Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 34(2), ss. 181-195 Haziran 2019
Çukurova University Journal of the Faculty of Engineering and Architecture, 34(2), pp. 181-195, June 2019

Kelime Temsil Yöntemleri ile Kelime Benzerliklerinin İncelenmesi

Murat AYDOĞAN[*1], Ali KARCI[2]

[1]*Bingöl Üniversitesi, Genç Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Bingöl*
[2]*İnönü Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Malatya*

Geliş tarihi: 09.01.2019 Kabul tarihi: 28.06.2019

Öz

Günümüzde büyük veri alanında meydana gelen gelişmelerle birlikte günlük işlenebilir durumda olan veri miktarı oldukça büyük boyutlara ulaşmıştır. Bu verilerin çok büyük bir kısmının metin (text) verilerinden oluşması, metin işleme alanında yapılan çalışmaları oldukça önemli ve popüler bir hale getirmiştir. Ancak bu alanda yapılan çalışmalar incelendiğinde başta İngilizce olmak üzere birçok dünya diline yönelik çeşitli çalışmalar yapılırken, Türkçeye özgü yapılan çalışmaların istenilen sayıda olmadığı görülmüştür. Bu nedenle bu çalışma için, python ortamı kütüphanelerinden biri olan Beautiful Soup kütüphanesi kullanılarak Türkçe metinlerden oluşan büyük bir derlem oluşturulmuştur. Bu çalışmada, kelimeleri vektör uzayında her bir kelimenin bir vektörle temsil edildiği yeni bir yaklaşım olan Word2Vec modeli algoritmalarından CBOW ve Skip-Gram algoritmaları ile Glove modeli kullanılmıştır. Oluşturulan derlem üzerinde Word2Vec yöntemi ile Türkçe kelimelerden oluşan ve bu kelimeler arasındaki anlamsal ilişkileri tespit etmeye çalışan bir model geliştirilmiş ve diğer modeller ile başarımı ve eğitim süreleri kıyaslanmıştır. Ayrıca çalışmanın bir diğer katkısı ise modelin performansını artırmak için Türkçe için etkisiz kelimeler listesi oluşturulmasıdır. Geliştirilen bu model ile özellikle Türkçe metin sınıflandırma problemlerinde daha yüksek bir sınıflandırma başarımının yakalanması hedeflenmektedir. Bu çalışma kapsamında oluşturulan model analiz edilip yakın anlamlı kelimeler incelendiğinde oldukça başarılı performans gösterdiği tespit edilmiştir. Veriseti ve kelime vektörleri Türkçe çalışmalara katkı sağlamak için erişime açık olarak paylaşılacaktır.

Anahtar Kelimeler: Doğal dil işleme, Kelime temsili, Word2Vec, GloVe

Investigation of Word Similarities with Word Embedding Methods

Abstract

Nowadays, the amount of data that can be processed daily has reached quite big dimensions with the developments in the big data. The fact that a large part of this data consists of text data has made the studies in the field of text processing very important and popular. However, when the studies in this area are examined, it has been observed that while various studies are carried out for many World languages, especially English, there are no desired number of studies conducted in Turkish. Therefore, a large corpus of Turkish texts was created using the Beautiful Soup library, one of the python environment libraries. In this study, CBOW and Skip-Gram algorithms from Word2Vec model algorithms and Glove model were used where each word is represented with a vector in the vector space. In this study, a model which consists of Word2Vec method and Turkish words and tries to detect the semantic relations between these

[*]Sorumlu yazar (Corresponding author): Murat AYDOĞAN, *maydogan@bingol.edu.tr*

words has been developed and the performance and training times have been compared with other models. In addition, another contribution of this study to improve the performance of the model to create a list of stop words for Turkish. With this model, it is aimed to achieve a higher classification performance especially for Turkish text classification problems. After analyzing the model formed within the scope of this study, it was detected that it showed a very successful performance when close words were examined. The dataset and word vectors will be shared with the public to provide contributions to Turkish studies.

Keywords: Natural language processing, Word embedding, Word2Vec, GloVe

1. GİRİŞ

Günümüzde internet dünyasının yaygın kullanımına bağlı olarak yaşanan gelişmelerle birlikte üretilen veri miktarı hızla artmaktadır. "Büyük Veri" olarak adlandırılan bu büyük veri kümeleriyle ilgili olarak IBM'e göre bir günde yaklaşık 2,5 kentilyon veri üretilmektedir. Üretilen bu verinin çok büyük kısmını ise (%90-95) metin verileri oluşturmaktadır. Forbes verilerine göre 2 yıl içerisinde üretilen metin verileri 40 zettabayt boyutuna ulaşacaktır. Ayrıca Google her saniyede 40 binden fazla arama sonucunu işlerken bu durum yaklaşık olarak günde 3,5 milyar aramaya tekabül etmektedir [1].

Bu nedenle metin işleme çalışmaları son dönemde oldukça önemli hale gelmiş ve yapılan çalışmalar incelendiğinde oldukça popüler olmuştur. Metin işleme alanında şu alanlarda çok sayıda çalışmalar yapılmaktadır.

- Yazım yanlışlarının düzeltilmesi
- Bir metnin özetini çıkarma
- Metnin içerdiği bilgiyi çıkarma
- Bilgiye erişim
- Metni anlama
- Bilgisayarla sesli etkileşim
- Konuşmayı anlama (konuşmayı metne dönüştürme)
- Soru yanıt dizgeleri
- Doğal diller arası çeviri [2].

Bir derlem içerisindeki metinlerin nasıl temsil edileceği metin işleme çalışmalarının en kritik noktalarından biridir. En basit tanımıyla, metinlerin sayısal ifadelere dönüştürülmesine kelime temsili (word embedding) denilmektedir. Aynı metin farklı şekilde farklı sayısal değerlere dönüştürülebilmektedir [3].

Kelime temsil yöntemleri, Frekans Bazlı Temsil ve Tahmin Bazlı temsil olmak üzere ikiye ayrılmaktadır. Daha geleneksel yöntemler olarak tanımlanan Frekans bazlı kelime temsilleri, dokümanlar içerisinde bulunan kelimeler ve bu kelimelerin frekanslarının tespit edilmesi prensibine dayanmaktadır [4].

Frekans bazlı kelime temsiline dayalı olarak geliştirilmiş ve en çok tercih edilen yöntem Bag of Words metodudur. Bu yönteme göre, dokuman içerisindeki her cümle benzersiz (unique) kelimelere ayrılarak, benzersiz kelime boyutunda bir matrise dönüştürülür. Matrisin sütunları dokuman içerisindeki kelimelerden oluşurken (N), satırları ise dokuman sayısından meydana gelir (D). Sonuç olarak tüm derlem DxN boyutunda bir matris olarak temsil edilmiş olur [5]. Frekans bazlı temsil yöntemlerinden bir diğeri ise TF-IDF (Term Frequency—Inverse Document Frequency) yöntemidir. Term Frequency, bir doküman içerisinde geçen terim frekanslarını tespit etmek için kullanılır. Inverse Document Frequency yöntemi ise birden fazla dokümanda kelimenin geçme sayısını bularak bu kelimenin terim olup olmadığını bağlaç vb (stop words-etkisiz kelime-) olduğu anlamaya çalışır [6]. Bu çalışmada da oluşturulan modelin verimliliğini artırmak için Türkçe kelime stop words listesine ihtiyaç duyulmuş ancak bu konuda tatmin edici bir liste bulunmadığı için çalışmanın bir katkısı olarak Inverse Document Frequency yöntemi kullanılarak bir stop words (etkisiz kelimeler) listesi oluşturulmuş bununla ilgili bilgiler çalışmanın ilerleyen kısımlarında verilmiştir. Co-Occurrence

Matrix ise benzer kelimelerin cümle içerisinde birlikte geçme eğiliminde olması ilkesine dayanmaktadır. Bu yöntemde oluşturulan matrisin hem satırlarında hem sütunlarında benzersiz (unique) kelimeler yer almaktadır. Matrisin hücrelerini ise, satır ve sütunda yer alan kelimelerin birlikte geçme frekanslarını içermektedir [7].

Frekans bazlı yöntemlerin en önemli iki dezavantajı vardır. Birincisi, kullanılan yönteme göre satır ve/veya sütunlarda benzersiz kelimeler bulunduğu için kelime sayısı boyutunda matrisler meydana gelir ve bunların birçoğunun değeri 0 olacağından seyrek matris (sparse matrix) denilen durum ortaya çıkmaktadır. Bir diğer önemli dezavantajı ise, kelimeler arasındaki anlamsal yakınlıkların tespit edilememesi durumudur [8].

Kelime temsil yöntemlerinden bir diğeri olan tahmin bazlı kelime temsil yöntemi ailesinden olan Mikolov ve arkadaşları [9] tarafından 2013 yılında geliştirilmiş olan temelinde yapay sinir ağı ile kelimelerin eğitilmesi ilkesine dayananWord2Vec modelidir. Bu model giriş olarak alınan kelimelere dayanarak hedef kelimeyi tahmin etme prensibine dayalıdır. Word2Vec modeli CBOW (Continous Bag of Words) ve Skip-gram olarak adlandırılan iki farklı algoritmadan oluşmaktadır.

Tahmin bazlı bir kelime temsil yöntemi olan Word2Vec yöntemi, girdi (input) olarak büyük bir derlem içerisindeki benzersiz kelimeleri alır ve belirlenen bir temsil vektörü boyutunda matris oluşturulmaktadır. Word2Vec modeli her defasında dokümandaki bir cümleyi pencere (window) denilen bir yapı içerisinde kaydırarak taramakta ve penceredeki hedef kelimeye göre genellikle çok sayıda boyuttan oluşan bir vektörü çıktı olarak üretmektedir [10].

Word2Vec yönteminin genel olarak birbirine benzeyen CBOW (Continuous Bag of words) ve Skip–Gram olmak üzere iki tür alt yöntemi vardır. Bu iki modelin temel olarak farklılıkları ise girdi ve çıktıları alma yöntemlerine dayanmaktadır [11].

Bu çalışmada, Türkçe içeriklerden oluşmuş bir derlem üzerinde Word2Vec yöntemi ile her bir kelime için kelime vektörleri oluşturularak kelime benzerlikleri ve anlam benzerlikleri incelenmiş, ayrıca bulunan sonuçlar Glove yöntemi ile kıyaslanmıştır.

2. ÖNCEKİ ÇALIŞMALAR

Kelime temsili (word embedding), kelimelerin harflerden vektör biçiminde sayılara dönüştürülmesi olarak açıklanabilir. Kelime temsil metotları ise bu dönüştürme işleminde tercih edilen tekniklerdir. Literatür incelendiğinde, özellikle sinir ağlarında son dönemlerde yaşanan gelişmelerle birlikte kelime temsilinin metin işleme çalışmalarında en kritik noktalardan bir olduğu kabul edilmektedir [12].

Amasyalı ve arkadaşları [13] çalışmalarında Türkçe metin sınıflandırma için 6 Türkçe veri seti üzerinde temsil yöntemlerinin performanslarını karşılaştırmışlardır. Çalışmalarında uyguladıkları kelime temsil metotları arasında en başarılı performansı n-gram yöntemi göstermiştir.

Arabacı ve arkadaşları [14] ise kelime temsil yöntemlerini kullanarak benzer cümle tespiti yapmışlardır. Çalışmada Word2Vec yöntemi ile birlikte Fisher kodlama kullanılmıştır. Çalışma sonunda bu teknikle en yüksek başarım değeri elde edilmiştir.

Esen ve Özkan ise TBMM tutanaklarına göre kelime temsil yöntemleri ile parti bağdaşıklığı açısından analiz çalışması yapmışlardır. Bunun için siyasi metinler otomatik olarak analiz edilmiş ve buna göre parti bağdaşıklığı ölçülmüştür [15].

Amasyalı ve arkadaşları [16] çalışmalarında Türkçe duygu durum analizi için metinlerin kelime, anlamsal ve karakter tabanlı temsilleri karşılaştırmışlardır. Türkçe veriseti üzerinde yaptıkları çalışmalarında karakter tabanlı temsiller daha başarılı olmuştur.

Ayata ve arkadaşları ise Türkçe tweetlerinin içeriğine göre duygu analizi yapmışlardır. Kelimeler vektörleştirilerek temsil edilmiş, makine öğrenmesi algoritmaları ile sınıflandırma işlemi yapılmıştır [17].

Keleş ve Özel [18] ise çalışmalarında Türkçe doküman arasındaki benzerlikleri belirleyebilmek için çeşitli uzaklık ölçütlerinin performanslarını karşılaştırarak en uygun yöntemi belirlemeye çalışmışlardır. Çalışma sonunda kosinüs benzerliği yönteminin daha başarılı performans göstermiştir.

Yapılan çalışmalar incelendiğinde genellikle yapılan çalışmaların bir kısmında geleneksel kelime temsil yöntemlerinin kullanıldığı görülürken, sinir ağı temelli modern temsil yöntemlerinin kullanıldığı çalışmalarda ise kullanılan veri setlerinin belli bir konseptten (TBMM tutanakları, siyasi metinler, olumlu-olumsuz müşteri/kullanıcı yorumları vs.) seçildiği görülmüştür.

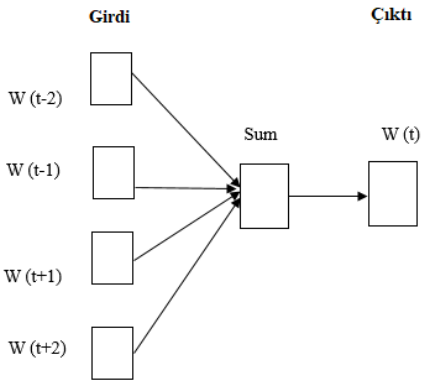
Bu durumun kelimeler arasında ki anlamsal ilişkilerin tam olarak ortaya çıkarılmasında bir dezavantaj oluşturacağı düşünüldüğünden bu çalışmada, çeşitli konu başlıklarından seçilen metinlerle bilinen en büyük etiketsiz Türkçe veri seti oluşturulmuştur. Bu veri seti üzerinde kelime temsil çalışmaları yapılırken aynı zamanda kelime temsil yöntemlerinin de performansları değerlendirilmiştir. Bu veri seti ve kelime vektörleri Türkçe doğal dil işleme çalışmalarına katkı sunmak amacıyla araştırmacılarla paylaşılacaktır. Çalışmanın bir diğer katkısı ise ön işleme çalışmalarına fayda sağlamak amacıyla etkisiz kelimeler listesi oluşturulmuştur. Ayrıca, özellikle Word2Vec metodu ve sistematiği hakkında Türkçe kaynak eksikliği görüldüğünden bu konu da detaylı olarak anlatılmaya çalışılmıştır.

3. YÖNTEM

3.1. CBOW Model (Continuous Bag of Words)

CBOW modelinde pencere boyutu (window size) merkezinde olmayan kelimeler girdi olarak alınıp, merkezinde olan kelimeler çıktı olarak tahmin edilmeye çalışılmaktadır. Bu işlem cümle bitimine kadar devam etmektedir [9].

Bu durum Şekil 1'de gösterilmeye çalışılmıştır. Burada w(t) ile gösterilen değer, cümlenin merkezinde bulunan ve tahmin edilmek istenen çıktı değeriyken, w(t-2)…..w(t+2) ile gösterilen değerler ise tercih edilen pencere boyutuna göre (window_size) merkezde olmayan çıktı değerleridir.



Şekil 1. CBOW modelinin yapısı [9].

Şekil 1'de açıklanan CBOW modelinin uygulaması dünyaca ünlü yapay zekâ araştırmacılarından biri olan Andrew Ng'nin söylediği bir söz üzerinden açıklanmak istenirse

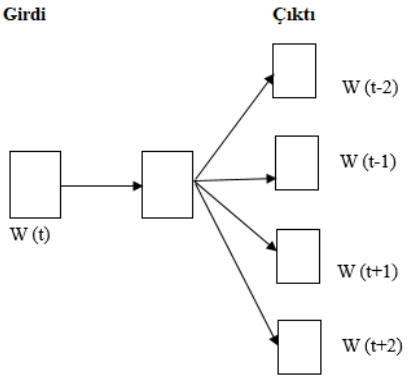
Örnek Cümle: "Yapay zekâ yeni elektriktir."
Bu cümleyi girdi olarak alan ve pencere boyutu=1 olan CBOW modeli şu şekilde çalışmaktadır. Önce "yapay" kelimesini pencerenin merkezine oturur, sonra sağındaki ve solundaki 1'er kelimeyi ayrı ayrı girdi olarak alıp (window_size=1) pencerenin merkezinde bulunan "yapay" kelimesisinir ağı modeli ile tahmin edilmeye çalışılmaktadır. Sonra pencere 1 sağa kaydırılarak, bu sefer de pencerenin merkezine "zeka" kelimesi gelmektedir. Çizelge 1'de cümlenin adım adım CBOW modeline göre işletilmesi gösterilmiştir. Kırmızı renkle yazılmış kelimeler çıktıyı, mavi renkli kelimeler ise girdiyi ifade etmektedir.

Çizelge 1. CBOW modeli

	Cümle (window_size=1)				GİRDİ	ÇIKTI
1	Yapay	zeka	yeni	elektriktir	(zeka)	(yapay)
2	Yapay	zeka	yeni	elektriktir	(yapay) (yeni)	(zeka) (zeka)
3	Yapay	zeka	yeni	elektriktir	(zeka) (elektriktir)	(yeni) (yeni)
4	Yapay	zeka	yeni	elektriktir	(yeni)	(elektriktir)

3.2. Skip – Gram Model

Skip-Gram modelinde ise CBOW modelinin tersi şeklinde bir işleyişe sahiptir. Merkezdeki kelime girdi olarak alınıp merkezde olmayan kelimeler çıktı olarak tahmin edilmeye çalışılır. Bu işlem cümle bitene kadar devam eder. Bu durum Şekil 2'de gösterilmiştir [10].



Şekil 2. Skip – Gram modelinin yapısı [10]

CBOW modelinde uygulaması yapılan örnek cümle skip-gram modelinde uygulanması ise Şekil 3'teki gibi olmaktadır.

Şekil 3'te anlatılan skip-gram algoritmasının optimize edilecek hata fonksiyonu ise Eeşitlik 1'de gösterilmiştir. Bu denklemde pencere büyüklüğü *m* ile ifade edilmiştir. Denklem, derlem içerisindeki tüm kelimeler için *t* pozisyonundaki bir kelimenin solunda ve sağında bulunan *m* (pencere boyutu) kadar kelimenin tahmin edilmesine dayanmaktadır.

$$J=-\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\le j\le m}\log P(w_{t+j}\mid w_t) \qquad (1)$$

t=1…..T ile ifade edildiği gibi bu işlem derlem içerisindeki tüm kelimeler için yapılarak, her kelimenin etrafındaki kelimeleri yüksek olasılıkla tahmin edebilen vektörler oluşturulmaktadır. Olasılıkların hesaplandığı kısım ise Eşitlik 2 ile daha ayrıntılı olarak gösterilmiştir.

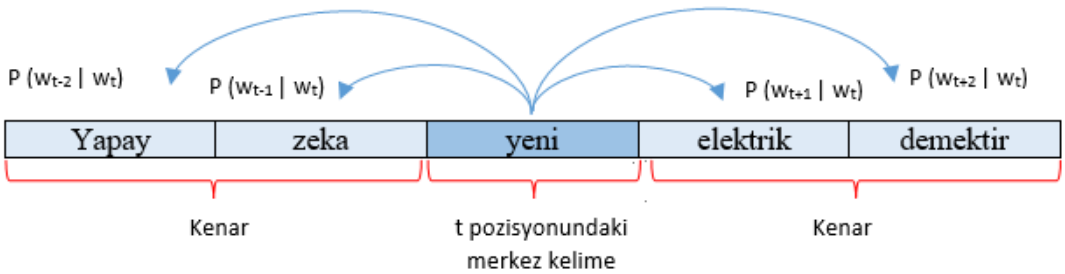
$$P(w_{t+j}\mid w_t)=\frac{\exp(u_{w_{t+j}}^{T}v_{w_t})}{\sum_{w=1}^{v}\exp(u_w^{T}v_{w_t})} \qquad (2)$$

Burada dikkat edilmesi gereken husus büyüklüğü *m* olan bir pencere içerisindeki kelimenin tahmin edilmesi işleminde, her kelimenin biri söz konusu kelimenin merkezde yani w_t durumunda iken bir diğeri ise kelime pencere içerisinde kenar kelime durumunda yani w_{t+j} durumunda olmak üzere iki adet vektörünün olmasıdır. Her kelime için kenar kelimelere göre olasılıkların nasıl hesaplandığını gösteren 2 numaralı formülde *v* sembolü o kelimenin merkez kelime iken vektör değerini, *u* ise kelimenin pencere içerisinde kenar kelime durumundayken vektörel değerini ifade etmektedir.

Pencere boyutunun 2 olduğu bir olasılık hesaplama işlemi örneği aşağıdaki gibidir. Bölme işleminin paydasında ise, bu işlemin derlem içerisinde yer alan tüm kelimeler için yapılarak değerlerin toplandığı görülmektedir.

$$p(\text{yapay}|\text{yeni})=\frac{\exp(u_{\text{yapay}}^{T}v_{\text{yeni}})}{\sum_{w=1}^{v}\exp(u_w^{T}v_{\text{yeni}})}$$

$$p(\text{zeka}|\text{yeni})=\frac{\exp(u_{\text{zeka}}^{T}v_{\text{yeni}})}{\sum_{w=1}^{v}\exp(u_w^{T}v_{\text{yeni}})} \qquad (1)$$



Şekil 3. Skip-Gram yönteminin uygulanması

Çizelge 2. Skip-Gram modeli

	Cümle (window_size=1)				GİRDİ	ÇIKTI
1	Yapay	zeka	yeni	elektriktir	(yapay)	(zeka)
2	Yapay	zeka	yeni	elektriktir	(zeka) (zeka)	(yapay) (yeni)
3	Yapay	zeka	yeni	elektriktir	(yeni) (yeni)	(zeka) (elektriktir)
4	Yapay	zeka	yeni	elektriktir	(elektriktir)	(yeni)

Çizelge 1 ve Çizelge 2 incelendiğinde Skip-Gram modelinin esasen CBOW modeli ile çok benzer olduğu daha öncede ifade edildiği gibi temel farklılığın kelimeleri girdi olarak alış şekillerinden kaynaklandığı net olarak ortaya çıkmaktadır.

3.3. GloVe

Glove, adını "Global Vectors for Word Representation" kelimelerinin baş harflerinden alan bir diğer kelime temsil yöntemidir. Doğal dil işleme alanında Word2Vec yönteminden sonra en çok kullanılan yöntemdir. Stanford üniversitesinde Pennington ve arkadaşları tarafından geliştirilmiştir [19].

Bir derlem üzerinde yakın anlamlı kelimelerin tespit edilmesinde Word2Vec yöntemine göre daha az başarılıdır. Glove, global kelime-kelime sayıları üzerinde eğitilen ve böylece istatistiklerin daha etkili olarak kullanılmasına olanak sağlayan bir yöntemdir. Glove modeli, analojik veri seti üzerinde %75 doğruluk oranına sahip bir kelime uzay modeli üretmektedir. Bu da Word2Vec yönteminin analoji tespitlerinde daha başarılı olduğunu göstermektedir. Glove yöntemine göre optimize edilecek hata fonksiyonu Eşitlik 3 ile gösterilmiştir.

$$J=\frac{1}{2}\sum_{i,j=1}^{W} f(P_{ij})\,(u_i^T v_j - \log P_{ij})^2 \qquad (3)$$

3 numaralı eşitliğe göre *u* ve *v* simgeleri global kelime-kelime sayıları ile oluşturulan matrisin satır ve sütun değerlerini göstermektedir. Eşitlik 3'te ki $f(P_{ij})$ tanımlanmış ağırlık fonksiyonudur. W ise oluşturulan sözlüğün boyutudur [19].

4. MATERYAL

Daha öncede bahsedildiği üzere metin işleme hatta daha üst kapsamıyla doğal dil işleme alanında yapılan çalışmalar, büyük veri kavramının ortaya çıkmasıyla hatta bu verilerin çok büyük bir kısmının metin (text) verilerinden oluşması nedeniyle çok büyük önem kazanmış ve son yıllarda oldukça popüler çalışma alanları haline gelmiştir. Ancak bu konuda Türkçe diline özgü yapılan çalışmaların sayısına bakıldığında sayılarının yetersiz olduğu görülmüştür. Bu nedenle Word2Vec yöntemiyle Türkçe içeriklerden oluşan veriler üzerinde bir çalışma yapılmıştır. Çalışmada kullanılan Türkçe derlem (korpus) iki bölümden oluşmaktadır.

Hazırlanan korpusun ilk bölümü, internetten faydalanarak bazı sorgularla veri çekmeye yarayan beautiful soup adlı bir python kütüphanesi ile oluşturulmuştur. Örneğin "teknoloji", "eğitim", "tarih", "türkiye", "ekonomi", "spor" , "siyaset" gibi kelimeler ile sorgu yapılmıştır. Bu kısım internetten edinildiği için bazı kelimelerde yazım hataları olabilmektedir.

Korpusun ikinci bölümü ise Türkiye Büyük Millet Meclisi tutanakları ve Wikipedia Türkçe makaleleri kullanılarak oluşturulmuştur. Bu bölümde ise, herhangi bir yazım hatası bulunmamaktadır.

Bu çalışma kapsamında, toplamda 22.090.767 satırdan ve 10.562.752.820 adet token haline gelmiş kelimeden oluşan yaklaşık 60 GB boyutunda bu alanda yapılmış çalışmalar içerisinde ki en büyük derlemlerden biri oluşturulmuştur.

4.1. Ön İşlem Çalışmaları

Metin dosyası içerisindeki metinler okunarak, nokta işareti referans alınarak metinler cümlelere bölünmüştür.

Cümlelerden harfler dışındaki tüm karakterler (sayılar, semboller, noktalama işaretleri) temizlenmiştir.

Cümleler ise, aralarındaki boşluklar dikkate alınarak kelimelere ayrılmıştır.

Tokenlara ayrılmış kelimeler, zemberek ortamı kullanılarak köklerine ayrılmıştır.

Kelimeler içerisinde stop words olarak bilinen etkisiz kelimeler elenmiştir. Ancak diğer diller için bu kelime listeleri daha önceden oluşturulmuşken Türkçeye özgü bu şekilde tatmin edici bir çalışma yapılmadığı için bu bölümde çalışmanın bir katkısı olarak etkisiz kelimeler listesi oluşturulmuştur.

Etkisiz kelimeler listesinin oluşturulmasında çalışma için hazırlanmış derlemin ikinci bölümü kullanılmıştır. Bunun nedeni derlemin bu bölümünün daha küçük olması ve hiçbir yazım hatasının bulunmamasıdır.

Etkisiz kelimeler listesinin oluşturulmasında ilk bölümde değinilen TF-IDF yöntemi kullanılmıştır [20].

Eşitlikte 4'te $tf_{i,j}$ ile terim sıklığı yani TF değeri hesaplanmaktadır. Bu değer eşitlikte de görüldüğü üzere *TF=kelimenin doküman da geçme sayısı/dokümandaki kelime sayısı* ile hesaplanmaktadır.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}} \quad (4)$$

Eşitlik 5 kullanılarak IDF değeri hesaplanmıştır. *IDF = log (toplam doküman sayısı/kelimeyi içeren doküman sayısı)*

$$idf\,(w) = \log\left(\frac{N}{df_t}\right) \quad (5)$$

TF-IDF matrisinin ağırlıklarının oluşturulması için ise Eşitlik 6 kullanılmıştır.

$$w = tf * idf \quad (6)$$

Çizelge 3 yorumlanırsa, derlem içerisinde en yaygın kelimeler frekansları göz önüne alınarak hesaplanmıştır. TF-IDF değerleri tablodaki gibi hesaplandığında ağırlık değeri ne kadar 0'a yakınsa o kelimenin derlem içerisinde o kadar yaygın (common word) bir kelime olduğu anlaşılmaktadır. Bu da derlem içerisinde kelimenin çok spesifik bir kelime olmadığını ortaya koymaktadır. Bu nedenle TF-IDF skoru 0 ile 0,1 aralığında olan kelimelerden etkisiz 250 kelime belirlenmiştir.

Daha önce de belirtildiği gibi bu çalışma için hazırlanan derlem içerisinde yaklaşık 10,5 milyar token kelime bulunmaktaydı. Etkisiz kelimeler listesinin oluşturulması ve uygulanması ile birlikte 848.908.720 adet kelime etkisiz kelimeler listesi içerisinde yer alan kelimelerden oluştuğu için model için hazırlanan derlem içerisinden elenerek temizlenmiştir.

Çizelge 3. TF-IDF ağırlık matrisleri

KELİME	TF	IDF	AĞIRLIK
'için'	126215/4742527= 0,026	log(387179 /126215) = 0,486	TF*IDF =0,012
'de'	98689/4742527= 0,020	log(387179 /98689) = 0,593	TF*IDF =0,011
'da'	68652/4742527= 0,014	log(387179 /68652) = 0,751	TF*IDF =0,010
'bu'	44037/4742527= 0,009	log(387179 /44037) = 0,944	TF*IDF =0,008
've'	26249/4742527= 0,005	log(387179 /26249) = 1,168	TF*IDF =0,005

4.2. Model Oluşturulması

Bu çalışmada daha öncede bahsedildiği gibi kelimelerin analizlerinin yapılmak üzere sayısallaştırılması maksadıyla Google tarafından Mikolov ve arkadaşları [9] tarafından geliştirilen Word2Vec yöntemi kullanılmıştır. Word2Vec yönteminde parametre seçimleri modelin performansını önemli ölçüde etkilemektedir. Yöntemde kullanılan bazı parametreler ve açıklamaları ile bu çalışmada geliştirilen modelde tercih edilen parametre değerleri Çizelge 4'te verilmiştir. Kullanılan parametre değerleri literatürde yapılan çalışmalar incelenerek ve kullanılan değerlerin sonunda elde edilen sonuçlar deneme yanılma ile gözlemlenerek en uygun değer değerler tercih edilmiştir [21].

Deneysel çalışmalarda, Python 3 programlama dili ve Anaconda ortamı ayrıca Microsoft Windows Server 2012 R2 işletim sistemi ile Intel Xeon E5-2630 2.20 CPU ve 64 GB bellek kullanılmıştır.

Çizelge 4. Word2Vec yönteminde kullanılan bazı parametreler

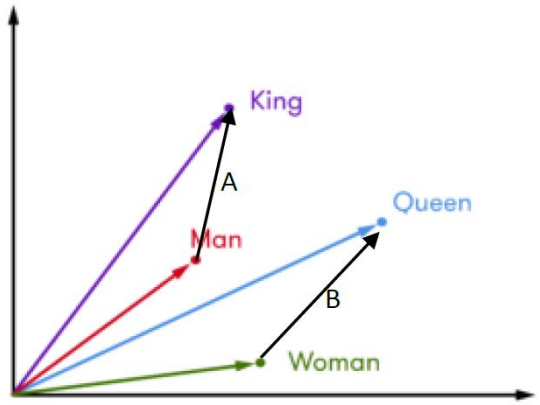
Parametre	Alabileceği Değer Aralığı	Açıklama	Kullanılan Değer
sentences	Model için oluşturulmuş, ön işlemden geçirilmiş kelimeler listesidir.	Modelin eğitilmesi için kullanılan veri kümesidir.	Derlem
size	Opsiyonel olarak integer değer alır.	Kelime vektörlerinin boyutudur.	300
window	Opsiyonel olarak integer değer alır.	Bir cümle içindeki mevcut ve tahmin edilen kelime arasındaki maksimum mesafedir.	10
min_count	Opsiyonel olarak integer değer alır.	Parametre değerinden daha düşük frekanstaki tüm kelimeler yok sayılır.	10
workers	Opsiyonel olarak integer değer alır.	Modeli eğitmek için çalışan thread sayısıdır. Çok çekirdekli makinelerle daha hızlı eğitim gerçekleşir.	10
sg	0 ve 1 değerlerini alır.	Skip-gram yöntemiyle eğitim için 1, CBOW yöntemi için 0 değeri kullanılır.	Çalışmada iki yöntemde kullanılmıştır.(0 ve 1)
alpha	Opsiyonel olarak float değer alır.	Başlangıç öğrenme oranıdır.	0,025

5. BULGULAR

Bu çalışmada, python programlama diline ait BeautifulSoup kütüphanesi kullanılarak oluşturulan derlem üzerinde ön işlem çalışmaları yapılarak Word2Vec modeli uygulanmıştır. Şekil 4'te görüldüğü gibi Mikolov ve arkadaşlarına göre, "king" ve "man" kelimelerinin vektörleri arasındaki uzaklık "queen" ve "woman" kelimeleri arasındaki uzaklığa eşit olmalıdır. Başka bir deyişle A vektörü B vektörüne eşit olmalıdır denilebilir. Mikolov ve arkadaşları bu eşitliği şöyle tanımlamışlardır [11].

vector (“king”) – vector ("man") ≈ vector (“queen”) – vector ("woman")

Bu denklem Şekil 4’te görselleştirilmiştir. Ayrıca bu gösterim yorumlandığında “king” kelimesinin vektörel değerinden “man” kelimesi çıkarılıp “woman” kelimesi eklenirse “queen” kelimesinin cevap olarak alınması beklenmektedir.



Şekil 4. Mikolov tarafından tanımlanmış kelime vektörlerinin gösterimi [11]

Bu çalışmada Word2Vec modeline ait CBOW ve Skip-gram yöntemleri Türkçe metinler üzerinde kıyaslanarak modelin performansı incelenmiştir. Word2Vec modelinde kelime vektörleri arasındaki benzerlik genellikle kosinüs benzerliği kullanılarak tespit edilmektedir. Bu çalışmada kosinüs uzaklık formülü ile birlikte İbn-i Sina (Öklid) uzaklığı da kullanılarak uzaklık formüllerinin de yakın anlamlı kelimelerin tespitinde nasıl bir rol oynadığı incelenmiştir.

Çalışmada, oluşturulan Türkçe derlem üzerinde önişlem aşamalarının ardından Word2Vec modeli ile uygun parametreler kullanılarak eğitim süreci tamamlanmıştır. Eğitim işlemi bittikten sonra derlem içerisindeki her bir benzersiz kelime için kelime vektörleri oluşturulmuştur.

Kelimeler arasındaki benzerlik ve yakın anlamların incelenmesinde bu kelime vektörleri kullanılmıştır. Çizelge 5 incelendiğinde örneklerde verilen kelimelerin vektörel karşılığı bulunarak, bu değerin kelime listesinde yer alan diğer benzersiz kelimelerin vektörel değerleri arasındaki kosinüs uzaklıkları incelenmiştir. Çizelge 5’te örnek kelimeler üzerinde bu kelimeye en yakın 5 kelime CBOW, Skip-Gram ve Glove yöntemlerine göre listelenmiştir.

İlk sütunda yakın anlamlı kelimelerin araştırıldığı kelime vektörü verilmiştir. Çizelge 5’te yer alan sonuçlar ilk olarak CBOW yöntemine göre incelendiğinde; “Pazartesi” kelimesi ile yakın anlamlı kelimelerin tespiti için pazartesi kelimesinin vektörüne en yakın vektörler sıralandığında listelenen kelimelerin Salı, Perşembe, Çarşamba şeklinde yine haftanın günlerinin bulunduğu görülmektedir. Yani bu durum modelin pazartesi kelimesini haftanın günlerinden biri olduğunu anladığı ve yakın anlamlı olarak haftanın diğer günlerini bulduğu şeklinde yorumlanabilir. Yine benzer şekilde “ocak” kelimesi ile yakın anlamlı kelimeler olarak yılın diğer ayları listelenmiş, “ankara” kelimesinde diğer önemli iller, “beş” kelimesinde diğer rakamlar benzer kelimeler olarak ortaya çıkmıştır.

Çizelge 5’te yer alan son örnek incelendiğinde ise “mi” kelimesine benzer kelimeler sorgulanmış cevap olarak mu?, mudur? vb. soru ekleri bulunmuştur. Böylece modelin “mi” sözcüğünün bir soru eki olduğunu tespit ettiği ve bu model kullanılarak bazı gramer tabanlı işlemlerinde yapılabileceği görülmektedir. CBOW algoritmasına göre sonuçların başarılı olduğu görülmektedir.

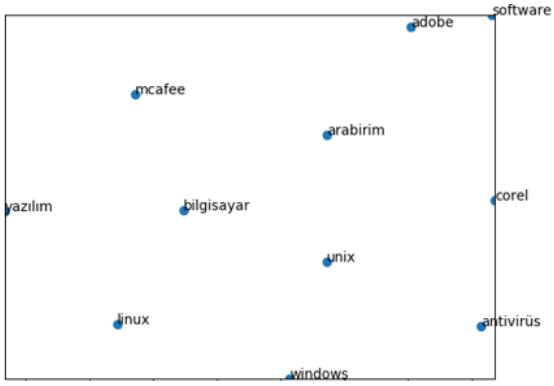
Çizelge 5 Skip-Gram algoritmasına göre analiz edildiğinde ise, hemen hemen CBOW algoritması ile aynı sonuçlara ulaşılmıştır. Yakın anlamlı kelimelerin tespitinde bir diğer Word2Vec yöntemi olan Skip-Gram algoritmasının da başarılı sonuçlar verdiği görülmektedir.

Çizelge 5 Glove yöntemine göre irdelendiğinde ise bu yöntemle de yakın anlamlı kelimelerin başarılı şekilde tespit edildiği ancak sıralanan 5 kelime içerisinde daha önce incelenen CBOW ve Skip-Gram algoritmalarında rastlanmayan bazı alakasız kelimelerinde listelendiği görülmektedir.

Çizelge 5. Word2Vec yöntemi ile eğitilen kelime benzerliklerinin incelenmesi

Kelime	CBOW	Skip-Gram	GLOVE
Vec (‘pazartesi’)	[(‘salı’, 0,9630045294761658), (‘perşembe’, 0,9585877656936646), (‘çarşamba’, 0,9553182721138), (‘cuma’, 0,9538884162902832), (‘cumartesi’, 0,9386611580848694)]	[(‘perşembe’, 0,7971779108047485), (‘çarşamba’, 0,7718572616577148), (‘salı’, 0,7703690528869629), (‘cumartesi’, 0,7620112895965576), (‘cuma’, 0,6702909469604492)]	[(‘çarşamba’, 0,5897661447525024), (‘salı’, 0,5786328315734863), (‘perşembe’, 0,5526260137557983), (‘annem’, 0,4993872046470642), (‘cumartesi’, 0,4941185712814331)]
vec (‘ocak’)	[(‘ağustos’, 0,9286551475524902), (‘mart’, 0,9277894496917725), (‘mayıs’, 0,9231933355331421), (‘şubat’, 0,9167772531509399), (‘temmuz’, 0,9152134656906128)]	[(‘şubat’, 0,5146446824073792), (‘mart’, 0,4734412431716919), (‘kasım’, 0,4670507311820984), (‘ekim’, 0,43222731351852417), (‘eylül’, 0,42404061555862427)]	[(‘nisan’, 0,7427805662155151), (‘mayıs’, 0,7262980341911316), (‘kasım’, 0,7259864807128906), (‘ekim’, 0,7126656770706177), (‘şubat’, 0,6853300333023071)]
vec (‘ankara’)	[(‘izmir’, 0,7307801842689514), (‘bursa’, 0,6751932501792908), (‘istanbul’, 0,6552906036376953), (‘adana’, 0,6191496849060059), (‘antalya’, 0,5946915149688721)]	[(istanbul, 0,5656736493110657), (‘izmir’, 0,5347393751144409), (‘antalya’, 0,5179589986801147), (‘bursa’, 0,5097886919975281), (‘erzurum’, 0,5054399967193604)]	[(istanbul, 0,5705186128616333), (‘erzincan’, 0,5660161375999451), (tokyo, 0,5624301433563232), (‘büyükelçi’, 0,5579214692115784), (‘mmo,org,tr’, 0,539327502250671)]
vec (‘beş’)	[(‘üç’, 0,900638222694397), (‘sekiz’, 0,8753728866577148), (‘dört’, 0,8631222248077393), (‘yedi’, 0,8289333581924438), (‘dokuz’, 0,779234766960144)]	[(‘sekiz’, 0,826900839805603), (‘üç’, 0,8074360489845276), (‘elli’, 0,7847145795822144), (‘dört’, 0,7656667232513428), (‘say’, 0,7624074220657349)]	[(‘o’, 0,6285558342933655), (‘altı’, 0,6276718378067017), (‘yirmi’, 0,6216707229614258), (‘sekiz’, 0,5732786059379578), (‘üç’, 0,5651163458824158)]
vec (‘mı’)	[(‘mu’, 0,565196692943573), (‘mıdır’, 0,5488075613975525), (‘mi’, 0,5180516839027405), (‘ne’, 0,5127418041229248), (‘misin’, 0,4763811230659485)]	[(‘mi’, 0,7778604030609131), (‘yoksa’, 0,7676596641540527), (‘mıdır’, 0,7531024813652039), (‘mu’, 0,7458171844482422), (‘ne’, 0,7390531301498413)]	[(‘mu’, 0,7941142916679382), (‘peki’, 0,7562239766120911), (‘evet’, 0,7499561309814453), (‘ney’, 0,7311218976974487), (‘herhalde’, 0,719590425491333)]

CBOW algoritmasına göre “bilgisayar” kelimesi ile tespit edilen yakın anlamlı kelimelerin kosinüs benzerliği kullanılarak tespit edilmesi ve 2 boyutlu gösterimi ise Şekil 5’te verilmiştir.



vec(‘bilgisayar’)
[('yazılım', 0.7579794526100159), ('software', 0.7499984502792358), ('linux', 0.7446211576461792), ('arabirim', 0.7420304417610168), ('adobe', 0.73287433385849), ('antivirüs', 0.7316679954528809), ('windows', 0.7205464243888855), ('corel', 0.7121461629867554), ('unix', 0.7088601589202881), ('mcafee', 0.7064476013183594)]

Şekil 5. Vektörün 2boyutlu düzlemde gösterimi

Word2Vec modeli kullanılarak vektörel değerler üzerinden bir kelime ile yakın anlamlı kelimeler tespit edilebildiği gibi bu yöntemde kelimelerin vektörel değerleri oluşturulduğu için kelimeler üzerinde bazı matematiksel işlemlerde yapılabilmektedir.

Çizelge 6 incelendiğinde, 1 numaralı eşitlikte Mikolov ve arkadaşlarının [9] çalışmalarında bahsettikleri eşitliğin Türkçe karşılığı denenmiştir. Buna göre, “kral” ve “erkek” kelimelerinin vektörlerinin arasındaki mesafenin, “kraliçe” ve “kadın” kelimelerinin vektörlerinin arasındaki

mesafeye eşit olması gerekmektedir. Yani "kraliçe" kelimesini elde edebilmek için "kadın" kelimesi eşitliğin diğer tarafına atıldığında (kral-erkek+kadın) oluşan eşitliğin vektörel karşılığının yaklaşık olarak "kraliçe" kelimesinin vektörel karşılığına eşit olmalıdır. 1 numaralı eşitlik incelendiğinde ise kral-erkek+kadın denkleminin sonucunda ortaya çıkan değere en yakın 5 değer sıralandığında CBOW ve Skip-Gram algoritmasına göre ilk sırada beklendiği üzere "kraliçe" kelimesi elde edilmiştir. Ancak Glove yönteminde bu sonuca ulaşılamamıştır. 2 numaralı eşitlikte ise "amca" – "erkek" = "teyze" – "kadın" eşitliği sınanmış, bu nedenle "teyze" kelimesi yalnız bırakılmıştır. Bu eşitlikteki amaç ise eşitlikten elde edilen vektörel değerin karşılığının "teyze" veya "hala" kelimesinin değerine yaklaşık olmasıdır. Bu eşitlikte de CBOW ve Skip-Gram algoritmasına göre beklenen sonuca ulaşılarak ilk sırada "teyze" kelimesi elde edilmiştir. Ancak "hala" kelimesi daha arka sıralarda yer almıştır. Glove yönteminde ise "teyze" kelimesi son sırada yer almıştır. 3 numaralı eşitlikte ise "doktor" ve "erkek" kelimeleri ile "hemşire" ve "kadın" kelimelerinin vektörel değerlerinin farklarının eşit olduğu düşünülerek daha önceki örneklerde olduğu gibi "hemşire" kelimesinin elde edilmesi amaçlanmıştır. Bu örnekte ise CBOW algoritmasına göre beklenildiği gibi "hemşire" kelimesi cevap kelime vektörü olarak döndürülmüştür. Ancak Skip-Gram algoritmasına göre "hemşire" kelimesi ikinci sırada yer almıştır. Bu durum daha nadir geçen kelimelerin tespitinde kullanılabileceği şeklinde yorumlanabilir. 3 numaralı eşitlik Glove algoritmasına göre bakıldığında ise hem beklenen sonuçlara ulaşılamadığı hem de elde edilen sonuçların alakalı kelimeler olmadığı görülmüştür.

Kelimelerin vektörleştirilerek üzerinde matematiksel işlemlerin yapılabildiği ve elde edilen sayısal değer ile ilgili vektörün arasındaki mesafeye en yakın kelimenin vektörünün döndürüldüğü bu uygulamada Glove yönteminin Word2Vec algoritmalarına göre daha başarısız olduğu görülmektedir.

Çizelge 6. Word2Vec yöntemi ile eğitilen kelime benzerliklerinin incelenmesi

CBOW	Skip-Gram	GLOVE
1. vec ("kral") - vec ("erkek") + vec ("kadın") = vec (cevap)		
[('kraliçe', 0,5058310031890869), ('prenses', 0,4587770998477936), ('imparator', 0,4512189030647278), ('prens', 0,44870319962501526), ('hükümdar', 0,4379579022979736)]	[(kraliçe, 0,6649389863014221), ('prens', 0,6619410514831543), (hükümdar, 0,6613924503326416), ('imparator', 0,6476287841796875), ('frigya', 0,6386576890945435)]	[('prens', 0,36667877435684204), (prenses, 0,3593514561653137), ('napolyon', 0,342054545879364), ('aleksandr', 0,3324899971485138), ('ferdinandı', 0,31234925985336304)]
2. vec ("amca") - vec ("erkek") + vec ("kadın") = vec (cevap)		
[('teyze', 0,6639469861984253), ('dayı', 0,5801525712013245), (hala, 0,5743905305862427), (abla, 0,5725879669189453), ('ağabey', 0,5711960792541504)]	[('teyze', 0,46290165185928345), ('enişte', 0,39535993337631226), ('dayı', 0,38905856013298035), ('anneanne', 0,38179734349250793), ('abla', 0,3774968683719635)]	[('oğul', 0,7785995602607727), ('dede', 0,7496525049209595), ('dayı', 0,743678092956543), ('baba', 0,7313336730003357), ('teyze', 0,7183244228363037)]
3. vec ("doktor") – vec ("erkek") + vec ("kadın") = vec (cevap)		
[('hemşire', 0,7681228518486023), ('hekim', 0,7603170275688171), ('hasta', 0,7203007936477661), ('kardiyolog', 0,682590901851654), ('hastabakıcı', 0,6806880235671997)]	[('hekim', 0,6817376017570496), ('hemşire', 0,6360365152359009), ('eczacı', 0,6312180161476135), ('hasta', 0,6196988821029663), ('bakıcı', 0,6053770184516907)]	[('adam', 0,37649837136268616), ('ameliyat', 0,37092214822769165), ('hasta', 0,33870816230773926), ('hanım', 0,3268465995788574), ('bleda', 0,3250676989555359)]

Çizelge 6'da bir önceki uygulamada Word2Vec algoritmaları daha başarılı oldukları için bu uygulamada Word2Vec yöntemine göre sonucalar incelendiğinde beklenen sonuçlara ulaşıldığı, özellikle Word2Vec yöntemi ile kelimelerin vektörel değerleri üzerinde matematiksel işlemlerin yapılabilmesi ve sonuçların oldukça başarılı performans gösterdiği görülmüştür.

vec(“türkiye”) - vec (“türk”) + vec (“japon”)	(‘japonya’, 0,5943777561187744)
vec(“türkiye”) - vec (“ankara”) + vec (“bakü”)	(‘azerbaycan’, 0,5270528793334961)
vec(“türkiye”) - vec (“osmanlı”) + vec (“bizans”)	(‘yunanistan’, 0,5426459312438965)
vec(“islam”) - vec (“türkiye”) + vec (“almanya”)	(‘yahudi’, 0,5666460990905762), (‘musevi’, 0,5661754012107849), (‘hıristiyan’, 0,5618569254875183)
vec(“lira”) - vec (“almanya”) + vec (“türkiye”)	(‘euro’, 0,5543266534805298), (‘sterlin’, 0,5256396532058716), (‘dolar’, 0,5223875641822815), (‘avro’, 0,4999542832374573),
vec(“odtü”) – vec(“türkiye”) + vec (“ingiltere”)	(‘birmingham’, 0,4685020446777344), (‘nortwestern’, 0,46640270948410034), (‘massachussets’, 0,46634599566459656), (‘sussex’, 0,4653938412666321),

Yine yukarıdaki tabloda görüldüğü gibi ilk iki satırda ülke-millet ve ülke-başkent sorgulaması yapılmış ikisinde de beklenen sonuçlar ilk sırada dönmüştür. Özellikle “Türkiye-Osmanlı” kelimeleri ile “Yunanistan-Bizans” kelimeleri arasındaki bağlantının tespit edilebilmiş olması

heyecan vericidir. Ülke-din sorgulamasında ise, sonuç olarak Almanya’da en çok inanılan din olan Hristiyanlığın ilk sırada çıkması beklenmiş ancak bu kelime üçüncü sırada görülmüştür. Bu durumun da derlem içerisinde geçen kelimeler ile alakalı olduğunun bir örneğidir. Derlem zenginleştikçe sonuçların doğruluk oranı da artmaktadır.

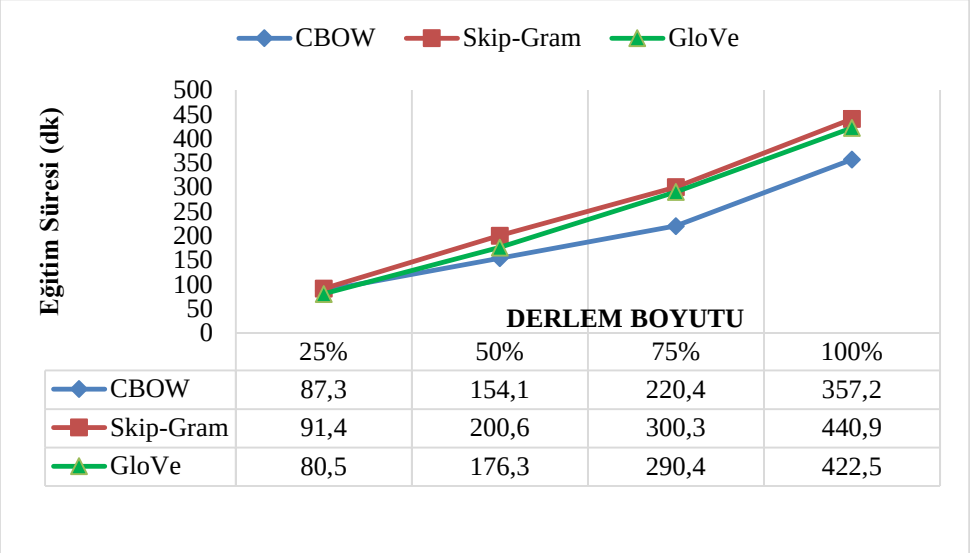
Aşağıda yer alan tabloda ise, verilen ilk iki kelime arasındaki benzerlik nokta çarpımı ile tespit edilerek çıkan sonuç ile son kelime vektörü arasındaki farkın hangi kelimenin vektörüne denk geldiği incelenmiştir. Böylece ünlü bir futbolcu olan Ronaldo ve futbol kelimelerinin vektörleri arasındaki fark ile ressam kelimesi ve bir ressamın isminin (icracı-icra edilen iş) ortaya çıkması beklenmiştir. Sonuç olarak tabloda görüldüğü gibi “Zonaro” kelimesi dönmüştür. Zonaro kelimesi Google da araştırıldığında ise, wikipedia kayıtlarında 19. yy’da Osmanlı sarayında hizmet vermiş Türk ressam olarak tanınan İtalyan bir ressam olduğu bilgisi ile karşılaşılmıştır [22]. Yani bizim bile bilmediğimiz bir bilgi model tarafından tespit edilmiştir.

Input**: en_yakın_benzerlik_bul**("ronaldo", "futbolcu", "ressam") ronaldo ve futbolcu kelimeleri, zonaro ve ressam kelimeleri kadar ilişkilidir... Output: ‘zonaro’
Input: **eslesmeyeni_bul**("bursa", "karadeniz", "marmara", “akdeniz”) Output: ‘bursa’

Bir diğer örnekte ise tanımlanan bir fonksiyon ile girdi (input) olarak verilen kelimelerin vektörleri incelenerek vektörler arasındaki en düşük değer en uzak mesafe bir başka deyişle anlam bakımından en uzak kelime olması beklenmiştir. Örnek olarak “Karadeniz, Akdeniz, Marmara ve Bursa” kelimeleri verilmiş model tarafından ilk 3 kelimenin deniz veya bölge isimleri olduğu, son kelimenin bu kelimeler ile aynı grupta yer almadığı tespit edilmiştir. Çıktı olarak “Bursa” kelimesi dönmüştür.

Derlem oluşturulduktan sonra kelime vektörlerinin CBOW, Skip-Gram ve Glove yöntemlerine göre eğitim süreleri Şekil 6’da gösterilmiştir. Derlem boyutunun %25, %50, %75 ve son olarak derlemin tamamı yani %100’lük kısmı eğitilerek eğitim süreleri incelenmiştir. Bu aşamada eğitim süreleri

ilk adımda CBOW algoritması 87,3 dk, Skip-Gram algoritması 91,4 dk ve Glove yöntemi 80,5 dk olarak elde edilmiştir. Yani bu aşamada en hızlı yöntemin CBOW algoritması olduğu bu algoritmayı Glove ve Skip-Gram algoritmasının takip ettiği söylenebilir. Grafik incelendiğinde bu durum son adıma kadar doğrusal olarak bu sıralama ile devam ettiği şeklinde yorumlanabilir.



	25%	50%	75%	100%
CBOW	87,3	154,1	220,4	357,2
Skip-Gram	91,4	200,6	300,3	440,9
GloVe	80,5	176,3	290,4	422,5

Şekil 6. Kelime vektörlerinin eğitim süreleri

6. SONUÇLAR

Bu çalışmada, metin işleme çalışmalarının en önemli bölümlerinden biri olan kelimelerin temsil edilmesi üzerinde çalışmalar yapılmıştır. Yaklaşık 60GB boyutunda 10,5 milyar kelimeden oluşan bir derlem oluşturularak bu alandaki en büyük çalışmalardan birisi yapılmıştır.

Çalışmanın katkılarından biri olan Türkçe diline özgü gereksiz kelimeler listesi TF-IDF yöntemiyle oluşturulmuş ve uygulamada kullanılmıştır. Bu işlem sonunda derlem içerisindeki kelimelerden önemli miktarda bir eleme yapılarak uygulamaya performans olarak katkı sağlanmıştır.

Oluşturulan derlem üzerinde, son dönemlerde geliştirilen en önemli iki model olan Word2Vec ve Glove yöntemleri ile kelime vektörleri oluşturulmuştur. Bu iki modelin en büyük avantajı olan kelimelerin anlamsal olarak yakınlıklarının tespit edilmesi konusu analiz edilmiştir. Bu analiz sonucunda görülmüştür ki Word2Vec yöntemi Glove yöntemine göre anlamsal benzerliklerin tespiti konusunda daha iyi sonuçlar vermiştir. Özellikle *"vec(kral) – vec(erkek) + vec(kadın) = vec(kraliçe)"* gibi bir denklem formatında vektörler üzerinde matematiksel işlemler yapılarak beklenilen kelimenin model tarafından önerilmesi işlemlerinde Word2Vec yöntemi oldukça üstün bir başarım göstermiştir.

Çalışma kapsamında Word2Vec yönteminin iki temel algoritması olan CBOW ve Skip-Gram algoritmaları da analiz edilmiştir. Çalışmada kullanılan derlem büyük bir derlem olduğu için bu derlem üzerinde CBOW algoritması daha iyi bir performans göstermiştir. Derlemin daha küçük bir kısmı ile çalışmalar yapıldığında ise, Skip-Gram yönteminin daha başarılı olduğu görülmüştür. Yine derlem içerisinde sık geçen kelimelerin temsilinde CBOW, nadir geçen kelimelerin temsilinde ise, Skip-Gram algoritmasının daha başarılı olduğu görülmüştür. Yani tercih edilecek algoritmanın, üzerinde çalışılan derleme göre tercih edilmesi daha uygun olacaktır.

Çalışmanın son kısmında ise, Word2Vec modeli CBOW ve Skip-Gram algoritmaları ile Glove yöntemi eğitim süreleri bakımından analiz edilmiştir. Öncelikle derlemin tamamı daha sonra derlemin %75'lik kısmı ardından %50 ve son olarak %25'i modellere derlem (corpus) olarak verilmiş ve eğitim süreleri kıyaslanmıştır. Bu inceleme sonunda CBOW algoritmasının eğitim süresinin tüm adımlarda Skip-Gram ve Glove algoritmalarından daha hızlı olduğu görülmüştür.

Çalışma sonunda kelime vektörlerinin kullanılarak anlamsal olarak yakın kavramların tespit edilmesinde hem Word2Vec hem de Glove yöntemlerinin başarılı oldukları ancak hem hız hem de başarım olarak Word2Vec yönteminin daha iyi olduğu görülmüştür. Oluşturulan kelime vektörlerinin, metin sınıflandırma problemlerde kullanılarak bu yöntemin sınıflandırma başarımın artıracağı düşünülmektedir.

7. KAYNAKLAR

1. Kaytan, M., Hanbay, D., 2017. Effective Classification of Phishing Web Pages Based on New Rules by Using Extreme Learning Machines. Anatolian Science-Bilgisayar Bilimleri Dergisi, 2(1), 15-36.

2. Adalı, E., 2012. Doğal Dil İşleme. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 5, 2.

3. Amasyalı, M.F., Çetin, M., Akbulut, C., 2013. Metinlerin Anlamsal Uzaydaki Temsil Yöntemlerinin Sınıflandırma Performansına Etkileri, Sigma, 5, 8-14.

4. Polat, H., Körpe, M. 2018. TBMM Genel Kurul Tutanaklarından Yakın Anlamlı Kavramların Çıkarılması. International Journal of Informatics Technologies, 11, 3.

5. Sen, M.U., Erdogan, H., 2014. Learning Word Representations for Turkish. In Signal Processing and Communications Applications Conference (SIU), 22nd 1742-1745. IEEE.

6. Gözükara, F., Özel, S.A., 2016. Türkçe ve İngilizce Yorumların Duygu Analizinde Doküman Vektörü Hesaplama Yöntemleri için Bir Deneysel İnceleme. Çukurova Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 31(2), 464-482.

7. Güngör, O., Yıldız, E., 2017. Linguistic Features in Turkish Word Representations. In Signal Processing and Communications Applications Conference (SIU), 25th 1-4. IEEE.

8. Şahin, G., 2017. Turkish Document Classification Based on Word2Vec and SVM Classifier. In Signal Processing and Communications Applications Conference (SIU), 25th, 1-4. IEEE.

9. Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficientestimation of Word Representations in Vectorspace, arXiv:1301.3781.

10. Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents. 31th International Conference on Machine Learning, China.

11. Mikolov T., Sutskever, I., Chen, K., 2013. 0010, Corrado, G., Dean, J., 2013. Distributed Representations of Words and Phrases and their Compositionality, AAAI Spring Symposium AI Technologies for Homeland Security 200591-98, cs.CL, 3111-3119.

12. Şenel, L.K., Yücesoy, V., Koç, A., Çukur, T., 2018. Interpretability Analysis for Turkish Word Embeddings. In 2018 26th Signal Processing and Communications Applications Conference (SIU) 1-4. IEEE.

13. Amasyalı, M.F., Balcı, S., Mete, E., Varlı, E.N., 2012. Türkçe Metinlerin Sınıflandırılmasında Metin Temsil Yöntemlerinin Performans Karşılaştırılması EMO Bilimsel Dergi, 2(4), 95-104.

14. Arabacı, M.A., Esen, E., Atar, M.S., Yılmaz, E., Kaltalıoğlu, B., 2018. Detecting Similar Sentences Using Word Embedding. In 2018 26th Signal Processing and Communications Applications Conference.

15. Esen, E., Özkan, S., 2017. Analysis of Turkish Parliament Records in Terms of Party Coherence. In 2017 25th Signal Processing and Communications Applications Conference (SIU) 1-4. IEEE.

16. Amasyali, M.F., Taşköprü, H., Çalışkan, K., 2018. Duygudurum Analizinde Kelimeler, Anlamlar, Karakterler Words, Meanings, Characters in Sentiment Analysis. In 2018

Innovations in Intelligent Systems and Applications Conference (ASYU).

17. Ayata, D., Saraclar, M., Ozgur, A., 2017. Turkish Tweet Sentiment Analysis with Word Embedding and Machine Learning. 1-4. 10.1109/SIU.2017.7960195.

18. Keleş, M.K., Özel, S.A., 2017. Similarity Detection Between Turkish Text Documents With Distance Metrics. In 2017 International Conference on Computer Science and Engineering (UBMK) 316-321. IEEE.

19. Penington, J., Socher, R., Manning, C.D., 2008. GloVe:GlobalVectors for Word Representation, Emprical Methods in Natural Language Processing (EMNLP), 1532-1543.

20. Jones, S., Karen, 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval. Journal of Documentation, 28(1), 11-21.

21. Rong, X., 2014. Word2Vec Parameter Learning Explained, arXiv:1411.2738.

22. https://tr.wikipedia.org/wiki/Fausto_Zonaro, (15.05.2019 tarihinde erişildi)