

# Kelime Vektörü Yöntemlerinin Model Oluşturma Sürelerinin Karşılaştırılması

*Araştırma Makalesi/Research Article*

 Metin BİLGİN<sup>1</sup>

<sup>1</sup>Bilgisayar Mühendisliği, Bursa Uludağ Üniversitesi, Türkiye

[metinbilgin@uludag.edu.tr](mailto:metinbilgin@uludag.edu.tr)

(Geliş/Received:19.10.2018; Kabul/Accepted:12.04.2019)

DOI: 10.17671/gazibtd.472226

**Özet**— Bu çalışmada duygu analizi için oluşturulan iki farklı veri kümesi, kelime vektörü algoritması olan Word2Vec ile modellenmiştir. Model oluşturulurken Word2Vec'in iki farklı yöntemi olan CBoW (Continuous Bag of Words) ve Skip-Gram kullanılmıştır. Word2Vec ile bir metnin modelini oluşturmak için genellikle Ortalama yöntemi kullanılmaktadır. Bu çalışmada hem CBoW hem de Skip-Gram yöntemleriyle bir metni modellemek için üç farklı yöntem önerilmiştir. Model oluşturma (eğitim zamanı) süreleri her ikisi içinde ölçülmüştür. Sonuç olarak modelleme süresi açısından CBoW'un Skip-Gram'dan daha başarılı olduğu deneysel olarak gösterilmiştir.

**Anahtar Kelimeler**— Word2Vec, model süresi, eğitim süresi , CBoW, Skip-Gram, doğal dil işleme

## Comparison of Modeling Time of Word Vector Methods

**Abstract**— In this study, two different datasets for sentiment analysis have been modeled by Word2Vec that it is a word vector algorithm. While the model is creating that has used two different methods CBoW and Skip-Gram of Word2Vec. Generally, the arithmetic mean is used for modeling a text with Word2Vec. In this study, three different methods for modeling a text are suggested on both CBoW and Skip-Gram. Its modeling time (training time) is measured. As a result, it was experimentally shown that CBoW is more successful than Skip-Gram based for modeling time.

**Keywords**— Word2Vec, modeling time, training time, CBoW, Skip-Gram, natural language processing

### 1. GİRİŞ (INTRODUCTION)

Kelime vektörleri son zamanlarda geliştirilen, danışmansız bir şekilde çalışan ve son yıllarda doğal dil işleme uygulamalarında kullanımı oldukça artan bir yöntemdir. Word2Vec olarak isimlendirilen bu yöntem ile verilen bir metin içindeki tüm kelimeler tanımlanan bir uzayda vektörler ile ifade edilir. Oluşturulan bu vektörler yardımıyla istenilen kelimeler arasındaki benzerlik ya da zıtlık vb. durumlar matematiksel olarak ifade edilebilir. Word2Vec'in kullanıldığı çalışmalar duygu analizi, metin sınıflandırma, metin temsili vb. konularla özetlenebilir.

Zhang ve ark. bir ürün hakkında yapılan 100.000 Çince yorum üzerinde pozitif-negatif şeklinde bir duygu analizi çalışması yapmışlardır. Sözlük (lexicon) ve cümle ögesi (parts of speech) üzerine yapılan iki farklı çalışma için Word2Vec tabanlı bir sistem kullanılmış ve sözlük

tabanında %89.95 ve cümle ögesini bulma çalışmasında %90.3'lük başarı oranlarına ulaşmışlardır [1].

Dickinson ve ark. bir şirkete ait hisse senedindeki değişimin hisse senedi ile ilgili atılan tivitlerin (tweet) nasıl etkilediğini araştırmak için duygu analizi tabanlı bir sistem geliştirmişlerdir. N-gram ve Word2Vec ile gerçekleştirilen çalışmada n-gram %65 başarı ile sınıflandırma gerçekleştirirken Word2Vec %75'lik bir başarılı sınıflandırma değerine ulaşmıştır [2].

Tang ve ark. tivitler üzerinde duygu analizi çalışması gerçekleştirmişlerdir. Çeşitli n-gramlar ve Word2Vec yöntemi ile gerçekleştirilen duygu analizi çalışmasında %77.33'lük doğru sınıflandırma değerlerine ulaşmıştır [3].

Polpinij ve ark. gerçekleştirdikleri çalışmalarında bir otele ait web sayfası üzerinden otel ile ilgili yapılan yorumların sınıflandırılması için Word2Vec tabanlı bir sistem

geliştirmişlerdir. Sınıflandırma için SVM (Support Vector Machines) algoritması kullanılmıştır [4].

Şahin, Türkçe metinler üzerinde Word2Vec ile model kuran ve SVM ile sınıflandıran bir sistem tasarlamıştır. Gerçekleştirilen çalışmada 22.729 belge üzerinde sınıflandırma işlemi gerçekleştirilmiştir [5].

Xue ve ark. Çinde kullanılan Sina Weibo adındaki bir mikroblog sitesi üzerinde duygu analizi çalışması gerçekleştirmiştir. Model oluşturmak için Word2Vec kullanılmış ve yapılan sınıflandırmada pozitif veriler %94, negatif veriler için %96 ve nötr veriler için %85'lik doğru sınıflandırma başarılarına ulaşılmıştır [6].

Bilgin ve Şentürk, Türkçe ve İngilizce metinler üzerinde Word2Vec'in doküman modellemek için genişletilmiş versiyonu olan Doc2Vec kullanarak, veri setinin modelini oluşturmuş ve sınıflandırma başarılarını ölçmüşlerdir [7]

Bilgin ve Köktaş, Türkçe metinler üzerinde yaptıkları duygusal analizi çalışmasında 3 sınıflı bir veri seti üzerinde kelime vektörlerini kullanmışlardır. Word2Vec temelli model oluşturma yaklaşımı kullanarak 2 farklı makine öğrenmesi yöntemiyle sistemin performansını test etmişlerdir [8].

Mahdaouyve ark. Word2Vec kelime vektörlerini ve Doc2Vec doküman vektörlerini kullanarak Arapça metinlerin sınıflandırılması üzerine bir çalışma gerçekleştirmişlerdir. Yaptıkları çalışmada vektörlerin boyutlarının azaltılmasının doğruluk üzerindeki olumlu etkilerini ve BOW'a (Bag of Words)'a göre kelime vektörlerinin kullanılmasının başarıyı artırdığını yaptıkları çalışmada göstermişlerdir [9].

Çoban ve Karabey yaptıkları müzik türü sınıflandırma çalışmasında geleneksel yöntem olan BOW'u, Word2Vec-Doc2vec ile kıyaslayarak SVM tabanlı bir metin sınıflandırma işlemi gerçekleştirmişlerdir. Ayrıca önışleme adımlarının ve vektör boyutlarının doğruluk üzerindeki etkilerini incelemişlerdir [10].

Ma ve ark. ders seçiminde anlamsal benzerlik analizi için TF-IDF, Word2Vec ve Doc2Vec tabanlı bir sistem önermiş ve ders seçimindeki başarımlarını ölçmüşlerdir [11].

Razzaghoori ve ark. Pers (İran) dilinde soru sınıflandırma için kelime vektörlerini ve frekanslarını kullanmışlardır. Sonuçların sınıflandırılması için SVM ve Neural Network sınıflandırıcıları kullanılmıştır [12].

Polat ve Körpe yaptıkları çalışmada TBMM (Türkiye Büyük Millet Meclisi) tutanakları üzerinde Word2Vec ve Glove ile modeller oluşturularak kullanılan bir kavramın en yakın anlamlı kavramları bulunmaya çalışılmıştır. Çalışmada Word2Vec ile oluşturulan model ile daha yüksek doğrulukta sınıflandırma yapan bir sistem tasarlanmıştır [13]

Bu makalenin tasarımı; giriş bilgilerinin ve literatür araştırmasının verildiği 1.Bölüm, kullanılan metotların anlatıldığı 2.Bölüm, kullanılan veri kümesi hakkında bilgilerin verildiği 3.Bölüm, yapılan deneylerle ilgili bilgilerin verildiği 4.Bölüm, sonuçların verildiği 5.Bölüm ve sonuçların tartışıldığı 6.Bölüm ile kaynakça bölümlerinden oluşmaktadır.

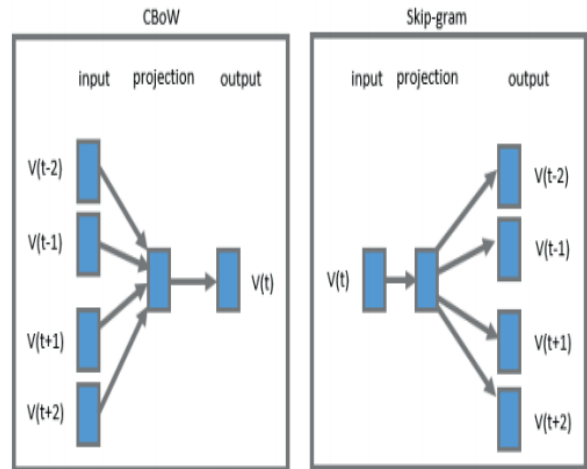
## 2. METOD (USAGE)

### 2.1. Word2Vec

Kelime vektörleri yöntemi, kelimeleri n boyutlu bir uzayda birer vektör olarak temsil etmek ve bu yol ile kelimeler arası uzaklıkları hesaplayarak aralarındaki anlamsal benzerliği tespit etme amacıyla önerilmiştir [14]. Word2Vec kelimeleri gömmek için kullanılan sinir ağı temelli bir yaklaşımdır. Büyük bir metin kümesi ile eğitilen bu model yüksek boyutlu uzayda her bir kelime için benzersiz bir vektör oluşturur. Oluşturulan bu benzersiz vektörlerin özelliği veri kümesindeki benzer anlamdaki kelimelerin birbirlerine yakın vektörler oluşturmalarıdır. CBoW ve Skip-gram olmak üzere iki metodu bulunmaktadır.

CBoW bir kelimeyi tahmin etmek için kelimeyi çevreleyen bir bağlamı kullanırken, Skip-gram sabit bir pencere boyutuna sahip kelimeleri çevreleyerek kelimeyi tahmin etmeye çalışır. Skip-Gram seyrek kelimeler için daha iyi sonuçlar üretebilir [14]. CBoW ve Skip-Gram yöntemleri Şekil 1'de görülmektedir.

Word2Vec işlemi sonucunda her kelimenin bir vektöre sahip olduğu bir sözlük elde edilmiş olur. Bu çalışmada duygu sınıflandırılması yapıldığı için amaç bu sözlüğü kullanarak doküman vektörlerini çıkarabilmektir. Bunu gerçekleştirmek için öncelikle dokümandaki her kelime vektörleriyle eşleştirilir ve  $k \times w$  boyutunda bir matris elde edilmiş olur. Burada  $k$ , dokümandaki kelime sayısını gösterirken  $w$ , vektör boyutunu gösterir.



Şekil 1. Word2Vec (CBoW ve Skip-gram) [8]  
(Word2Vec (CBoW and Skip-Gram))

### 3. VERİ KÜMESİ (DATA SETS)

Gerçekleştirdiğimiz çalışmada pozitif ve negatif olarak etiketlenmiş iki sınıfa ait verileri barındıran iki farklı veri kümesi kullanılmıştır. X ismini verdiğimiz veri kümemiz, Amazon sitesinden toplanan kullanıcı yorumlarını içermektedir [15]. Y ismini verdiğimiz veri kümemiz ise, Openmind sitesi üzerinden elde edilmiştir [16]. Veri kümelerinin içerdiği örnek sayıları Tablo 1’de görülmektedir.

Tablo 1. Veri Kümelerine ait Bilgiler  
(Data sets’ Properties)

Veri Kümesi	Pozitif Örnek Sayısı	Negatif Örnek Sayısı	Toplam Örnek Sayısı
X (Amazon)	5311	6021	11332
Y (Openmind)	3995	3091	7086

#### 3.1. Ön İşlem

Doğal Dil İşleme (DDİ) problemlerinde metinler üzerinde çeşitli ön işlemler gerçekleştirilmelidir. Bu işlemler yapılan uygulamanın doğruluğunu artırmaya yardımcı olduğundan önemlidir. Çalışma ortamı olarak KNIME ortamı kullanılmıştır [17]. Ön işlemler içinde aynı ortamda bulunan bazı kütüphaneler kullanılmıştır. İlk basamakta OpenNLPWhiteSpaceTokenizer ile tokenization işlemi gerçekleştirilmiştir. Ardından metinler küçük harfe çevrilmiştir ve ardından gövdeleme işlemi gerçekleştirilmiştir. Sonraki işlem ise Durma Kelimeleri (Stopwords) ve noktalama işaretlerinin temizlenmesi işlemidir.

Durma kelimeleri “a, the, and” gibi belgenin anlamından çok biçimiyle alakalıdır ve silinmesi metnin anlamında bir değişikliğe sebep olmayacaktır. Bu sebeple metin içerisindeki durma kelimeleri temizlenmiştir. Yapım ekleri ile kelimeler yeni anlamlar kazansalar da çekim ekleri ile aynı durum söz konusu değildir. Bu durumda çekim eki almış 2 kelime farklı kelime gibi değerlendirilirse işlem boyutu gereksiz yere artırmış olacaktır. Gövdeleme yapılmasının amacı bu durumu engellemektir. Gövdeleme ile kelimelerin kendileri yerine kökleri kullanılmaktadır.

### 4. DENEY (EXPERIMENT)

Bu çalışmada model oluşturmak için CBoW ve Skip-Gram gibi iki farklı Word2Vec yöntemi kullanılmıştır. Bu çalışmada yapılan araştırma Word2Vec algoritmasının iki farklı yöntemi olan CBoW ve Skip-Gram için model oluşturma sürelerinin araştırılması üzerinedir. Çalışmanın gerçekleştirildiği ortamın parametreleri;

- Intel Core i5-7200 2.5 GHz CPU
- 8 GB Ram
- 64 Bit İşletim Sistemi

şeklinde.

Word2Vec aslında kelime tabanlı bir vektör oluşturma yöntemidir. Bu çalışma ise verilen bir metni duygusal yönden sınıflandırmak üzerinedir. Sınıflandırma yapılırken daha önceden belirlenen 2 farklı sınıf (pozitif, negatif) kullanılarak metnin vermek istediği duygu tahmin edilmeye çalışılmıştır.

Gerçekleştirmek istediğimiz işlem aslında Doc2Vec’in temel alanına girmektedir. Word2Vec gibi kelime vektörlerini kullanarak bu tarz bir problemin çözümü için Word2Vec’in oluşturduğu vektör üzerinde bazı işlemler yapılması gerekmektedir. Bunun için literatürde yapılan bazı çalışmalar mevcuttur. Literatürdeki çalışmalarda bir metin içerisindeki kelime vektörleri Word2Vec yardımıyla oluşturulduktan sonra oluşan her bir özelliğin aritmetik ortalaması alınarak her metin dosyası başlangıçta belirtilen özellik sayısı ile ifade edilmektedir.

Gerçekleştirilen çalışmada aritmetik ortalamaya alternatif olarak oluşturulan vektörlerin medyanının (Medyan), maksimumunun (Max) ve en düşük-en yüksek (Min-Max) değerlerin ortalamasının alınması şeklinde üç farklı yeni yöntem önerilmiş ve bu üç yeni yöntemin model oluşturma süreleri araştırılarak iki farklı Word2Vec yöntemi için ölçümlenmiştir.

Aritmetik ortalama bir ana kütle veya bir örneklem veri değerlerinin toplamının o ana kütledeki terim sayısına veya örneklem büyüklüğüne bölünerek elde edilen merkezsel konum değeridir [18]. Aritmetik ortalamaya ait formül Eşitlik 1’de görülmektedir [19].

$$\bar{A}_j = \frac{\sum_{i=1}^n A_j}{n} \quad (1)$$

$\Sigma$  - Toplama sembolü

$A_j$  - j. özelliğin değeri

n - kelime sayısı

Word2Vec yönteminin aritmetik ortalama yaklaşımı için boyutun 100 alınarak oluşturulan örnek matrisi Tablo 2’de görülmektedir.

Medyan terimi, bir veri dizisinin küçükten büyüğe veya büyükten küçüğe sıralanmasından sonra bu dizinin tam ortasında bulunan değerdir. Eğer veri büyüklüğü tek sayılı ise medyan verilen bir veri değerine eşit olur. Eğer veri büyüklüğü çift sayılı ise medyan orta iki değer ortalaması olur. Medyan bulmak için basit bir algoritmaya göre sıralanmış veri değerlerinin kalan en küçük ve en büyük değerleri birer birer elimine edilir; veri sayısı tek ise en son kalan tek veri medyandır; eğer veri sayısı çift ise son kalan iki veri çiftinin ortalaması medyan olur [18]. Medyan için kullanılan formüller Eşitlik 2-3’te görülmektedir [19].

Tablo 2. Word2Vec-Aritmetik Ortalama (Boyut=100)  
(Word2Vec-Mean (Dimension=100))

Doküman No	Sıra	Kelime	Özellik [0]	Özellik [1]	...	Özellik [99]	Sınıf
1	1	Erva	0.569	0.124		0.745	
	2	ve	0.230	0.895		0.124	
	3	Emir	0.147	0.740		0.653	
	...	...	0.652	0.661		0.882	
	n	geldi	0.021	0.302		0.111	
<b>Aritmetik Ortalama</b>			$\frac{(\sum_1^n \text{özellik [0]})}{n}$	$\frac{(\sum_1^n \text{özellik [1]})}{n}$		$\frac{(\sum_1^n \text{özellik [99]})}{n}$	Negatif

Eğer özellik kelime sayısı tek ise ;

$$\text{Medyan} = \left(\frac{n+1}{n}\right). \text{terim} \quad (2)$$

Eğer kelime sayısı çiftse;

$$\text{Medyan} = \frac{\left(\frac{n}{2}\right). \text{terim} + \left(\frac{n}{2} + 1\right). \text{terim}}{2} \quad (3)$$

Maksimum, her bir özellik için tepe noktasını gösteren değerdir ve formülü Eşitlik 4'te görülmektedir.

$$\text{Max} = \max(A_j) \quad (4)$$

max = j. özelliğin maksimum değeri

$A_j$  = j. özelliğin değeri

Minimum- Maksimum ise her bir özellik için tepe ve taban noktalarının aritmetik ortalamasının alınması ile belirlenen değerdir. Min-Max formülü Eşitlik 5'te görülmektedir.

$$\text{MinMax} = \frac{(\max(A_j) + \min(A_j))}{2} \quad (5)$$

max = j. özelliğin maksimum değeri

min = j. özelliğin minimum değeri

Word2Vec ile model oluşturmak için kullanılan parametreler Tablo 3'te verilmektedir. Parametreler için varsayılan değerler kullanılmıştır.

Tablo 3. Word2Vec Parametreleri  
(Parameters of Word2Vec)

Parametre	Anlamı	Değeri
-size	Oluşturulacak vektör uzunluğu	100
-window	Bakılacak komşu sayısı	5
-min_count	Belirtilen frekanstan düşük olanları silmek için	5
-alpha	Öğrenme Oranı	0.025
-cbow	Metot seçimi (Skip-Gram=0,CBoW=1)	0 ya da 1

## 5. SONUÇLAR (RESULTS)

İki farklı veri seti (X ve Y) üzerinde Word2Vec algoritmasının iki farklı metodunun (CBoW ve Skip-Gram) model oluşturma sürelerini karşılaştırmak için bir çalışma gerçekleştirilmiştir. KNIME ortamında gerçekleştirilen çalışmada CBoW ve Skip-Gram metodlarının model oluşturma süreleri hesaplanmıştır. Hesaplamanın doğruluğunu artırmak için iki metot içinde çalışma çapraz geçirme (cross-validation) için k=10 olacak şekilde tekrarlanmış ve elde edilen sürelerin ortalamaları alınmıştır. Yapılan çalışmaya ait sonuçlar Tablo 4-5 ve Şekil 2-3'te görülmektedir.

Tablo 4. X veri seti için Model Süreleri  
(Model Time for X data set)

Yöntem	Süre (ms)	
	CBoW	Skip-Gram
Ortalama	811.23	827.69
Max	865.14	879.26
Min-Max	912.45	930.41
Medyan	1599.69	1610.42

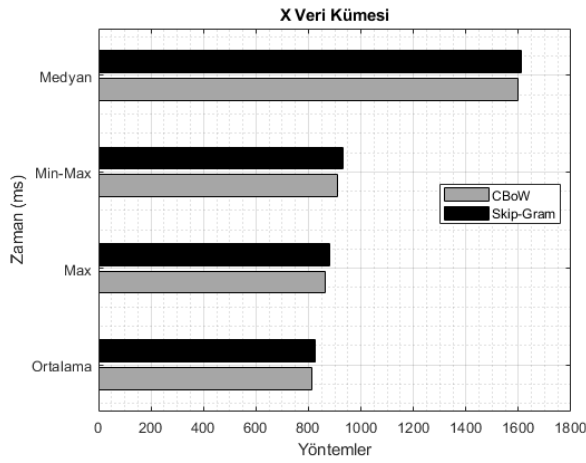
X veri seti için yapılan çalışmada CBoW, Skip-Grama göre daha az sürede model oluşturmayı başarmıştır.

Tablo 5. Y Veri seti için Model Süreleri  
(Model Time for Y data set)

Yöntem	Süre (ms)	
	CBoW	Skip-Gram
Ortalama	604.22	607.28
Max	626.47	629.44
Min-Max	675.12	677.91
Medyan	1055.38	1061.11

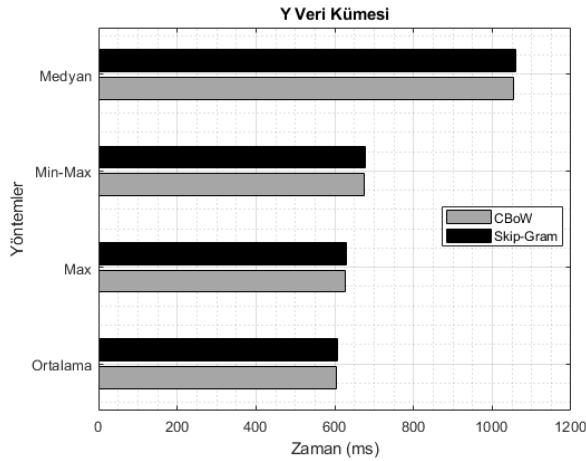
Y veri seti için yapılan çalışmada da CBoW, Skip-Gram'a göre daha az sürede model oluşturmayı başarmıştır.

Her iki veri seti içinde en kısa sürede modeli oluşturmayı Ortalama yöntemi gerçekleştirirken, en uzun sürede model oluşturmayı Medyan yöntemi gerçekleştirmiştir.



Şekil 2. X veri seti için Model Süreleri  
(Model time for X data set)

X veri kümesi üzerinde CBoW yöntemi Skip-Gram'a göre daha hızlı bir eğitim süreci tamamlamış yani daha kısa sürede model oluşturmayı başarmıştır.



Şekil 3. Y veri seti için Model Süreleri  
(Model time for Y data set)

Y veri kümesi üzerinde CBoW yöntemi ile Skip-Gram arasında çok az bir fark olsa da CBoW, Skip-Gram'a göre daha hızlı bir eğitim süreci tamamlamış yani daha kısa sürede model oluşturmayı başarmıştır.

## 6. TARTIŞMA (DISCUSSION)

Bu çalışmada iki farklı veri seti için son zamanlarda çok geniş kullanım alanına sahip Kelime vektörü algoritması olan Word2Vec için model oluşturma süreleri araştırılmıştır. Word2Vec'in iki önemli yöntemi olan CBoW ve Skip-Gram için model oluşturma süreleri karşılaştırılmak istenmiştir.

Deneylerde kullanılan veri kümeleri duygusal analiz çalışmasında kullanılmak için hazırlanmış veri kümeleridir. Veri kümeleri üzerinde çeşitli ön işlemler (durma kelimelerini temizleme, gövdeleme vb.) yapıldıktan sonra cümlelerin vektörlerinin çıkarılması

gerekmektedir. Word2Vec algoritmaları kelimelerin vektörünü çıkarabilmektedir. Cümlelerin belirtilen boyutta bir vektörünün çıkarılması için şimdiye kadar literatürde kullanılan yöntem oluşan her özelliğin aritmetik ortalamasını almaktır. Biz bu çalışmada kullanılan bu klasik yöntem ek olarak 3 farklı hesaplama (doküman vektörü oluşturma) daha önermekteyiz. Böylece sistemin model oluşturma sürelerinin, farklı hesaplama şekillerine vereceği tepkileri ölçmeyi hedefledik.

Önerdiğimiz 3 farklı doküman vektörü oluşturma yöntemleri ile normalde kullanılan ortalama yöntemini kullanarak iki farklı veri kümesinin eğitimi gerçekleştirilmiştir.

Gerçekleştirilen çalışmada 2 farklı kelime vektöründen CBoW, Skip-Grama göre daha kısa sürede eğitimi tamamlamış ve model oluşturmayı başarmıştır. Gerçekleştirdiğimiz çalışmanın doğruluğunu göstermek adına Mikolov ve ark. tarafından gerçekleştirilen çalışma verilebilir. Bu çalışmada CBoW ve Skip-Gramın model oluşturma süreleri karşılaştırılmış ve CBoW'un model oluşturma süresi daha kısa sürmüştür [14].

Önerilen 3 yöntem içinde Medyan en uzun sürede model oluşturmayı tamamlamıştır. En kısa sürede ise Max yöntemi modeli oluşturmuştur. İşlem karmaşıklığı açısından bakılırsa bu durumun gerçekleştirdiğimiz çalışmanın doğruluğunu gösterdiği açıktır. Max yöntemine göre daha fazla işlem gerektiren ama Medyan yönteminden daha az işleme gereksinimi olan Min-Max'ın iki yöntemin arasında değer göstermesi de çalışmanın sonuçlarını doğrular niteliktedir.

Çalışma sonuçları incelendiğinde gerek CBoW'un gerekse de Skip-Gram'ın model oluşturma sürelerinin aslında birbirine çok yakın olduğu görülmektedir. Veri seti boyutu arttıkça CBoW ile Skip-Gramın arasında model oluşturma sürelerinin arttığı Şekil 2-3'te görülmektedir. X veri kümesi 11332 örnek içerirken Y veri kümesi 7086 örnek içermektedir. Veri kümelerinin boyutlarının farklılığı Şekil 2 ve Şekil 3 arasındaki farklılığı açıklamak için yeterli görülmektedir.

Önerilen 3 farklı doküman oluşturma yöntemi ve klasik olarak kullanılan ortalama alma yöntemi ile iki farklı veri kümesinin eğitimi yapılmış ve model oluşturma süreleri karşılaştırılmıştır. Bundan sonraki çalışmamızda oluşan bu dört farklı doküman vektörü için test aşamasındaki doğruluk araştırılacaktır. Ayrıca farklı veri setleri ile çalıştırılacak modellerin gösterebileceği farklı bir durum olup olmadığı gözlenmeye çalışılacaktır.

## KAYNAKLAR (REFERENCES)

- [1] D. Zhang, H. Xu, Z. Su, Y. Xu, "Chinese comments sentiment classification based on word2vec and SVMperF", *Expert Systems with Applications*, 42(4), 1857-1863, 2015.
- [2] B. Dickinson, W. Hu, "Sentiment analysis of investor opinions on Twitter", *Social Networking*, 4(03), 62-71, 2015.

- [3] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification", **52nd Annual Meeting of the Association for Computational Linguistics**, Baltimore, Maryland, USA, 1555-1565, 23-25 Haziran, 2014.
- [4] J. Polpinij, N. Srikanjanapert, P. Sopon, "Word2Vec Approach for Sentiment Classification Relating to Hotel Reviews", **International Conference on Computing and Information Technology**, Bangkok, Thailand, 308-316, 6-7 Temmuz, 2017.
- [5] G. Şahin, "Turkish document classification based on Word2Vec and SVM classifier", **Signal Processing and Communications Applications Conference**, Antalya, Turkey, 1-4, 15-18 Mayıs, 2017.
- [6] B. Xue, C. Fu, Z. Shaobin, "A study on sentiment computing and classification of sina weibo with word2vec", **International Congress on Big Data**, Anchorage, AK, USA, 358-363, 27 Haziran-2 Temmuz, 2014.
- [7] M. Bilgin, İ.F. Şentürk, "Sentiment analysis on Twitter data with semi-supervised Doc2Vec", **International Conference on Computer Science and Engineering**, Antalya, Turkey, 661-666, 5-8 Ekim, 2017.
- [8] M. Bilgin, H. Köktaş, "Word2Vec Based Sentiment Analysis for Turkish Texts", **International Conference on Engineering Technologies**, Konya, Turkey, 106-109, 26-28 Ekim, 2017.
- [9] A. El Mahdaouy, E. Gaussier, S. O. El Alaoui, "Arabic Text Classification Based on Word and Document Embeddings", **International Conference on Advanced Intelligent Systems and Informatics**, Cairo, Egypt, 32-41, 24-26 Ekim, 2016.
- [10] Ö. Çoban, I. Karabey, "Music genre classification with word and document vectors", **Signal Processing and Communications Applications Conference**, Antalya, Turkey, 1-4, 15-18 Mayıs, 2017.
- [11] H. Ma, X. Wiang, J. Hou, Y. Lu, "Course recommendation based on semantic similarity analysis", **International Conference on Control Science and Systems Engineering**, Beijing, China, 638-641, 17-19 Ağustos, 2017.
- [12] M. Razzaghoori, S. Hedieh, K.J. Iman, "Question classification in Persian using word vectors and frequencies", *Cognitive Systems Research*, 47, 16-27, 2018.
- [13] H. Polat, M. Körpe, "TBMM Genel Kurul Tutanaklarından Yakın Anlamli Kavramların Çıkarılması", *Bilişim Teknolojileri Dergisi*, 11(3), 235-244, 2018.
- [14] T. Mikolov, K. Chen, G. Corrado, J. Dean, "Efficient Estimation of Word Representations in Vector Space", **International Conference on Learning Representations**, Arizona, USA, 1-12, 2-4 Mayıs, 2013.
- [15] Internet: X Data Set, <https://www.dropbox.com/s/aji68llxmtcuu5l/data.zip>, 05.09.2018.
- [16] Internet: Y Data Set, <https://www.kaggle.com/c/si650winter11/data>, 05.09.2018
- [17] M.R. Berthold, N. Cebron, F. Dill, T.R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, B. Wiswedel, "KNIME-the Konstanz information miner: version 2.0 and beyond", *ACM SIGKDD explorations Newsletter*, 11(1), 26-31, 2009.
- [18] Internet: Mean, <https://www.wikipedia.com>, 08.09.2018.
- [19] Internet: Mean-Median, <https://byjus.com/mean-median-mode-formula>, 08.09.2018.