attention-based LSTM for Aspect-level Sentiment Classification," pp. 606–615, 2016.

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] Beautiful Soup, "Beautiful soup documentation." https://www.crummy.com/software/BeautifulSoup/bs4/doc/, 2019. [Online; accessed 12-October-2019].

[17] Stokastik, "Understanding word vectors and word2vec." https://www.stokastik.in/understanding-word-vectors-and-word2vec/, 2019. [Online; accessed 12-October-2019].

[18] H. Ahmetoğlu and R. Daş, "Derin Öğrenme ile büyük veri kumelerinden saldırı türlerinin sınıflandırılması," in *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–9, Sep. 2019.

[19] Shervine Amidi-Stanford University, "Recurrent neural networks." https://stanford.edu/~shervine/l/en/teaching/cs-230/cheatsheet-recurrent-neural-networks, 2019. [Online; accessed 12-October-2019].

[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1724–1734, 2014.

[21] R. Rehurek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, (Valletta, Malta), pp. 45–50, ELRA, May 2010. http://is.muni.cz/publication/884893/en.

[22] L. Van Der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2625, 2008.

[23] F. Chollet *et al.*, "Keras." https://keras.io, 2015. [Online; accessed 12-October-2019].

[24] W. Contributors, "Wikimedia downloads." https://dumps.wikimedia.org/, 2019. [Online; accessed 12-October-2019].