KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

GİZLİ DIRICHLET AYRIMI VE WORD2VEC YÖNTEMLERİNİN BİRLEŞİMİ İLE ÖZGÜN BİR METİN TEMSİL MODELİ GELİŞTİRİLMESİ

HALİL İBRAHİM ÇELENLİ

KOCAELİ 2020

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

YÜKSEK LİSANS TEZİ

GİZLİ DIRICHLET AYRIMI VE WORD2VEC YÖNTEMLERİNİN BİRLEŞİMİ İLE ÖZGÜN BİR METİN TEMSİL MODELİ GELİŞTİRİLMESİ

HALİL İBRAHİM ÇELENLİ

Doç. Dr. Sevinç İlhan OMURCA
Danışman, Kocaeli Üniversitesi .....

Doç. Dr. Murat Can GANİZ
Eş Danışman, Marmara Üniversitesi .....

Prof. Dr. Nevcihan DURU
Jüri Üyesi, Kocaeli Üniversitesi .....

Dr. Öğr. Üyesi Orhan AKBULUT
Jüri Üyesi, Kocaeli Üniversitesi .....

Dr. Öğr. Üyesi Bilge ŞİPAL
Jüri Üyesi, İstanbul Kültür Üniversitesi .....

Tezin Savunulduğu Tarih: 17.04.2020

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışması, kelime kalıplama yöntemlerinin geliştirilmesi ve yeni bir kalıplama yöntemi algoritması geliştirmek amacıyla gerçekleştirilmiştir.

Tez çalışmamda desteğini esirgemeyen, çalışmalarıma yön veren, bana güvenen ve yüreklendiren tez danışmanım Doç. Dr. Sevinç İlhan OMURCA ve eş danışmanım Doç. Dr. Murat Can GANİZ hocama sonsuz teşekkürlerimi sunarım.

Yüksek lisans öğrenimim boyunca görüşleri ile çalışmalarıma katkıda bulunan, karşılaştığım her zorlukta desteğini ve zamanını esirgemeyen hocam Dr. Aydın GEREK ve Dr. Ekin EKİNCİ hocam ile TÜBİTAK 116E047 nolu proje kapsamındaki tezim sırasında beni maddi açıdan destekleyen Tübitak Araştırma Destek Programları Başkanlığı'na teşekkürlerimi sunarım.

Hayatım boyunca bana güç veren en büyük destekçilerim, her aşamada sıkıntılarımı ve mutluluklarımı paylaşan sevgili babam Süleyman ÇELENLİ, annem Seyhan ÇELENLİ ve ablam Dr. Azize ZEHRA ÇELENLİ BAŞARAN'a sonsuz teşekkürlerimi sunarım.

Nisan – 2020 Halil İbrahim ÇELENLİ

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

α	:	Dirichlet parametresi
b	:	Softmax fonksiyon parametresi
b	:	Eğilim değeri
β	:	Dirichlet parametresi
c	:	Pencere boyutu temsili
d	:	Değişken
D	:	Doküman koleksiyonu
FP	:	Yanlış pozitif
FN	:	Yanlış negatif
h	:	Kelime vektörlerinin birleşim parametresi
k	:	Değişken
L	:	Loss (Kayıp)
lg	:	Lojistik regresyon parametresi
m	:	TopWord2vec modeli doküman temsili
n	:	Kelime ağırlığı parametresi
φ	:	Konu kelime dağılımı
s	:	Ağırlık vektörü
t	:	Değişken
T	:	Toplam konu değişkeni
TN	:	Doğru negatif
TP	:	Doğru pozitif
θ	:	Doküman konu dağılımı
U	:	Softmax fonksiyon parametresi
V	:	Word2vec eğitiminden sonraki toplam korpus kelime değişkeni
V'	:	Word2vec vektörlerinin olasılık dağılımları ile çarpım sonucu
y	:	Normalize edilmemiş log olasılığı
z	:	Konu parametresi
w	:	Kelime temsili
W	:	Kelime matrisi

Kısaltmalar

BOW	:	Bag of Words (Kelime Çantası)
CLSTM	:	Contextual Long Sort-Term Memory (Bağlamsal Uzun-Kısa Süreli Bellek)
CNN	:	Convolution Neural Network (Konvolüsyonel Sinir Ağları)
DM	:	Distributed Memory (Dağıtık Bellek)
GMM	:	Gaussian Mixture Model (Gauss Karışım Modeli)
GNM	:	Gaussian Naive Bayes (Gaussian Naif Bayes)
KNN	:	k-Nearest Neighbors (En Yakın K Komşu)
LDA	:	Latent Dirichlet Allocation (Gizli Dirichlet Ayırımı)
LSI	:	Latent Semantic Analysis (Gizli Semantik Analiz)

RNNLM	:	Recurrent Neural Network Language Modeling (Tekrarlayan Sinir Ağı Dil Modeli)
SVM	:	Support Vector Machine (Destek Vektör Makineleri)
TF-IDF	:	Term Frequency- Inverse Document Frequency (Terim Frekansı-Ters Doküman Frekansı)

GİZLİ DIRICHLET AYRIMI VE WORD2VEC YÖNTEMLERİNİN BİRLEŞİMİ İLE ÖZGÜN BİR METİN TEMSİL MODELİ GELİŞTİRİLMESİ

ÖZET

Son zamanlarda veri miktarındaki artış ile derin öğrenme, makine öğrenmesinin en popüler alanı olmaya başlamıştır. Bu artış ile Doğal Dil İşleme alanında da yeni yöntemlerin geliştirilmesini sağlamıştır.

Metinsel verilerin temsil edilmesi, geleneksel yöntemler üzerinde Kelime Çantası Modeli gibi kelime temsil yöntemleri kullanılarak temsil edilir. Fakat yeni yöntemler üzerinde hızlı ve verimli olabilmesi için kelime kalıplama yöntemleri kullanılmaya başlanmıştır. Kelime kalıplama yöntemlerinin en popüler olanı Word2vec yöntemidir. Word2vec yöntemi kelimelerin bağlamlarındaki istatistiklere bakarak, yapay sinir ağlarını kullanarak her kelime için bir vektör gösterimini öğrenmektedir. Dokümanların temsil edilmesi için ise Doc2vec olarak bilinen kelime kalıplama yöntemi temelli yöntem kullanılmaktadır.

Konu modelleme teknikleri ise kelimelerin konu olasılık dağılımları üzerinde rastgele bir araya gelerek dokümanları oluşturmaktadır. En sık kullanılan modeli Gizli Dirichlet Ayırımı (LDA) modelidir. LDA modeli konuların dokümanlar üzerindeki dağılımı ile kelimelerin konular üzerindeki dağılımı olmak üzere 2 farklı dağılım üretmektedir.

Tez çalışması içerisinde Word2vec yöntemi, LDA model dağılımları ile birleştirip yeni bir kelime kalıplama vektörü geliştirilmiştir. Bu sayede dokümanlar daha iyi temsil edilmiştir. Geliştirilen yöntem ile doküman temsilinde kullanılan Doc2vec yöntemleri sınıflandırma algoritmaları kullanılarak karşılaştırılmıştır. Sınıflandırma sonucunda geliştirilen yöntemin sonuçları iyileştirdiği ve model karmaşıklığını azalttığı gösterilmiştir.

Anahtar Kelimeler: Doc2vec, Kelime Kalıplama, LDA, TopWord2vec, Word2vec.

COMBINING LATENT DIRICHLET ALLOCATION AND WORD2VEC FOR A NOVEL DOCUMENT REPRESENTATION MODEL

ABSTRACT

Recently, with the increase in the amount of data, deep learning has become the most popular field of machine learning. With this increase, new methods have been developed in the field of Natural Language Processing.

Representation of textual data is represented on traditional methods using word representation methods such as the Bag of Words model. However, word embeddings methods are use in order to be fast and efficient on new methods. The most popular method of word embeddings is Word2vec. The Word2vec method learns to view a vector for each word using artificial neural networks, looking at the statistics in the context of the words. For the representation of the documents, the word embedding method known as Doc2vec is use.

Topic modeling techniques are randomly generated on the topic probability distributions of the words and establish the documents. The most commonly used model is the Latent Dirichlet Allocation (LDA). The LDA model produces 2 different distributions, the distribution of topics on documents and the distribution of words on topics.

This thesis, a new word embedding vector was developed by combining the Word2vec method with the LDA model distributions. In this way, the documents are better represented. The developed method and Doc2vec methods document representation were compared using classification algorithms. It has been shown that the method developed as a result of classification improves results and reduces model complexity.

Keywords: Doc2vec, Word Embedding, LDA, TopWord2vec, Word2vec.

GİRİŞ

Günümüzde metin verilerinin çoğalması ile bu verilerin anlamsal ve sözdizimi benzerliklerinin çıkarılması için yeni yöntemler geliştirilmiştir. Bu yöntemler istatistiksel tabanlı yöntemler, anlamsal benzerlik (semantic similarity) yöntemleri ve derin öğrenme tabanlı yöntemler vb. olmaktadır. Metin verileri üzerinde doküman sınıflandırma, bilgi alma (information retrieval), metin madenciliği vb. işlemler önemli bir araştırma konusudur. Doküman sınıflandırmanın temel amacı belirtilen bir dokümana en uygun etiketin atanmasıdır. Birçok sınıflandırma algoritması benzer hipoteze sahip olmaktadırlar. Sınıflandırma algoritmaları bu süreçte anlamsal veri bilgisini içermemekle birlikte anlamsal veri bilgisini içerebilecek şekilde tasarladığında daha güçlü hale gelmektedirler [1].

Doküman sınıflandırma sistemlerinde çoğu zaman Kelime Çantası (BOW) yaklaşımı gibi geleneksel kelime temsil yöntemleri kullanılmaktadır. Bu yöntemler kelimelerin anlamsal ilişkisinden daha çok birbirleri ile birlikte geçme durumları üzerinden değerlendirmektedir. Bu yöntemler kullanıldığında derlemin (sınıflandırılacak dokümanların bütünü) içerisinden kelimeler üzerinde işlemler yapılmaktadır.

Son zamanlarda geleneksel kelime temsil yöntemlerinin aksine modern kelime temsil yöntemleri ile kelimelerin anlamsal ve söz dizimi benzerlikleri daha net bir şekilde çıkarılmaktadır. Sadece anlamsal ve sözdizimi benzerliklerinin dışında kelimenin çevresinde bulunan bağlamları da çıkarabilmekte olup düşük boyutlu vektörler ile ifade edilmektedirler. Modern kelime temsil yöntemleri içerisinde en popüler yöntem Word2vec yöntemidir [2]. Word2vec yöntemi bir kelime üzerinden çevresinde bulunan kelimeleri çıkarabildiği gibi, çevresinde bulunan kelimeler üzerinden kelime tahmin işlemini de gerçekleştirebilen bir kelime temsil yöntemidir. Dokümanlar için modern temsil yöntemleri de bulunmaktadır. Bunlar içerisinde en popüler olanı Doc2vec yöntemidir [3]. Bu yöntem Word2vec yöntemine benzerlik gösterir. Fakat burada kelimeler yerine dokümanlar temsil edilmektedir. Temel olarak dokümanlar içerisinde kelimeler tahmin edilirken eğitilmiş bir paragraf vektörü kullanılmakta olup

birçok farklı doküman temsil yönteminden daha yüksek başarım oranı sağlamaktadır [4].

Dokümanların anlamsal benzerliğini yakalamak için kullanılan diğer bir yöntem ise konu modelleme teknikleridir. Konu modelleme yöntemleri denetimsiz öğrenme ile gerçekleştirilmektedirler. Konu modelleri metin sınıflandırma, metin özetleme (summarization) veya bilgi alma gibi metin madenciliği konularını gerçekleştirmek için gizli doküman gösterimlerini öğrenmektedirler. En popüler ve yaygın kullanıma sahip konu modeli Gizli Dirichlet Ayırımı (LDA) modelidir [5]. LDA, benzer kelimeleri konular özelinde bir araya getirerek konuların dağılımları üzerinden dokümanları temsil etmektedir. LDA üzerinde dağılımlar multinomial dağılımın genel bir yapısı olan dirichlet dağılımı olarak kabul edilir.

Doküman sınıflandırma yöntemlerinin başarısı temel olarak dokümanları temsil eden vektörlerin kalitesine dayanmaktadır. Bu nedenle derlem hakkında anlamsal ve bağlamsal (contextual) bilgiyi iyi yakalayabilen düşük boyutlu vektörler oluşturmak önemlidir [6].

Bu tez çalışmasında iki farklı temsil yöntemi kullanılarak dokümanlar temsil edilmiştir. Birinci temsil yöntemi Doc2vec yöntemidir [3]. İkinci temsil yöntemi, tez kapsamında önerilen, konu kelime vektörlerini oluşturarak dokümanların temsil edildiği TopWord2vec yöntemidir. Önerilen ve geliştirilen yöntem ile amacımız derlem içerisinde bulunan her belgeyi karakterize eden gizli temaları ele alan kelime vektörleri ile daha iyi bir temsil yöntemi oluşturmaktır.

Önerilen TopWord2vec yönteminde, bulunan gizli konular (latent topics) eğitimli derlemin anlamsal temsilini geliştirmek için kullanılmaktadırlar. Kelime kalıplama vektörlerini yakalamada konu (kelimeler üzerindeki dağılım) ve doküman (konular üzerindeki dağılım) ilişkisinin etkileri incelenmiştir.

Bu amaçla doküman konu ve konu kelime dağılımları LDA modelinin eğitilmesi ile üretilir. Ayrıca derlem içerisinde bulunan her dokümanın kelime vektörleri Word2vec yöntemi ile üretilmektedir. Sonuçta derlemde bulunan her bir belge için LDA ve kelime vektörlerinin çıktı dağılımları hem kelime ilişkilerinin hem de konu

bileşenlerinin anlamsal bilgi olarak değerlendirildiği yeni doküman vektörü oluşturmak için çarpılır.

Tez kapsamında gerçekleştirilen deneylerde geliştirilen doküman vektörü yaklaşımı (TopWord2vec) Doc2vec temsil yöntemi ile karşılaştırılır. Bu karşılaştırma yapılırken doküman vektörleri çeşitli sınıflandırıcı algoritmalarına girdi olarak verilmektedirler. Sınıflandırma algoritmaları olarak makine öğrenmesi literatüründe yaygın ve başarı ile uygulanan Destek Vektör Makineleri (SVM), En Yakın K Komşu (KNN), Rastgele Orman, Bayes ve Lojistik Regresyon algoritmaları seçilmiştir. Ardından sınıflandırıcı algoritmaların başarım oranları birbirleri ile karşılaştırılır. Yapılan deneylerde geliştirilen yöntem ile çıkan başarım oranlarının kelime kalıplama yöntemlerinden olan Doc2vec ile karşılaştırıldığında başarım oranlarının daha yüksek olduğu gözlenmiştir. Ayrıca eğitim süreleri göz önüne alındığında, TopWord2vec'in Türkçe ve İngilizce doküman koleksiyonlarında Doc2vec'ten daha hızlı olduğu gözlenmiştir. Farklı yaklaşımlar ile farklı sayıda seçilen konuların geliştirilen yöntem üzerindeki etkisi araştırılmıştır. LDA modeli üzerinde farklı şaşkınlık (perplexity) değerleri kullanılarak en etkili konu sayısı belirlenmeye çalışılmıştır [7].

1. İLGİLİ ÇALIŞMALAR

Kelimelerin veya dokümanların sürekli uzayda (continuous space) dağıtılmış vektör temsilleri metin madenciliği sistemlerinde uzun zamandır kullanılmaktadır [8]. İlham verici yaklaşımlardan birisi doğrusal bir projeksiyon katmanı ve doğrusal olmayan bir gizli katmanı olan ileri beslemeli bir sinir ağı oluşturan [9] modelidir. Bu sinir ağı modeli n-gram modelleri üzerinde iyileştirmeler sağlamıştır. Mikolov ve arkadaşlarının Tekrarlayan Sinir Ağı Dil Modeli (RNNLM) [10] öğrenmede yerel ve küresel dokümanlar arasındaki ilişkileri anlamsal açıdan daha iyi temsil eden kelime vektörleri oluşturmak için sunulmuştur. Dokümanların temsil edilmesini sağlayan Doc2vec modeli, Le ve Mikolov [3] tarafından önerilmiştir. Makalelerinde, paragraf gibi değişken uzunluktaki metinlerden sabit uzunlukta vektörler öğrenen denetimsiz bir algoritma olarak anlatmışlardır. Paragraf vektörünün Kelime Çantası modelinden daha başarılı olduğunu göstermişlerdir. Doc2vec modeli bir doküman vektörünü, kelime kalıplarının doğrusal bir kombinasyonuyla eğitmektedir.

Pennington ve diğ. tarafından önerilen Glove modeli [11] küresel olarak derlem istatistiklerini doğrudan yakalayan bir modeldir. Bunu başarmak için global kelime oluşumları (word co-occurrences) üzerinde belirli bir ağırlığın eğitilmesini sağlayan en küçük kareler modelini (least squares model) geliştirmişlerdir.

Yukarıda bahsedilen yöntemler iyi sonuçlara sahiptirler fakat anlamsal olarak cümleleri veya paragrafları temsil etmek için yeterli değillerdir. Bu nedenle literatürde daha iyi bir kelime veya paragraf vektörü için LDA ile kelime kalıplama modellerini birleştiren çalışmalar bulunmaktadır.

Le ve Mikolov [3], farklı uzunluktaki paragraflar için sürekli uzayda dağıtık vektör gösterimlerini öğrenen denetimsiz bir yapı önermiştir. Paragraf vektörü, paragrafın konusunu hatırlayan bir bellek gibi düşünülebilir. Bu denetimsiz yöntem Dağıtık Bellek (DM) adı ile anılmaktadır. Cümleye veya paragrafa rahatlıkla uygulanabilen bir yöntemdir. Ayrıca yöntemde kullanılan paragraf vektörleri benzersiz özelliği taşımaktadır. Bir kelime gerçekte farklı anlamlara gelebilir fakat vektör olarak tek bir

temsili bulunmaktadır. Bu işlem paragrafın yanlış anlamsal temsiline yol açması nedeniyle modelin zayıf yönü olarak düşünülebilir [12].

Liu ve diğ. [6], her kelimenin farklı konular altında farklı kelime kalıplarının içerdiğini düşünen konusal kelime kalıplama (topical word embedding) modelini önermişlerdir. Kelime konu ilişkisini kelime kalıplama modeline entegre ederek kelimenin farklı konulardaki çeşitli anlamlarını keşfetmişlerdir. LDA ve Skip-Gram [2] modelleri sırasıyla kelime konu çiftlerini elde etmek ve sırasıyla konusal kelime kalıplarını öğrenmek için uygulanmaktadırlar.

Wang ve diğ. [12], her kelimeyi farklı konular altında farklı kelime kalıpları kullanarak konuları kelime kalıpları ile birleştirerek temsil edip bunları konusal paragraf vektörü oluşturmak için kullanmışlardır. Paragrafın anlamsal bilgisini daha iyi keşfetmeyi amaçlamışlardır. Her kelimeyi belirli bir konu ile vermek için LDA yöntemini uygulamışlardır. Sonrasında Skip-Gram modeli kullanarak konulara atanan her kelime için birden fazla kelime kalıbı temsili elde etmişlerdir.

Ghosh ve diğ. [13], LSTM modeline konular dahil edilmiş olup Bağlamsal Uzun-Kısa Süreli Bellek (CLSTM) olarak adlandırılmaktadırlar. Denetimli konu modelleri yerine vektörler ile konu modelleme kullanılmıştır.

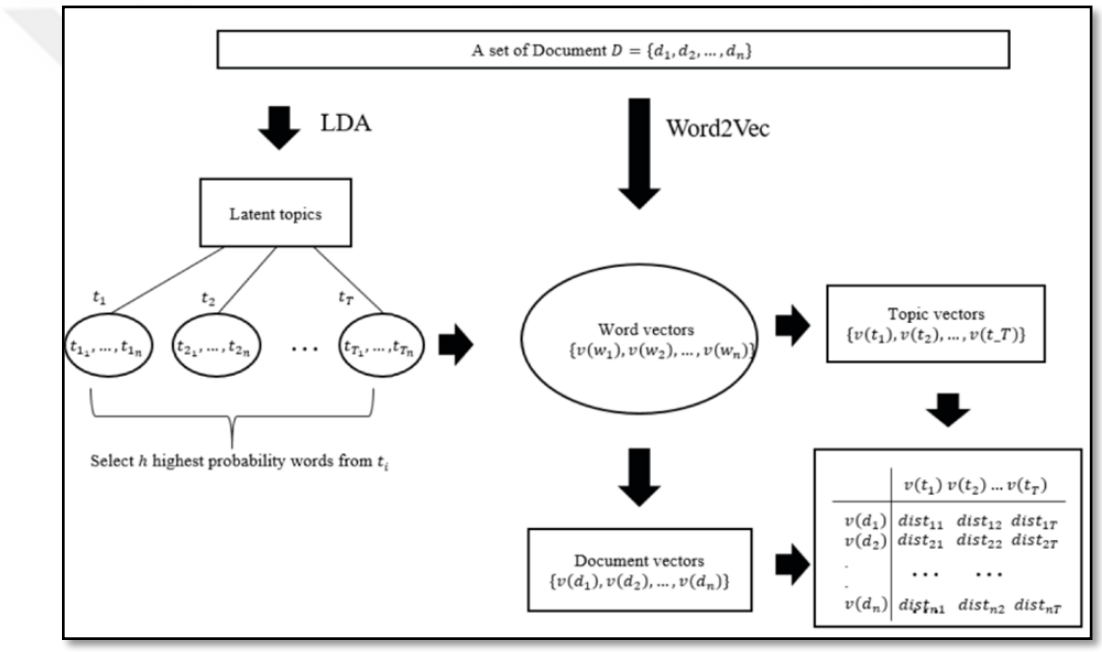
Mikolov ve Zweig [14], konuyla ilgili bir tekrarlayan sinir ağı dili modellerini, LDA yöntemi kullanılarak sabit uzunlukta metin bloklarına uygulayarak çıkarmışlardır. Burada konu bilgisini modelin girişinde sağlamışlardır.

Jiang ve diğ. [1], kelime vektörlerini öğrenmek için Word2vec algoritmasını uygulamışlardır. Sonrasında aynı konudaki kelime vektörlerinin, her Gauss'un gizli bir konuyu temsil ettiği, tüm kelimelerin dağılımını tanımlayan bir Gauss Karışım Model'ine (GMM) uyduğunu düşünmüşlerdir. Metin oluşturma işleminden sonra, konuları nedeniyle metinleri temsil etmek için düzenlenmiş GMM kullanır. GMM ağırlık katsayısı (coefficient) kelimenin her konuya ait olma olasılığı ile hesaplanmaktadır.

Nguyen ve diğ. [15], gizli özellik temsillerinin konu modellerini geliştirebilmek için kullanabileceğini göstermeye çalışmışlardır. Kelime-konu haritalaması (mapping),

çok büyük derlem üzerinde eğitilen gizli özellik vektör temsilleriyle birlikte iki farklı dirichlet çoklu konu (dirichlet multinomial) modelinin dahil edilmesiyle geliştirilmiştir.

Wang ve diğ. [16], Word2vec ve LDA modelini yalnızca doküman konu ilişkileri açısından değil aynı zamanda kelimelerin arasındaki bağlamsal (contextual) ilişkiler oluşturmak için karma bir yöntem önermiştir. Burada konu vektörleri ile doküman vektörleri arasındaki öklit uzaklığı kullanılarak vektörler oluşturulmuştur. Modellerin performansı, 20NewsGroup veri kümesinde Destek Vektör Makineleri (SVM) modeli kullanılarak incelenmiştir. Şekil 1.1 üzerinde oluşturulan model gösterilmiştir.



Şekil 1.1. LDA ve Word2vec yöntemlerinin öklit uzaklığı ile birleştirilmesi [16]

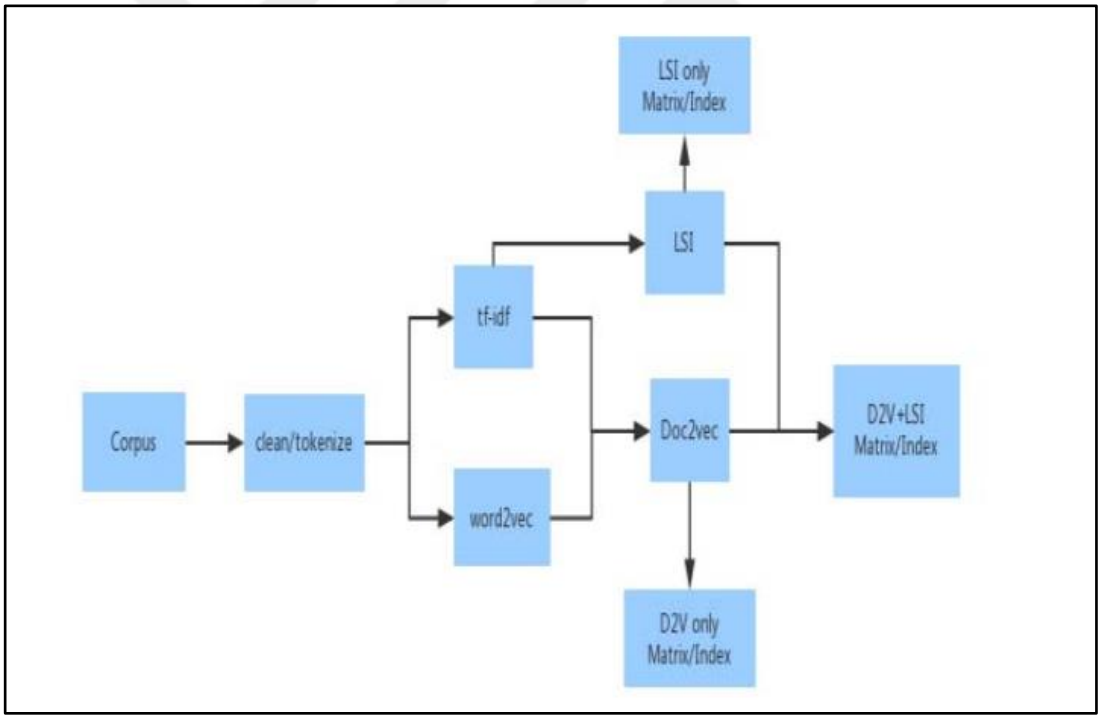
Yatırımcılar arasında fikir alışverişi yapılan bir sosyal medya platformu üzerinden Kelime Çantası temsil yöntemi, Doc2vec ve Konvolüsyonel Sinir Ağları (CNN) algoritması ile karşılaştırıldığında CNN algoritmasının en iyi sonucu verdiği sonucuna varılmıştır [17].

Çin medikal hastaneleri üzerinden toplanan veriler ile nem-ısı sendromu sınıflandırılması yapılmaya çalışılmıştır. En iyi sonucu Word2vec algoritmasının Terim Frekansı-Ters Doküman Frekansı (TF-IDF) ve En Yakın K Komşu (KNN) ile birleştirilmesiyle oluştuğu gösterilmiştir [18].

Türkçe ve İngilizce twitter verileri üzerinden Doc2vec yöntemi ile duygu analizi yapılmaya çalışılmıştır. Dağıtık Bellek yöntemi ile Kelime Çantası yöntemi karşılaştırıldığında Kelime Çantası yönteminin daha iyi sonuçlar verdiği belirtilmiştir [19].

Türkçe metinler üzerinde yapılan bir çalışmada, Word2vec kelime vektörlerinin kullanımı Kelime Çantası yöntemi ile karşılaştırılmıştır. Word2vec kelime vektörlerinin, TF-IDF ağırlıklandırmalı Kelime Çantası yönteminden daha iyi sonuçlar verdiği belirtilmiştir [20].

Öğrencilerin doğru ders seçimini sağlamalarına yönelik yapılan bir çalışmada Doc2vec ve Gizli Anlamsal Analiz (LSI) birlikte kullanıldığında tek kullanımlarına göre daha iyi sonuç verdiği belirtilmiştir [21]. Şekil 1.2 üzerinde oluşturulan model gösterilmiştir.



Şekil 1.2. Docvec ve Gizli Anlamsal Analiz model birleşimi [21]

2. YÖNTEM

Metinlerin genel olarak makine öğrenmesi uygulamalarında kullanılabilmeleri için belirli temsil yöntemleri ile temsil edilmeleri gerekmektedir. Bu temsil yöntemleri anlamsal analiz, konu modelleme, metin sınıflandırma gibi bir çok farklı alanlarda kullanılmaktadır. Temel olarak kelime temsil yöntemleri sayısal vektörler üzerine kuruludur. Bu vektörler kullanılan temsil yöntemine göre farklı özellikler içermektedirler.

Kelime temsil yöntemleri dokümanları temsil etmek içinde kullanılmaktadırlar. Bu sayede dokümanlar arası ilişkiler, kurulan makine öğrenmesi modelleri ile açıklanabilmektedir. Örnek olarak spor alanıyla ilgili bir doküman için kullanılan terimler (kelimeler) ile magazin alanıyla ilgili bir doküman için kullanılan terimler birbirinden ayrışmaktadırlar. Ayrışan terimler spor ve magazin dokümanları için ayırt edici bir özellik olmaktadırlar. Bu terimler kelime temsil yöntemleri ile dokümanlar arasındaki ilişkileri çıkarmaktadır.

Kelime temsil yöntemlerini geleneksel ve modern yöntemler olarak ikiye ayırabiliriz. Geleneksel yöntemler, ilk makine öğrenmesi algoritmaları için elverişli oldukları düşüncesi ile oluşturulmuştur ve sabit uzunluklu vektörler ile oluşturulmaktadırlar. Örnek olarak bir doküman içerisinde geçen tekil kelimelerin sayısı kadar boyutlu vektörler, dokümanı temsil eden sabit uzunluklu vektör olarak adlandırılmaktadırlar. Sabit uzunluklu vektörler makine öğrenmesi algoritmalarına giriş olarak verildiği gibi çıkış olarakta sabit uzunluklu vektörler alınmaktadır. Modern yöntemler ise sabit bir uzunluk yerine farklı uzunlukta (boyutlu) modeller ile oluşturulabilmektedirler. Özellikle kurulan derin öğrenme modellerinde anlamsal analiz, doküman sınıflandırma gibi alanlarda daha iyi sonuçlar çıkarabilmektedirler.

Modern kelime temsil yöntemleri, kelime vektör uzayında kelimelerin arasındaki anlamsal ve sözdizimi ilişkilerini çıkarmak için kullanılmaktadırlar. Bu yöntemler kelimeleri vektör uzayında çok boyutlu vektörler ile ifade ederek sıklıkla yapay sinir ağı tabanlı mimariler üzerinde metinlerin temsil edilmesi için kullanılmaktadırlar [9].

Modern kelime temsil yöntemleri, yapay sinir ağı tabanlı modeller ile büyük metin verileri üzerinde kelime çantası yöntemi gibi geleneksel temsil yöntemlerinin hız ve başarım konusunda yetersiz kalması durumunda ortaya çıkmış bulunmaktadırlar. Bu yöntemler özellikle büyük metin verilerin temsil edilmesinde kullanıldığında sınıflandırma alanında başarılı olmaktadırlar [20, 22, 23].

Geleneksel yöntemler, sıklıkla çok fazla kelime olmadığı veya çok fazla doküman olmadığı durumlarda iyi sonuçlar çıkarmaktadır. Temel sebebi ise fazla doküman veya kelimeye sahip kurulan modellerde çıktı süreci uzun bir zaman aldığı gibi dokümanlar iyi temsil edilmediği için başarım (accuracy) oranları düşük olabilmektedirler. Yeni yöntemler ise sıklıkla çok fazla kelime olduğu veya çok fazla doküman olduğu durumlarda iyi sonuçlar çıkarmaktadırlar. Nedeni ise farklı vektör uzunlukları ile oluşturulan modeller ile dokümanlar daha iyi temsil edilip başarım oranları geleneksel yönteme göre yüksek olmaktadırlar. Modern yöntemler büyük kelime haznesine sahip verilerde ağırlıklı olarak derin öğrenme algoritmalarında çok katmanlı mimariler ile kullanılmaktadırlar.

2.1. Kelime Çantası (BOW) Yöntemi

Kelime Çantası yöntemi, her kelimenin sırasız bir şekilde ele alındığı ve kelimelerin bir çanta içinde toplanması olarak düşünülen bir sırasız temsil yöntemidir. Burada her kelime birbirinden bağımsız olarak ele alınmaktadır. Çanta içerisinde bulunan her kelime bir özellik olarak ele alınıp kelimelerin geçme sıklığına göre işlem yapılır. Sık geçen kelimeler sınıflandırma, konu modelleme vb. için önemli özellikleri içermektedirler. Fakat her sık geçen kelime aslında önemli olmayabilir. Örnek olarak “O” ve “ve” kelimeleri dokümanlar içerisinde sıklıkla geçmelerine rağmen ayırt edici bir özellik olarak bulunmamaktadır. Sebebi ise diğer dokümanlarda da sıklıkla geçilebileceği için birbiri ile ilişkili olmayan dokümanları ilişkili olarak çıkarabilecek olmalarıdır.

Örnek olarak 2 farklı dokümanın Kelime Çantası yöntemiyle temsil edilmesi;

1. Doküman: Zehra okula yakın mesafede oturuyor. Ayrıca Zehra okula gitmeyi çok seviyor.

2. Doküman: Halil Zehra ile birlikte sinemaya çok gider.

Toplam Tekil Kelimeler (Vocabulary): [“Zehra”, “okula”, “yakın”, “mesafede”, “oturuyor”, “Ayrıca”, “gitmeyi”, “çok”, “seviyor”, “Halil”, “ile”, “birlikte”, “sinemaya”, “gider”]

Toplam Tekil Kelime Sayısı (Vocabulary Size): 14

1. Dokümanın Kelime Çantası Yöntemi ile Temsili:

[2, 2, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0]

2. Dokümanın Kelime Çantası Yöntemi ile Temsili:

[1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1]

2.2. Word2vec Yöntemi

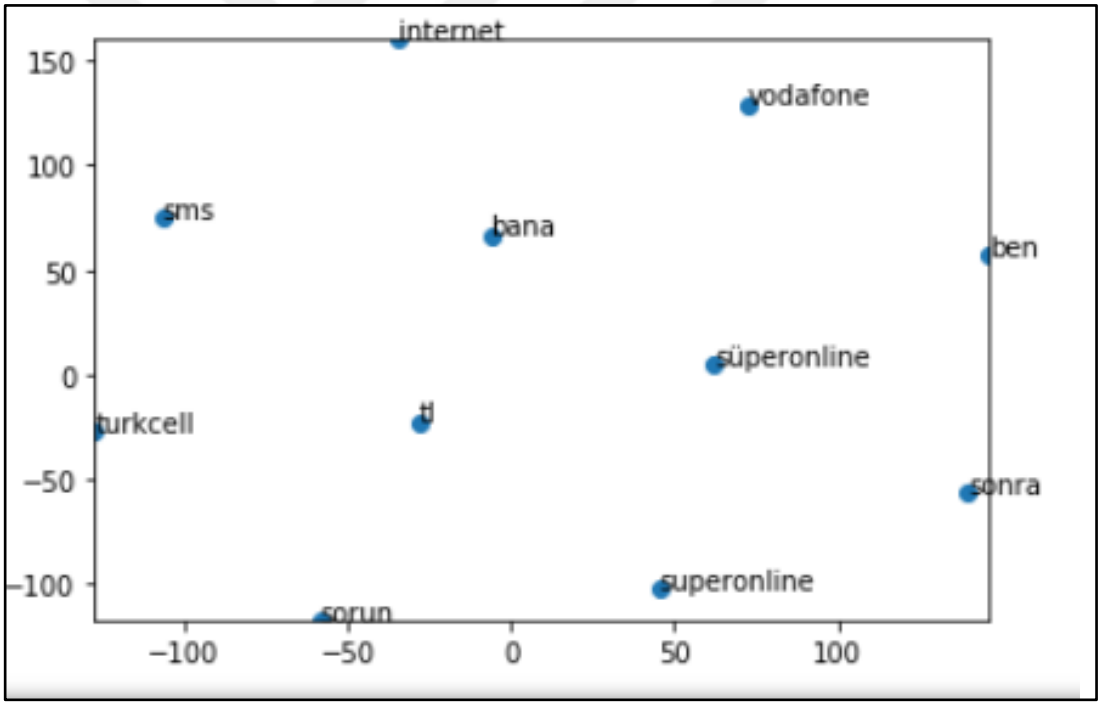
Modern kelime temsil yöntemleri içerisinde en sık kullanılan yöntemdir [2]. Kelimeler çok boyutlu vektörler ile temsil edilmektedirler. Vektör boyutu, toplam kelime sayısına bağlı bulunmamaktadır. Bu sayede istenilen uzunlukta vektörler oluşturulmaktadırlar. Oluşturulan vektörler ile kelimelerin anlamsal ve sözdizimi ilişkileri ikili (one hot encoding) yöntemini kullanan makine öğrenmesi modellerinden daha iyi sonuçlar çıkardığı görülmektedir [24].

Word2vec, Büyük metin derlemlerinin yapay sinir ağları kullanılarak düşük boyutlu vektörleri (low dimensional vectors) oluşturan bir kalıplama (embedding) yöntemidir. 2013 yılında Tomas Mikolov tarafından tanıtılmıştır. Temel olarak girdi (input layer) çıktı (output layer) ve gizli (hidden layer) katmanlarından oluşmaktadırlar. Kelimeleri 2 önemli parametre olan pencere genişliği ve kalıplama boyutu gibi hiper parametreleri kullanarak vektörler şeklinde şeklinde temsil etmektedirler. Burada pencere genişliği, kelime bağlamları (context) içerisinde yer alan hedef kelimenin sağındaki ve solundaki kelime sayısını ifade ederken, kalıplama boyutu ise vektör boyutunu temsil etmekte olup, gizli katmandaki nöron sayısına da denk gelmektedir. 2 ayrı modeli bulunmaktadır. Bunlar; Sürekli Kelime Çantası (Continuous Bag of Words) ve Skip-Gram modelleridir. Modeller, kelime bağlamlarını (context) kullanarak kelimeleri tahmin etme üzerine kurulmuştur [2]. Örnek olarak “Süperonline” kelimesinin 100 boyutlu vektör gösterimi ve vektör uzayında gösterimi Şekil 2.1 ile 2.2 üzerinde gösterilmiştir.

```
print(new_model['süperonline'])
```

```
[ 0.00087085  0.00470586 -0.00030294  0.00447303  0.00093864 -0.00370474
  0.00600333  0.00179739  0.00196089  0.00241442  0.00080457  0.00242651
 -0.00132446 -0.00014065  0.00682907  0.00175999  0.00367266 -0.00598207
 -0.00526605 -0.00145621  0.00880417  0.00225458 -0.00377975 -0.00250336
  0.00385645 -0.00334837  0.00409962  0.00313349  0.00153113  0.00485971
  0.00108441  0.00081667  0.00547692 -0.00376104  0.00194059 -0.00203735
  0.00296621 -0.00311403  0.00076961 -0.00180855  0.00681291 -0.00243099
  0.00201186  0.001964    0.0001879  -0.00233274  0.00169418 -0.00032374
  0.00217338  0.00320698 -0.00025195  0.00313142  0.00060279  0.00389472
 -0.001128    0.00501278  0.00340902  0.00133189  0.00730792 -0.00701947
 -0.00434948 -0.00665466  0.00048795  0.00062654 -0.00025399  0.00164503
  0.00642627 -0.00222312 -0.00166013 -0.00116532  0.00193651  0.00216966
 -0.00203724  0.00589405 -0.00137546 -0.00186901  0.00077087  0.00561433
 -0.00102988 -0.00444244  0.00039354 -0.00094423  0.00070008  0.00201566
 -0.00068253  0.00128079 -0.00583394  0.00433953 -0.00093023 -0.00088251
 -0.00223092  0.00332335 -0.00522417 -0.00426133 -0.00330883 -0.00380222
  0.00240783  0.00412937 -0.00301038  0.00551244]
```

Şekil 2.1. Süperonline kelimesinin 100 boyutlu vektör gösterimi



Şekil 2.2. Süperonline kelimesinin vektör uzayında gösterimi

2.2.1. Sürekli Kelime Çantası (CBOW) modeli

Sürekli Kelime Çantası (CBOW) modeli bağlama bağlı olarak hedef kelimeyi tahmin etmektedir. Örnek olarak "Senin evin bahçesi çok güzel" cümlesi için Sürekli Kelime Çantası modeli kullanılarak, 'senin', 'evin', 'çok', 'güzel' kelimeleri girdi olarak verilirken bu kelimelerin arasına hangi kelimenin en yüksek olasılıkla geleceği tahmin edilmeye çalışılmaktadır [2].

Sürekli Kelime Çantası modeli, bağlam kelimelerini kullanarak wj hedef kelimesi için, hedef kelimenin arkası ve önündeki c kadar kelimeyi ele almaktadır. Burada c pencere boyutunu temsil etmekte olup Denklem (2.1) üzerinde gösterildiği gibi hesaplanır;

$$C(wj)=\{wj-c,wj-c+1,...,wj-1,wj+1,wj+2,...,wj+c\} \quad (2.1)$$

D olarak verilen bir dokümanda K tane kelime olduğunu düşünürsek, Denklem (2.2) ve (2.3) üzerinde üzerinde ortalama log olasılığı maksimum bulunmaya çalışılır;

$$D=\{w1, w2, ..., wK\} \quad (2.2)$$

$$L(D) = \frac{1}{K}\sum_{j=c}^{K-c} \log p\left(w_j \mid w_{j-c}, \ldots, w_{j+c}\right) \quad (2.3)$$

Tahminleme kısmı softmax fonksiyonu üzerinden gerçekleştirilmektedir. Softmax fonksiyonu derin öğrenme üzerinde yapay sinir ağlarından gelen değerleri kullanarak olasılık temelli Kayıp (Loss) değeri üreten aktivasyon fonksiyonudur. Hesaplanması Denklem (2.4) üzerinde gösterildiği gibidir;

$$\log p(w_j \mid w_{j-c}, \ldots, w_{t+c}) = \frac{e^{y_{w_j}}}{\sum_i e^{y_i}} \quad (2.4)$$

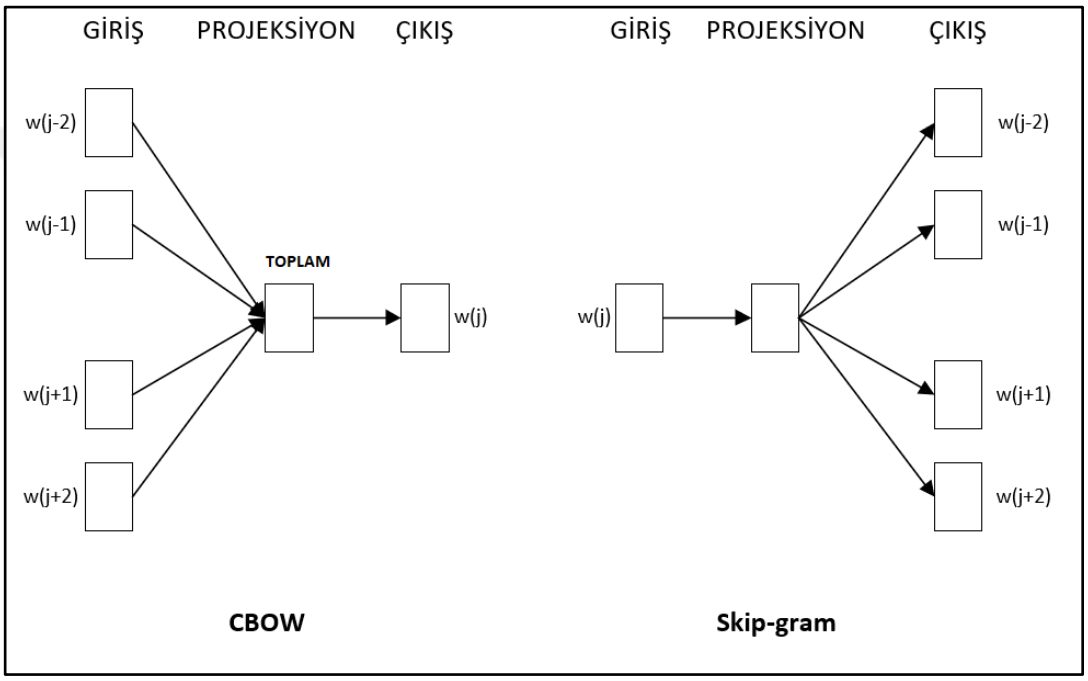
Her çıktı kelimesi i için normalize edilmemiş log olasılıkları olan y(i) için hesaplamalar Denklem (2.5) üzerindeki gibi yapılmaktadır;

$$y = b+Uh(w_{j-c}, \ldots ,w_{j+c};W) \quad (2.5)$$

Denklem (2.5) üzerinde hesaplanan U ve b parametreleri softmax fonksiyonuna ait olup, h parametresi kelime vektörlerinin birleşimi veya ortalaması ile oluşturulmuştur.

2.2.2. Skip-Gram modeli

Skip-Gram modeli verilen hedef kelimenin çevresindeki kelimeler bulmaya çalışmaktadır. CBOW modelinden temel farkı girdiler ile çıktıların yer değiştirmiş olmasıdır. Örnek olarak "Senin evin bahçesi çok güzel" cümlesi için Skip-Gram modeli kullanılarak 'bahçesi' kelimesi girdi olarak verilir ve pencere boyutu 2 verildiğinde 'senin', 'evin', 'çok', 'güzel' kelimesi çıktı olarak kullanılmaktadır. CBOW ve Skip-Gram mimarileri Şekil 2.3 üzerinde gösterilmiştir [2].



Şekil 2.3. CBOW ve Skip-Gram mimarileri [2]

D olarak verilen bir dokümanda K tane kelime olduğunu düşünürsek, CBOW modeli benzer şekilde, Denklem (2.6) ve (2.7) üzerinde gösterildiği gibi ortalama log olasılığı maksimum bulunmaya çalışılır;

$$D=\{w1, w2, ..., wK\} \tag{2.6}$$

$$L(D) = \frac{1}{K}\sum_{j=1}^{K}\sum_{-c\le i\le c, i\neq 0} \log p\left(w_{j+i}\middle|w_j\right) \tag{2.7}$$

Softmax fonksiyonu kullanılarak $p\left(w_{j+i}\middle|w_j\right)$ değeri tanımlanır. Ayrıca 2 farklı vektör içinde güncelleme işlemini gerçekleştirmektedir. Koşullu olasılık değerlerinin güncellenmesi Denklem (2.8) üzerinde gösterilmiştir;

$$p(w_m|w_i) = p(w_O|w_I) = \frac{\exp\left(v'^{T}_{w_O} v_{w_I}\right)}{\sum_{w=1}^{K} \exp\left(v'^{T}_{w} v_{w_I}\right)} \quad (2.8)$$

Ardından softmax fonksiyonu üzerinden sonuçlar üretilmektedir.

Her iki model bağlamlar ve kelimeler arasındaki ilişkileri yapay sinir ağlarını kullanarak öğrenme işlemini gerçekleştirmektedirler. Bu sayede kelimeler arasındaki ilişkileri ortaya çıkarmaktadırlar. Kelimeler ve bağlamlar arasındaki ilişkileri yakalayarak, anlamsal bilgiyi taşıyan vektörler olarak temsil edilmektedirler.

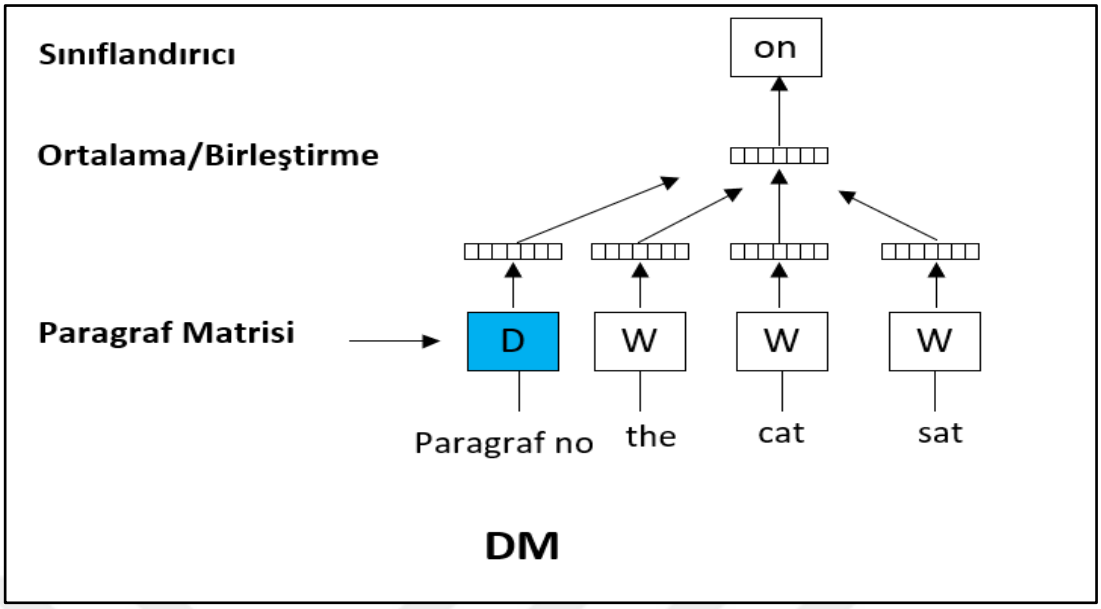
2.3. Doc2vec Yöntemi

Tomas Mikolov tarafından 2014 yılında tanıtılmıştır. Dokümanlar için geliştirmiş bir model olmak ile birlikte kalıplama vektörlerinin dokümanlar arasındaki anlamsal ve sözdizimi ilişkilerini çıkartmak için kullanılmaktadır. Temel olarak BOW yönteminin anlamsal zayıflığı giderilmeye çalışılmıştır [3].

Doc2vec yöntemi paragraf vektörlerini kullanmaktadır. Bu vektörler yapay sinir ağlarını kullanarak otomatik olarak özellik çıkarımı yapmaktadırlar. Word2vec yönteminden temel farkı kelimelerin temsil edilmesi yerine dokümanlar temsil edilmektedir. Doc2vec modeli 2 farklı modele sahiptir. Bunlar; Dağıtık Bellek (DM) ve Dağıtık Kelime Çantası (DBOW) modelidir.

2.3.1. Dağıtık Bellek (DM) modeli

Dağıtık Bellek (DM) modeli CBOW modeline benzemektedir. Şekil 3 üzerinde gösterildiği gibi ele alınmaktadır. Burada D matrisi belgeleri temsil ederken, W matrisi ise kelimeleri temsil etmek için kullanılan benzersiz sütun matrisleridir. Paragraf numarası burada hatırlatıcı vektör görevinde bulunmaktadır. Paragraf numarası ayrıca dokümanlar için ayırt edici bir özellik olarak bulunmaktadır. Kelimeler paragraf numarası ile birlikte ele alınmaktadırlar. Dağıtık Bellek Modeli Şekil 2.4'de gösterilmiştir.



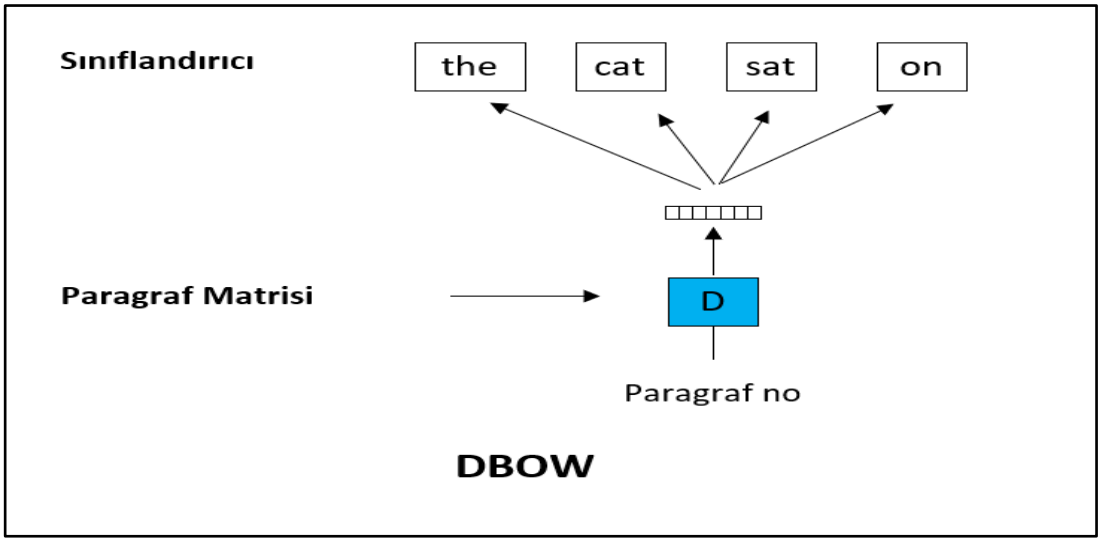
Şekil 2.4. DM ile "on" kelimesinin tahmin edilmesi [3]

DM modelinin, CBOW modelinin belgeler üzerinde uygulanması olarak düşünebiliriz. Burada temel farklılık D matrisinin eklenmesidir. D matrisi kelimelerin ne sıklıkta olduğuna ve düzenine göre tahminleme yapılmaktadır. Denklem (2.9) üzerinde gösterilmiştir;

$$y = b+Uh(w_{j-c}, \dots ,w_{j+c};W,D) \quad (2.9)$$

2.3.2. Dağıtık Kelime Çantası modeli (DBOW)

Dağıtık Kelime Çantası (DBOW) modeli, DM modelinin ters işlevi olarak gösterildiği gibi Skip-Gram modeline benzemektedir. Burada paragraf numaraları üzerinden kelimeler tahmin edilmeye çalışılır. CBOW modeline göre daha basit bir model olmak ile birlikte daha az veri depolama işlemini gerçekleştirmektedir. Şekil 2.5 üzerinde gösterildiği gibi ele alınmaktadır.



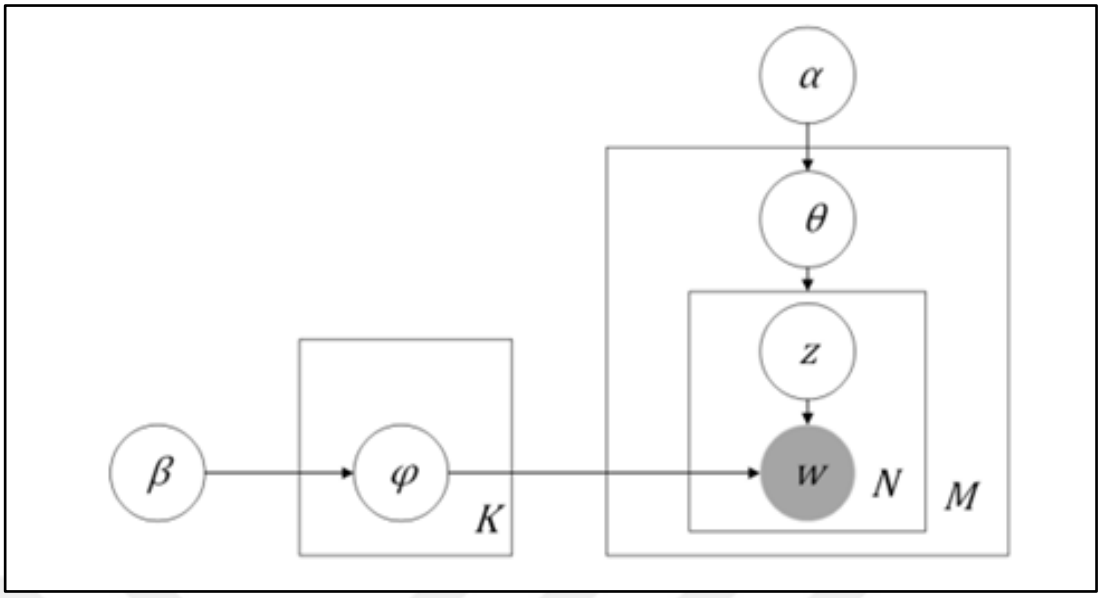
Şekil 2.5. DBOW ile “the” “cat” “sat” ve “on” kelimelerinin tahmini [3]

2.4. Gizli Dirichlet Ayırımı (LDA) Yöntemi

Konu Modelleme yöntemlerinden en bilineni olarak adlandırılan Gizli Dirichlet Ayırımı (LDA) 2003 yılında David ve diğ. tarafından tanıtılmıştır [5]. Temel olarak doküman koleksiyonlarını temsil etmektedir [25]. Kelimelerin konu olasılık dağılımları üzerinde rastgele bir araya gelerek dokümanları oluşturduğu bir modeldir. Burada konu olarak ifade edilen kelimelerin oluşturmuş olduğu bir temadır. Bu tema bir dokümanı temsil etme görevinde bulunmaktadır. Bu tema birden fazla olabilmektedir. Örnek olarak müşteriler “A” mağazasının hangi yönlerini beğeniyor, hangi yönlerini beğenmiyor? Cevap olarak düşünüldüğünde müşteriler mağazanın kendilerine olan davranışlarını, ürünlerini beğenebilirken, fiyatlarını beğenmeyebilirler. Bu örnek üzerinde cevaplar birden fazla olabilmekte ve LDA yöntemi kullanılarak bu cevaplar içerisinde istenilen sayıda tema yani konu çıkarılabilmektedir. LDA yöntemi üretici grafiksel bir modeldir [26].

Burada gizli kelimesi dokümanı meydana getiren gizli konuları keşfetmektedir [27]. Üretici model (generative) olarak belirtilmesinin nedeni ise olasılıksal olarak kelimelerin gizli değişkenler çevresinde meydana gelmesidir.

LDA yönteminin grafiksel gösterimi Şekil 2.6 üzerinde gösterilmiştir.



Şekil 2.6. LDA model gösterimi [5]

Burada gözlenebilir tek değişken w olarak temsil edilip diğer temsiller gizli değişkenler olarak gösterilmektedirler. z Değişkeni bir konu olarak ele alınırken θ değişkeni doküman konu dağılımını göstermektedir. w, z, θ Değişkenlerinin her biri α değişkenin önce simetrik bir dirichletten bağımsız olarak çizilmektedir. φ Değişkeni ise konu kelime dağılımı göstermektedir. Bu değişken de β değişkeninden önce simetrik bir dirichletten bağımsız olarak çizilmektedir. Gizli Dirichlet Ayırımı modeli bir dokümanın K başlığından oluştuğunu ve dokümanlar için üretkenlik olasılık dağılımının olduğunu varsaymaktadır;

$$p(\varphi, \theta, z, w) =$$

$$\left(\prod_{k=1}^{K} p(\varphi_k|\beta)\right)\left(\prod_{m=1}^{M} p(\theta_m|\alpha)\right)\left(\prod_{n=1}^{N} p(z_{m,n}|\theta_m)\left(w_{m,n}|z_{m,n}, \varphi_k\right)\right) \quad (2.10)$$

Model parametrelerini elde etmek için aşağıda Denklem (2.11) üzerinde verilen posterior dağılım kullanılır;

$$p(\varphi, \theta, z|w, \alpha, \beta) = \frac{p(\varphi,\theta,z,w|\alpha,\beta)}{p(w|\alpha,\beta)} \quad (2.11)$$

Bu dağılım, paydadaki gözlenen verilerin düşük olasılığı nedeniyle hesaplanamayacak kadar zor olmaktadır. Burada Model parametrelerinin elde edilmesi için belirli örnekleme yöntemleri kullanılmaktadır. Bu çalışmada Collapsed Gibbs Örnekleme yöntemi kullanılmıştır. Collapsed Gibbs Örneklemesinin temel olarak Gibbs

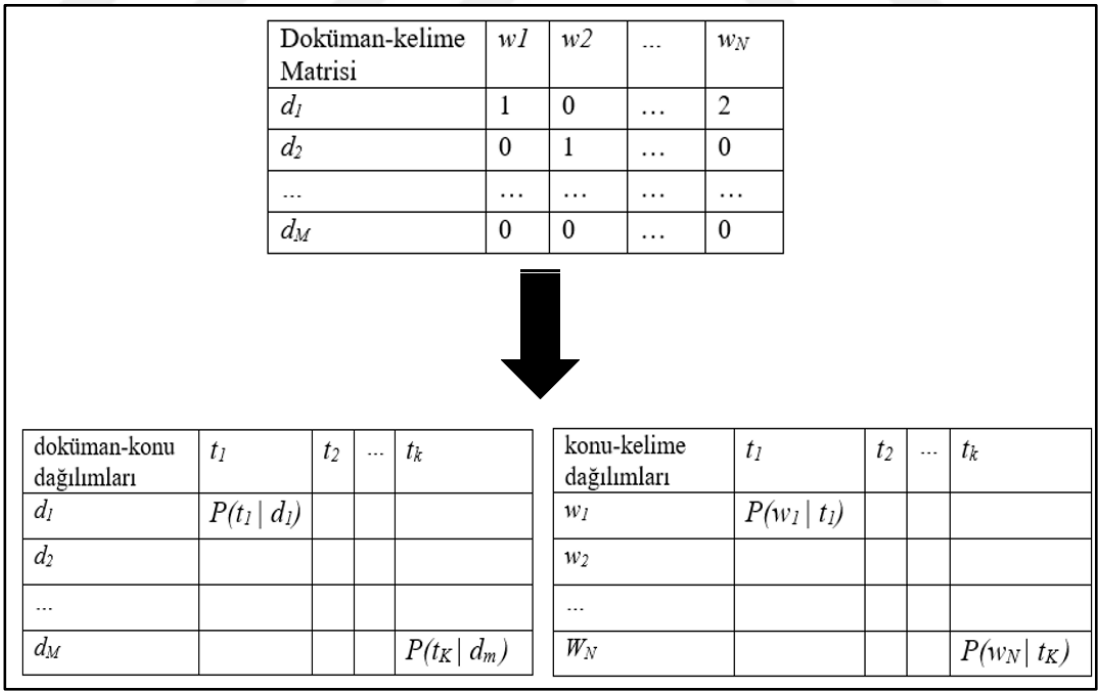
örneklemesinden farkı gizli değişkenler üzerinden örnekleme işlemini gerçekleştirmesidir.

Gizli Dirchlet Ayırımı modeli herhangi bir ön bilgiye ihtiyaç duymamaktadır. Kelimeler rastgele olasılıksal dağılımlar üzerinden belirli konuların çıkarılmasını sağlamaktadır. Derlem büyüklüğünün fazla olması Gizli Dirichlet Ayırımı modelinin iyi sonuçlar çıkarırken, performans konusunda yavaş olabilmektedir.

2.5. TopWord2vec Yöntemi

Tez çalışması kapsamında önerilen ve geliştirilen modeldir. Burada konu modelleme ile kelime kalıplama modelleri birleştirilerek oluşturulan doküman vektörleri ile dokümanlar daha iyi temsil edilmeye çalışılmıştır. Modelimizin adımları aşağıdaki gibidir.

Adım 1: Doküman konu dağılımı içeren belge vektörleri ve konu kelime dağılımını içeren kelime vektörleri Şekil 2.7'de gösterildiği gibi LDA modelinin bir sonucu olarak üretilmektedir.



Şekil 2.7. LDA model çıktıları: doküman vektörleri (doküman konu dağılımı) ve konu vektörleri (konu kelime dağılımı)

Konu modelleri, verilen bir dizi doküman $D = \{d_1, d_2, \dots, d_M\}$ ve konuların $T = \{t_1, t_2, \dots, t_K\}$ karışımı olduğu fikrine dayanmaktadır. Burada Konular kelimelerin $W = \{w_1, w_2, \dots, w_N\}$ üzerindeki olasılık dağılımlarıdır. Bu nedenle Gizli Dirichlet Ayırımı gibi konu modellerinde, bir kelimenin dokümanın konusunu nasıl temsil ettiği iki dağılım ile tanımlanabilir. Bunlar; konuların dokümanlardaki dağılımı $P(T|D)$ ve kelimelerin konulardaki dağılımı $P(W|T)$ olarak belirtebiliriz. $P(T|D)$ Olasılık terimi bir konunun dokümandaki olasılığını çıkarırken, $P(W|T)$ olasılık terimi ise bir konuyla ilgili bir kelimenin olasılığını çıkarmaktadır. Eğer doküman belirli bir konudan geliyor ise, dokümandaki belirli konunun olasılığı en yüksek olmalıdır. Bir kelime belirli bir konudan geliyorsa, bu sefer konudaki belirli kelimenin olasılığı en yüksek olmalıdır.

Örneğin "eğitim" gibi belirli bir konuda "öğrenci" kelimesi olasılığı en yüksek ise, "eğitim" konusunu yansıtmak için "öğrenci" kelimesi önemlidir. Ayrıca "öğrenci" kelimesini içeren bir dokümanda "eğitim" konu olasılığı en yüksek ise, "öğrenci" kelimesi için "eğitim" konusunu bu dokümana yansıtmak önemlidir.

Eğer n kelimesinin k konusunun altındaki olasılık dağılımının maksimum olduğu varsayılırsa, o kelimenin konusu t_{k^*} olarak tanımlanır. Bu konu bilgisine dayanarak yeni bir ağırlık derecesi olarak α_n^m Denklem (2.12) ve (2.13) önerilir. Bu denklem m dokümanında n kelimesinin önemini göstermektedir. α_n^m Değeri ne kadar yüksek olursa, m dokümanı için w_n değeri o kadar belirleyici olmaktadır;

$$t_{k^*} = \operatorname{argmax} P(w_n|t_k)\ ;\ k = 1, \dots, K \tag{2.12}$$

$$\alpha_n^m = P(w_n|t_{k^*}) \times P(t_{k^*}|d_m)\ ;\ n = 1, \dots, N \tag{2.13}$$

Adım 2: Kelime vektörleri Word2vec modeli kullanılarak elde edilmektedir. Burada öğrenen kelime vektörleri, kelime ve bağlamlar arasındaki ilişkiyi öne çıkarmaktadırlar. Bu bilgiler derlemde bulunan kelimelerin anlamsal ilişkilerini tutmaktadırlar. Word2vec modeli derlemde bulunan her bir kelimeyi belirli bir uzunlukta vektörler ile temsil etmektedirler.

Word2vec eğitiminden sonra derlemdeki kelimeler $V = \{v_1, v_2, \dots, v_N\}$ olarak temsil edilmektedirler.

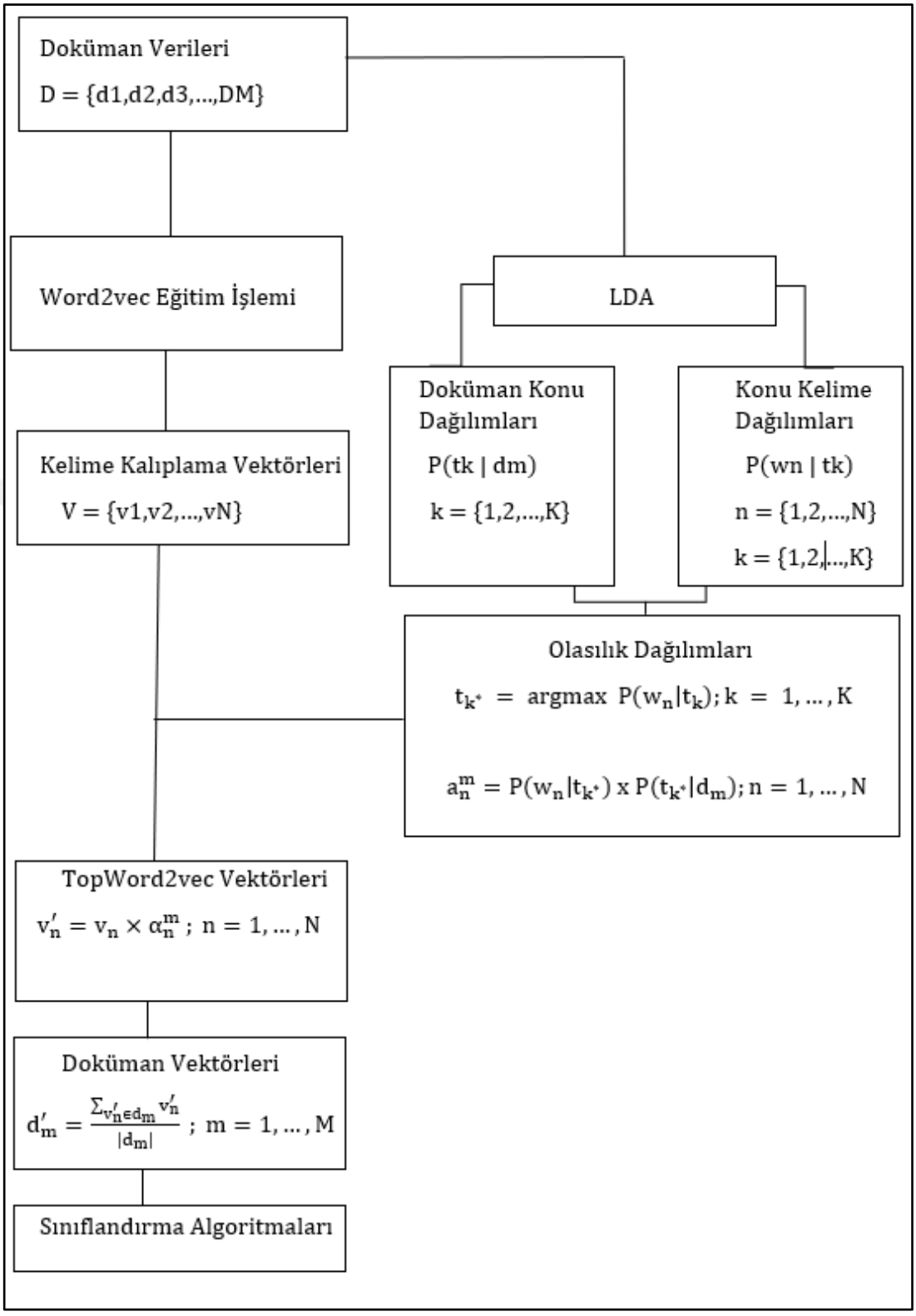
Adım 3: Kelimelerin anlamsal ilişkileri önceki 2 adımın sonuçları birleştirilerek güçlendirilmektedir. İkinci adımda elde edilen Word2vec kelime vektörlerinin sayısal değerleri ile ilk adımda verilen olasılık dağılımları çarpılarak yeni Word2vec kelime vektörleri oluşturulur. Bu vektörler $V' = \{v'_1, v'_2, \dots, v'_N\}$ olarak temsil edilmektedirler. Yeni oluşan vektörler, m dokümanı için temsil edilmektedirler. m belgesinde n kelimenin ağırlığı Denklem (2.14) üzerinde gösterildiği gibi elde edilmektedir;

$$v'_n = v_n \times \alpha_n^m \; ; \; n = 1, \dots, N \tag{2.14}$$

Adım 4: Yeni oluşan vektörlerimiz geliştirilen modelin vektörleridir. Bu vektörleri TopWord2vec vektörleri olarak tanımlamaktayız. TopWord2vec vektörlerimiz dokümanlarımızı temsil edeceği için bir normalizasyon işleminden geçirilip dokümanlarımızı temsil etmesi gerekmektedir. Bunun için doküman içerisinde bulunan TopWord2vec vektörlerimizin toplamı alınarak Denklem (2.15) üzerinde gösterildiği gibi doküman içerisinde bulunan toplam kelime sayısına bölünerek bir normalizasyon işleminden geçirilir;

$$d'_m = \frac{\sum_{v'_n \in d_m} v'_n}{|d_m|} \; ; \; m = 1, \dots, M \tag{2.15}$$

Bu adımlar derlemde bulunan tüm dokümanlar için gerçekleştirilir ve doküman vektörlerimiz geliştirilen model ile temsil edilir. Bu adımların tamamlanmasından sonra doküman vektörleri çeşitli sınıflandırıcılar ile sınıflandırılmaktadırlar. Önermiş olduğumuz TopWord2vec (Word2vec +LDA) Şekil 2.8 üzerinde gösterilmiştir.



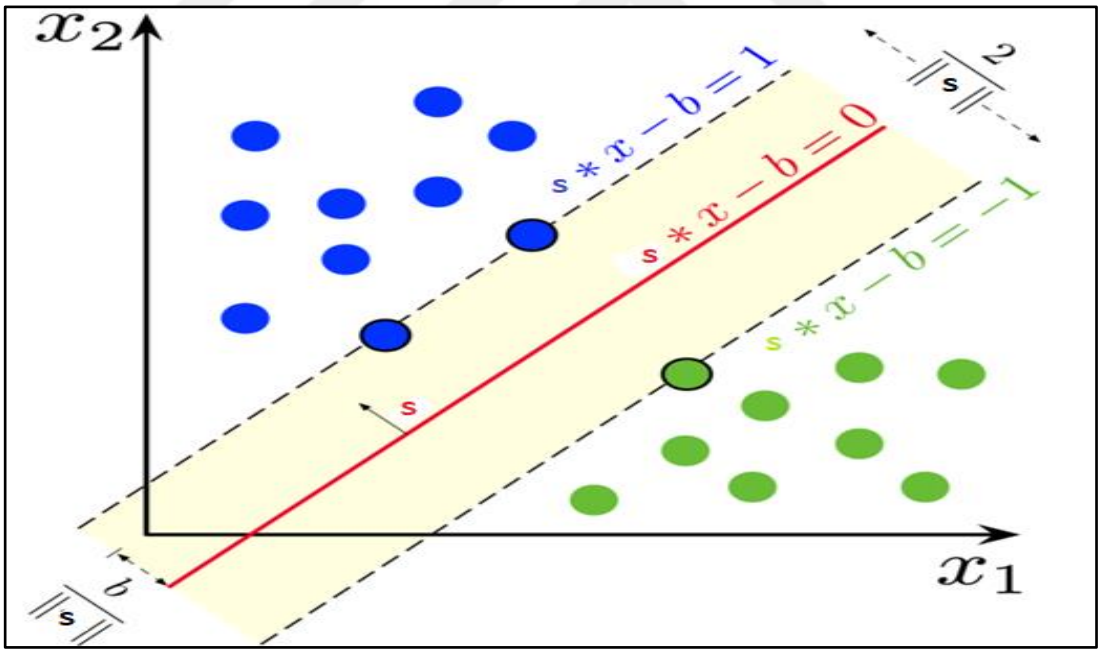
Şekil 2.8. Word2vec + LDA model

3. SINIFLANDIRMA ALGORİTMALARI

Tez çalışması içerisinde yapılan deneylerde SVM, KNN, Lojistik Regresyon, Rastgele Orman ve Bayes algoritmaları kullanılmıştır. Bu algoritmalar gözetimli öğrenme algoritmaları olup, eğitilen veriler üzerinden test verileri ile modelin değerlendirmesi sağlanmaktadır.

3.1. SVM Algoritması

SVM algoritması iyi bir gözetimli sınıflandırıcıdır [28]. Matematiksel olarak sofistik bir model olmasına rağmen birçok alanda yüksek sınıflandırma oranlarına ulaşmaktadır. SVM algoritması doğrusal ve doğrusal olmayan verilerle çalışabilmektedir fakat genellikle doğrusal verileri sınıflandırmak için kullanılmaktadırlar.



Şekil 3.1. Doğrusal SVM ile iki farklı verinin sınıflandırılması [29]

Şekil 3.1 üzerinde gösterildiği gibi doğrusal SVM modeli farklı sınıfa ait iki veriyi uzaklıklarına göre bir hiper düzlem üzerinde birbirine en yakın noktadan ayırabilme özelliğine sahip bir algoritmadır. Bu en yakın noktalar destek vektörleri olarak ifade edilmektedir. Burada s değeri ağırlık vektörünü (hiper düzlemin normali) iken *b*

değeri ise eğilim değerini göstermektedir. Sınıflandırma etiketi olarak genelde (1,-1) kullanılmakta olup Denklem (3.1) üzerinde hiper düzlem ifade edilirken Denklem (3.2) ve (3.3) üzerinde sınıf atamaları gösterilmiştir;

$$\vec{s} \cdot \vec{x} - b = 0 \tag{3.1}$$

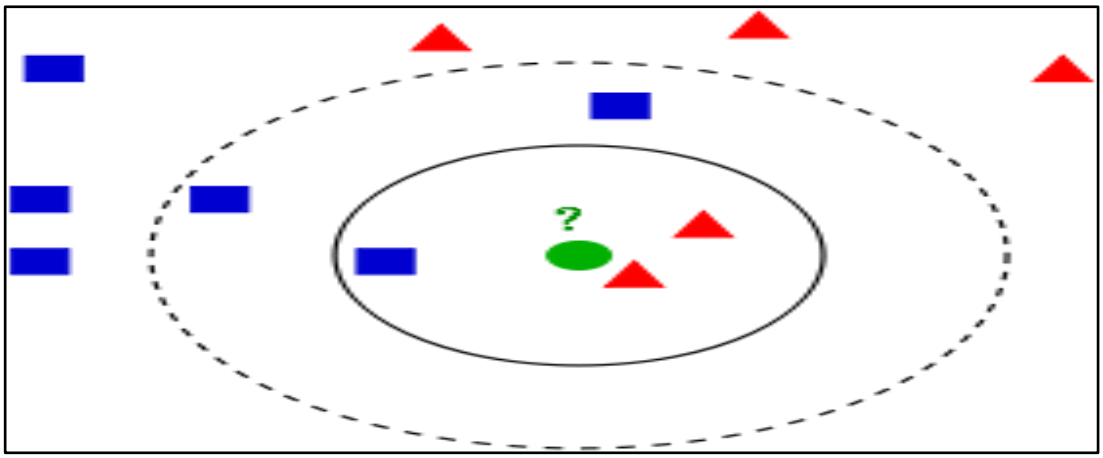
$$\vec{s} \cdot \vec{x} - b \geq 1 \quad ; 1 \text{ etiketli sınıf} \tag{3.2}$$

$$\vec{s} \cdot \vec{x} - b \leq -1 \; ; \text{-1 etiketli sınıf} \tag{3.3}$$

SVM modeli üzerinde çoklu sınıflandırmaların karar verme aşamalarında çekirdek fonksiyonları kullanılmaktadır. Bu fonksiyonlar SVM modelini iyileştirmek için sıklıkla kullanılmaktadır. Tez çalışması içerisinde lineer çekirdek fonksiyonu kullanılarak çalışmalar yapılmıştır. Radyal tabanlı fonksiyon çekirdeği lineer çekirdeği kadar iyi sonuçlar verememiştir.

3.2. KNN Algoritması

KNN algoritması tembel (lazzy) bir sınıflandırıcıdır [30]. KNN algoritması tembel bir sınıflandırıcı olması nedeniyle eğitim süresi kısa sürerken tahmin etme süresi uzun sürmektedir. Sınıflandırılacak veri noktalarının etiketli en yakın k komşusunun en sık görülenin seçilmesi prensibi ile çalışmaktadır. Yakınlıklar ölçülürken, uzaklık örneğin Öklid uzaklığı veya benzerlik işlevi olarak kullanılan Kosinüs benzerliği gibi metrikler ile hesaplanabilmektedir.



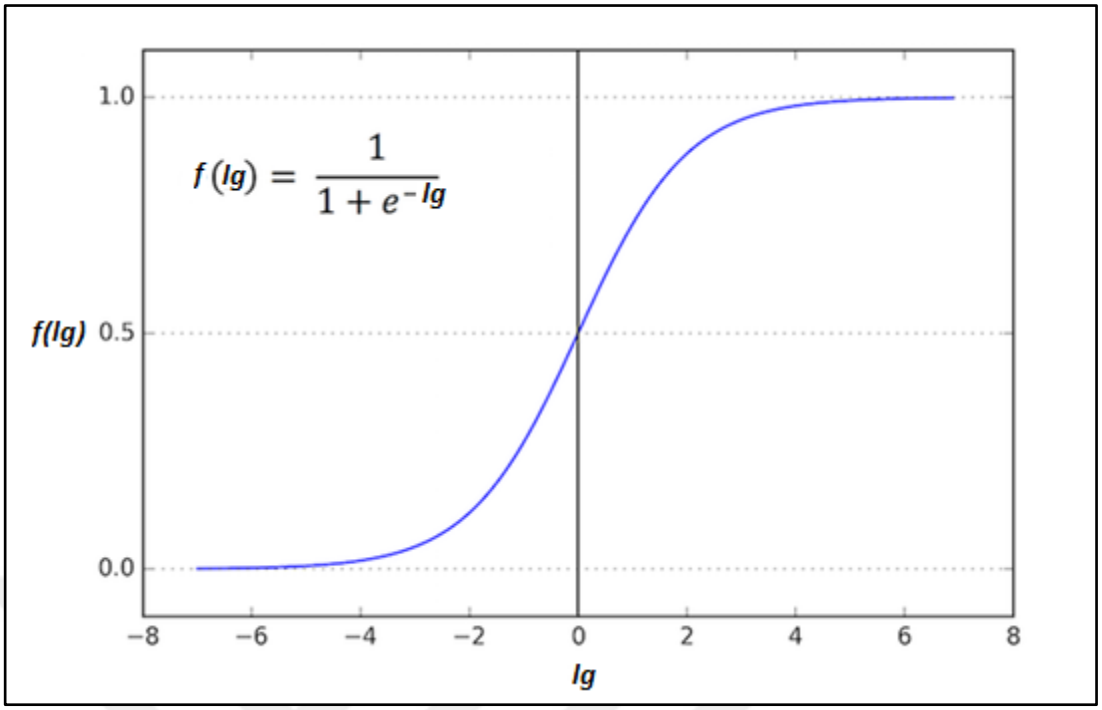
Şekil 3.2. İki farklı sınıfa ait örnek veri gösterimi [31]

Örnek olarak Şekil 3.2 üzerinde mavi kare ve kırmızı üçgen olmak üzere 2 sınıf bulunmaktadır. Yeşil daire verimiz test verisi olup hangi sınıfa atanmalıdır? Burada ilk olarak uzaklık parametremiz (k hiparparemetresi) etkili olmaktadır. k Değerimiz verilen bir noktaya en yakın komşu sayısı olarak nitelendirilmektedir. Eğer k değerimiz 3 ise yeşil dairemizin en yakın 3 komşusu ele alınacaktır ve kırmızı üçgen verisi daha fazla olacağı için kırmızı üçgen sınıfına atanacaktır. Eğer k değerimiz 5 ise yeşil dairemizin en yakın 5 komşusu ele alınacaktır ve mavi kare verimiz daha fazla olacağı için mavi kare sınıfına atanacaktır. KNN algoritmasının büyük veriler üzerinde uygulanması kolay olmasına rağmen hesaplanması maliyetlidir. Tez çalışması içerisinde k hiperparametresi 3 olarak verilmiştir.

3.3. Lojistik Regresyon Algoritması

Lojistik Regresyon, bir gözetimli sınıflandırıcı algoritmasıdır. İkili ve çoklu sınıflandırmalarda sıklıkla kullanılan ve iyi sonuçlar çıkartmaktadır. Lojistik Regresyon algoritması 0 ile 1 arasında değerler üretmektedir. Lineer Regresyon algoritmasından farkı hedef değişkenimizin olasılıklarının doğal logaritması kullanılarak eğri oluşturulur. Denklem (3.4) üzerinde gösterildiği gibi lojit fonksiyonu bulunmaktadır. Buradaki f(lg) değerimiz olasılık değerimiz iken, lg değerimiz ise Lojistik Regresyon parametrelerimizdir. Şekil 3.3 üzerinde gösterildiği gibi lojistik fonksiyon $-\infty / +\infty$ aralığında tüm değerler girdi olarak alınabildiği gibi sonuç değerlerimiz sadece 0 ile 1 arasında değerler üretmektedir. Şekil 3.3 üzerinde 0 ile 0.5 arasında değer döndürenler bir sınıf olurken, 0.5 ile 1 arasında sonuç üretilenler diğer sınıfa atanmaktadır;

$$f(lg) = (1 / 1+e^{(-lg)}) \quad (3.4)$$

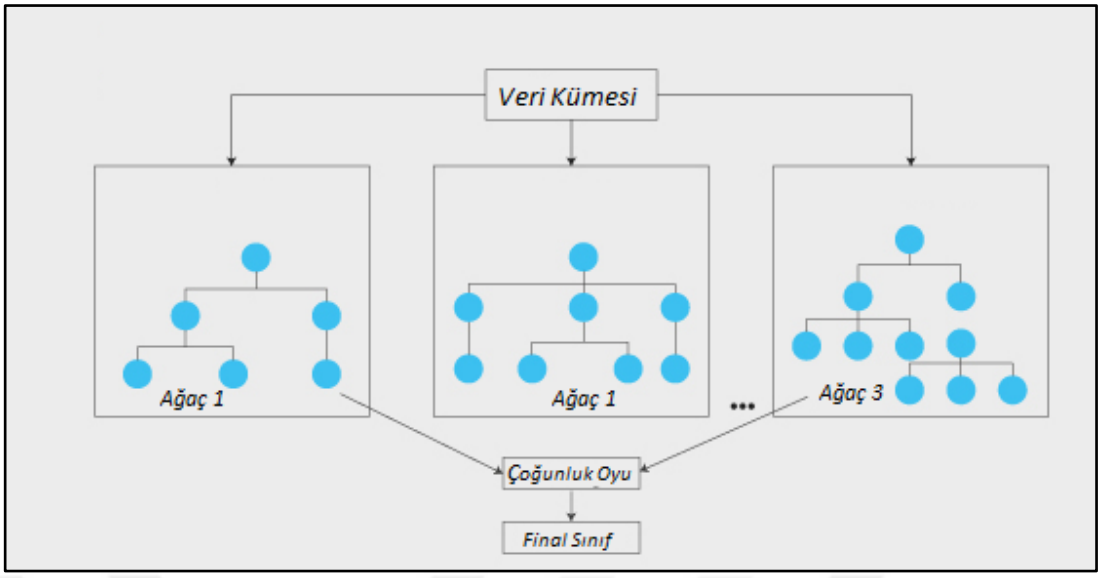


Şekil 3.3. Lojistik Regresyon algoritması ile sınıfların ayrılmasının gösterimi [32]

Logistik Regresyon algoritmasının temel amacı bağımlı ile bağımsız değişkenler arasındaki ilişkiyi en az sayıda parametre ile kurmayı amaçlamaktadır. Kategorik değişkenlerin sınıflandırılmasında sıklıkla kullanılmaktadır.

3.4. Rastgele Orman Algoritması

Karar ağacı merkezli bir sınıflandırma ve regresyon algoritmasıdır [33]. Karar ağaçlarında düğümler üzerinde bölünmeler gerçekleştirilirken en kullanışlı düğüm üzerinden bölünmeler gerçekleştirilir. Rastgele Orman algoritmasında ise çoklu ağaç yapısı kullanıldığı için bölünmeler farklı senaryolar ile gerçekleştirilir ve farklı senaryolar sınıflandırma için kullanılacaksa çoğunluk oyu, regresyon için kullanılacaksa ortalamaları alınarak işlemler gerçekleştirilir. Şekil 3.4 üzerinde örnek olarak 3 farklı karar ağacının Rastgele Orman ile sınıflandırılması gösterilmiştir. Burada farklı karar ağaçlarından elde edilen sonuçlar üzerinden bir oylama yapılarak doğru sınıf ataması temsili olarak gösterilmiştir.

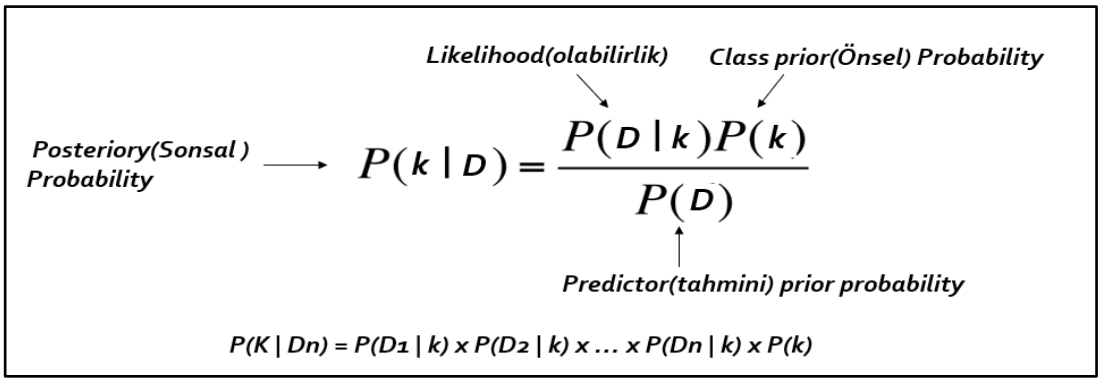


Şekil 3.4. Rastgele Orman algoritması sınıflandırma gösterimi [34]

Rastgele Orman algoritması çoğu zaman karar ağaçlarından daha yüksek başarım çıkarır ve çoklu karar ağacı yapısı sayesinde overfit sorununun üstünden gelmektedirler.

3.5. Naif Bayes Algoritması

Bir Naif Bayes sınıflandırıcısının temeli şartlı olasılığa dayanmaktadır. Verilerin hangi sınıflara ait olacağı her bir sınıf için hesaplanır, ardından en yüksek olasılığa sahip sınıflara veriler eklenir. Burada D dokümanları k sınıflarına Denklem (3.5) ve (3.6) üzerinde gösterildiği gibi olasılık bazlı olarak atanmaktadır. Burada Sonsal olasılık dokümanın en yüksek olasılık oranını çıkardığı sınıfı atandığını belirtmekte olup Olabilirlik değeri dokümanın sınıfa ait olma olasığını vermektedir.



Şekil 3.5. Naif Bayes algoritması gösterimi

Naif Bayes sınıflandırıcıları metinler üzerinde sıklıkla kullanılmaktadırlar. Tez çalışması içerisinde Denklem (3.5) ve (3.6) üzerinde gösterilen Bernoulli Naif Bayes ve Gaussian Naif Bayes modelleri kullanılmıştır. Gaussian Naif Bayes modeli ile verilerimizin Gaussian dağılımına uyduğunu kabul ederiz. Bernoulli Naif bayes modeli ile özelliklerimiz birbirinden bağımsız olarak düşünülüp ikili sınıflandırma problemlerinde sıklıkla kullanılmaktadır. Ayrıca Çoklu Naif Bayes sınıflandırıcıya göre daha basit bir modeldir;

$$p(x = v \mid K_k) = \frac{1}{\sqrt{2\pi\ \sigma_k^2}} e^{-\frac{(v-u_k)^2}{2\sigma_k^2}} \quad \text{; Gaussian Naif Bayes} \tag{3.5}$$

$$p(x \mid K_k) = \prod_{i=1}^{n} p_{ki}^{x_i}(1 - p_{ki})^{(1-x_i)} \quad \text{; Bernoulli Naif Bayes} \tag{3.6}$$

Naif Bayes algoritmaları metin sınıflandırmalarında sıklıkla uygulanmakta olup yüksek başarım özelliğine sahiplerdir [35].

4. DENEYSEL ÇALIŞMA

4.1. Performans Ölçütleri

Sınıflandırma algoritmalarının değerlendirilmesi için doğruluk oranı ve perplexity değeri baz alınmıştır. F-ölçüsü değerleri doğruluk değerlerine yakın olduğu için ele alınmamıştır. Doğruluk oranı değeri hesaplanırken kullanılan TN, TP, FP ve FN değerleri sırasıyla doğru negatif, doğru pozitif, yanlış pozitif ve yanlış negatif değerlerini ifade etmektedir. Bu değerler karışıklık matrisi (confusion matrix) üzerinden elde edilmektedirler. Şekil 4.1 üzerinde örnek olarak pozitif ve negatif sınıflandırma için oluşturulan karışıklık matrisi verilmiştir.

		Gerçek Değerler: Pozitif	Gerçek Değerler: Negatif
Tahmin Edilen Değerler	Pozitif	TP	FP
	Negatif	FN	TN

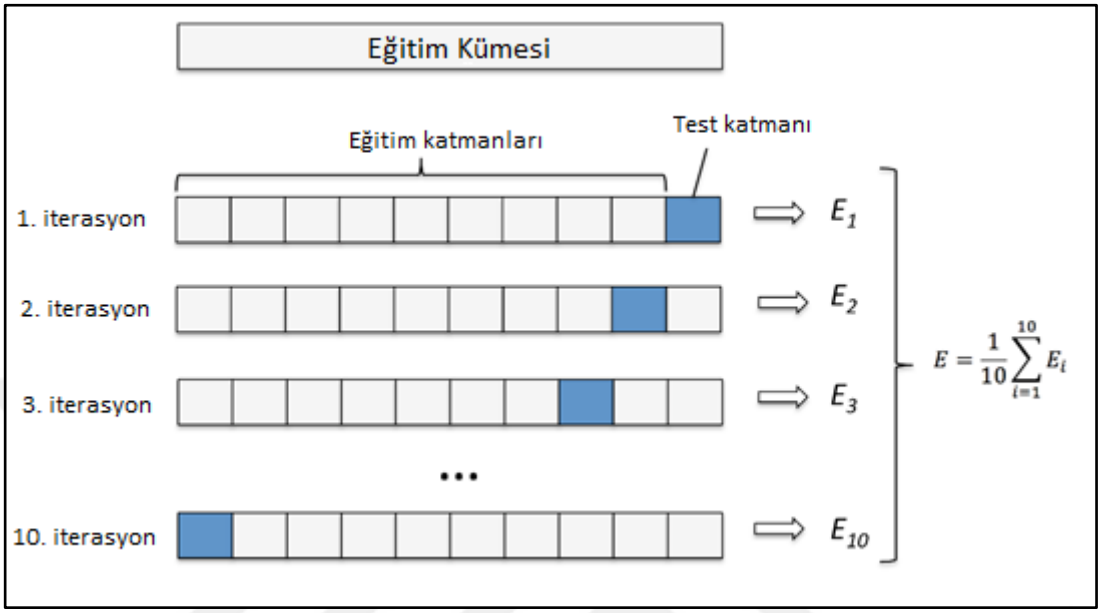
Şekil 4.1. Karışıklık matrisi gösterimi

Doğruluk oranı Denklem (4.1) üzerinde görülebileceği gibi doğru etiketlenmiş verilerin bütün etiketli verilere bölünmesi olarak gerçekleştirilmekte olup, sınıflandırma algoritmalarını değerlendirmek için sıklıkla kullanılmaktadırlar;

$$\text{Doğruluk oranı} = \frac{TN+TP}{TP+FP+FN+TN} \tag{4.1}$$

Doğruluk oranları ele alınırken 10-kat çapraz geçerleme yöntemi kullanılmıştır. 10-kat çaprazlama yöntemi ile veriler 10 parçaya bölünmektedir. Her 9 parça eğitim verisi olurken geriye kalan 1 parça test veri kümesi olarak kullanılmaktadır. Burada her

iterasyonda farklı eğitim ve test verileri oluşturulduğu için farklı doğruluk oranları çıkarmaktadır. Çıkan doğruluk oranlarının ortalaması modelimizin doğruluk oranı olarak kullanılmaktadır. Şekil 4.2 üzerinde temsili olarak gösterilmiştir.



Şekil 4.2. 10-Kat çapraz geçerleme gösterimi [36]

LDA modelinin değerlendirilmesi için şaşkınlık (perplexity) değeri hesaplanmıştır. Şaşkınlık değeri Denklem (4.2) üzerinde gösterildiği gibi bir modelin daha önce görmediği yeni veriler üzerinden ne kadar şaşırdığını yakalayan bir konu ölçüm metodudur. Burada M toplam doküman sayısı olarak ifade edilirken, N ise d belgesinde bulunan toplam kelime sayısını ifade etmektedir;

$$\text{perplexity}(D_{\text{test}}) = \exp\{-\frac{\sum_{d}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d}\} \quad (4.2)$$

Geliştirilen TopWord2vec modeli üzerinde uygun konu seçim işlemi için şaşkınlık değerleri kullanılmıştır.

4.2. Veri Kümeleri ve Önişlem

Kullanılan veriler eğitim ve sınıflandırma işlemlerinde Türkçe ve İngilizce olarak ayrılmıştır.

Türkçe veriler içerisinde bulunan 1150 Haber, 3000 Tweet ve 10 Milyon Tweet veri kümelerini Yıldız Teknik Üniversitesinin Kemik Dil Grubu[1] üzerinden alınmıştır. Ayrıca TTC-3600 [37] ve 9127 Tweet [2] veri kümeleri de sınıflandırma algoritmalarında kullanılmıştır. Türkçe veriler haber ve tweet verileri olarak iki farklı şekilde ele alınmıştır. Haber veri kümesi olarak 1150 Haber ve TTC-3600 verileri sınıflandırıcı olarak kullanılmıştır.

1150 Haber veri kümesi ekonomi, magazin, sağlık, siyasi ve spor olarak 5 sınıftan oluşmaktadır. Her sınıfta 230 veri olmak ile birlikte toplamda 1150 adet veri bulunmaktadır.

TTC-3600 Haber veri kümesi ekonomi, kültür-sanat, sağlık, politika, spor ve teknoloji olarak ayrılıp her sınıf içerisinde 600 veri olmak ile birlikte toplamda 2600 adet veri bulunmaktadır.

Tweet veri kümesi olarak 3000 Tweet ve 9127 Tweet verileri sınıflandırıcı olarak kullanılmıştır. 3000 Tweet veri kümesi olumlu, olumsuz ve nötr tweet olmak üzere 3 sınıftan oluşmaktadır. Tweetlerin 756 tanesi olumlu, 1287 tanesi olumsuz ve 957 tanesi nötr olup toplamda 3000 adet tweet verisi vardır.

9127 Tweet veri kümesi olumlu ve olumsuz olmak üzere 2 sınıftan oluşmaktadır. Tweetlerin 5174 tanesi olumlu 3953 tanesi olumsuz olup 9127 adet tweet verisi vardır.

10 Milyon Tweet veri kümesi diğer Türkçe verilerle birleştirilip toplamda 10.073.869 adet veri eğitim sırasında kullanılmıştır.

Türkçe veri kümelerinin özellikleri Tablo 4.1 üzerinde gösterilmiştir.

[1] www.kemik.yildiz.edu.tr/?id=28
[2] https://github.com/MatBilML/turkish-nlp-datasets

Tablo 4.1. Türkçe veri kümesi özellikleri

Veri Kümesi	Tanım	Sınıf Sayısı	Derlem Boyutu (Tekil Kelime Sayısı)	Ortalama Doküman Uzunluğu (Tekil Kelime Sayısı)
1150 Haber	1150 Adet Türkçe haber verisi	5	50.973	218
3000 Tweet	3.000 Adet Türkçe tweet verisi	3	12.040	10
9127 Tweet	9.127 Adet Türkçe tweet verisi	2	26.682	16
TTC-3600	3.600 Adet Türkçe haber verisi	6	104.774	239

Tablo 4.1'de görüldüğü gibi, tweet verileri ortalama 12 tekil kelimeden oluşurken, haber verileri ortalama 228 kelimeden oluşmaktadır. Bu sayede geliştirilen model hem kısa metinler hem de uzun metinler üzerinde karşılaştırmalar yapılarak test edilmiştir.

İngilizce veriler, Türkçe verilerin aksine sadece tweet verilerinden oluşturulmuştur. Sent Collection[3] verileri olarak Archeage, Iphone6, Hobbit kullanılıp ek olarak Sts-gold [38] veri kümesi ile 4 farklı tweet verisi sınıflandırıcı olarak kullanılmıştır.

Tablo 4.2 üzerinde kullanılan İngilizce veri kümelerinin özellikleri verilmiştir.

Tablo 4.2. İngilizce veri kümesi özellikleri

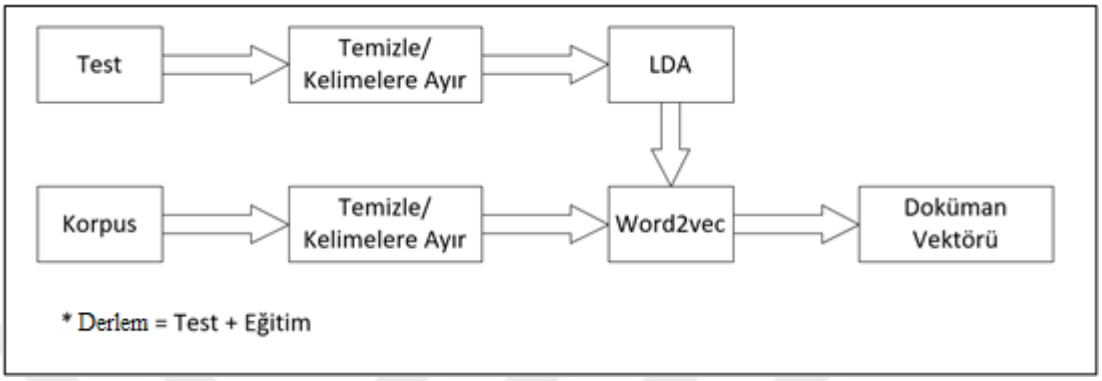
Veri Kümesi	Pozitif	Negatif	Toplam Veri	Derlem Boyutu (Tekil Kelime Sayısı)	Ortalama Doküman Uzunluğu (Tekil Kelime Sayısı)
Archeage	724	994	1.718	2.945	13
Iphone6	371	161	532	1.439	10
Hobbit	354	168	532	1.311	13
Sts-gold	632	1.402	2.034	4.609	12

İngilizce oluşturulan modellerde 1.682.876 adet ingilizce tweet verisi diğer İngilizce verilerle birleştirilip sadece eğitim sırasında kullanılmıştır.

TopWord2vec deneylerinde doğru kelime vektörlerinin çıkarılması için derlem verileri üzerinden kelime kalıplama (embedding) eğitiminin gerçekleştirilmesi gerekmektedir. TopWord2vec deneylerinde dokümanlar kelimelere ayrılıp (tokenize),

[3] http://www.dt.fee.unicamp.br/~tiago//sentcollection/

sayısal karakterler, noktalama işaretleri ve durma kelimeleri (stopwords) kaldırılmıştır. Test verisi, LDA modeli ile Word2vec modeline gönderilir. Ardından Word2vec vektörleri ile birleştirilerek doküman vektörleri oluşturulur. Her bir test verisi için bu işlem tekrar edilmiştir. Şekil 4.3. üzerinde gösterilmiştir.



Şekil 4.3. Word2vec+LDA genel gösterim

Programlama dili olarak python dili tercih edilip, python üzerinde geniş bir çatıya sahip Anaconda[4] aracı tercih edilmiştir. Makine öğrenmesi kütüphanesi olarak Scikit-Learn [39] ile derin öğrenme kütüphanesi olarak Gensim [40] tercih edilmiştir.

4.3. Parametre Seçimi

Kullanılan algoritmalar için parametre seçimi, hipotezimizi modellemek için makine öğrenmesinde önemli bir işlemdir.

Deneylerde kullanılan algoritmalarda optimum parametreler seçilmiştir. Word2vec ve Doc2vec yöntemleri kullanılırken sırasıyla CBOW ve DM modelleri kullanılmıştır. Temel parametreler; vektör boyutu:100, pencere uzunluğu:5, minimum frekans:1, epok:5 ve negatif örnekleme:5 olarak kullanılmıştır.

LDA yöntemi üzerinde kullanılan parametreler; toplam konu sayısı:10, alfa=50/toplam konu sayısı, beta=0.01 ve iterasyon=1000 olarak verilmiştir. Ayrıca uygun konu sayısının seçilmesi için şaşkınlık (perplexity) ölçümü ile doğru konu sayısı seçilmiştir. Tablo 4.3 ve Tablo 4.4 üzerinde görüldüğü gibi hem Türkçe hem İngilizce veri kümelerinde konu sayısı 10 olarak verildiği zaman en iyi sonuçlar elde edilmiştir.

[4] https://www.anaconda.com/

Tablo 4.3. Türkçe veri kümelerinde farklı konular ile perplexity sonucu

Veri Kümesi	10 Konu	20 Konu	50 Konu	100 Konu
1150 Haber	**-48.56**	-55.67	-73.77	-64.83
3000 Tweet	**-64.62**	-70.98	-77.56	-80.52
9127 Tweet	**-35.95**	-40.14	-41.76	-42.84
TTC-3600	**-30.87**	-35.52	-40.28	-36.11

Tablo 4.4. İngilizce veri kümelerinde farklı konular ile perplexity sonucu

Veri Kümesi	10 Konu	20 Konu	50 Konu	100 Konu
Archeage	**-38.42**	-46.34	-61.64	-73.85
Iphone6	**-65.46**	-73.34	-100.65	-101.03
Hobbit	**-48.49**	-58.44	-76.16	-85.92
Sts-gold	**-50.51**	-61.42	-79.10	-90.83

4.4. Deneysel Sonuçlar

Doc2vec modeli kullanılarak eğitilen veri sayısının arttırılmasının model performasına etkileri incelenmiştir. İlk olarak 1150 Haber veri kümesi sınıflandırıcı veri kümesi olarak kullanılmıştır. Model eğitiminde Tablo 4.5. üzerinde görülebileceği gibi 20 Milyon'a kadar kademeli olarak artan tweet veri kümesi kullanılmıştır. Vektör boyutu 100 olarak verilip, kök indirgeme (stemmer) işlemi Snowball Stemmer [5] aracı kullanılarak gerçekleştirilmiştir. KNN, GNB (Gaussian Naive Bayes) ve SVM sınıflandırma algoritmaları kullanılmıştır [23].

[5] http://snowballstem.org/

Tablo 4.5. Farklı tweet verileri ile Doc2vec model sonuçları

Veri Sayısı	KNN	GNB	SVM
20 Milyon	%81	**%69**	%89
10 Milyon	%83	**%69**	**%90**
5 Milyon	**%84**	%68	%88
1 Milyon	%81	%67	%88
500 Bin	%80	%64	%86
100 Bin	%61	%55	%74

Tablo 4.5 üzerinde görüldüğü üzere, veri sayısının artması ile model eğitimleri daha iyi yapılabildiği için, sınıflandırma algoritmalarındaki doğruluk oranları da yükselmektedir.

TopWord2vec deneylerimizde gizli konuları çıkarmak için LDA ve kelime vektörlerini oluşturmak için Word2vec algoritması kullanılmıştır. Konu kelime vektörlerini oluşturmak için hem kısa hemde uzun metinler kullanılmıştır. TopWord2vec yaklaşımı ile öğrenen gizli doküman gösterimleri, Doğrusal (lineer) SVM, KNN(k=3), Rastgele Orman, Bernoulli Naif Bayes ve Lojistik Regresyon gibi iyi bilinen denetimli öğrenme algoritmaları ile sınıflandırılmıştır. Sınıflandırma sonuçlarında F-ölçüsü ve doğruluk değerlerinin değerleri çok yakın olması nedeniyle doğruluk oranları kullanılmıştır. Elde edilen sonuçlar Türkçe ve İngilizce veri kümeleri için ayrı olarak verilmiştir. Tablo 4.6 üzerinde Doğrusal SVM (Lineer SVM) algoritma sonuçları verilmiştir.

Tablo 4.6. Türkçe veri kümelerinde SVM doğruluk oranları

Veri Kümesi	Sınıflandırma Oranı (%)	
	TopWord2vec	Doc2vec
1150 Haber	**90.26**	88.43
3000 Tweet	**58.79**	47.60
9127 Tweet	**79.84**	74.82
TTC-3600	**87.39**	86.50

Tablo 4.6 incelendiğinde SVM algoritması kullanılarak TopWord2vec vektörü ile elde edilen sonuçların tüm Türkçe veri kümeleri üzerinde Doc2vec yöntemi kullanılarak elde edilen sonuçlardan iyi olduğu görülmüştür.

Tablo 4.7 üzerinde KNN algoritmasının sonuçları verilmiştir. KNN algoritmasının sonuçları incelendiğinde önermiş olduğumuz TopWord2vec yöntemi ile Türkçe veri kümeleri üzerinde minumum %6 oranında artış sağlandığı görülmüştür.

Tablo 4.7. Türkçe veri kümelerinde KNN doğruluk oranları

Veri Kümesi	**Sınıflandırma Oranı (%)**	
	TopWord2vec	**Doc2vec**
1150 Haber	**88.52**	82.96
3000 Tweet	**49.36**	33.80
9127 Tweet	**77.45**	64.91
TTC-3600	**84.67**	78.42

Rastgele Orman algoritmasının sınıflandırma sonuçları Tablo 4.8 üzerinde verilmiştir. Rastgele Orman algoritması ile önermiş olduğumuz yöntem Doc2vec yönteminden daha yüksek sonuçlar vermiştir. Türkçe veri kümeleri üzerinde TopWord2vec yöntemi minimum %6 artış sağlanmıştır.

Tablo 4.8. Türkçe veri kümelerinde Rastgele Orman doğruluk oranları

Veri Kümesi	**Sınıflandırma Oranı (%)**	
	TopWord2vec	**Doc2vec**
1150 Haber	**85.04**	77.22
3000 Tweet	**49.23**	41.29
9127 Tweet	**76.21**	70.06
TTC-3600	**82.61**	74.19

Bernoulli NB (Naif Bayes) algoritmasının sınıflandırma sonuçları Tablo 4.9 üzerinde verilmiştir. Önceki sınıflandırma algoritmalarında olduğu gibi TopWord2vec yöntemi Doc2vec yönteminden daha iyi sonuçlar vermiştir.

Tablo 4.9. Türkçe veri kümelerinde Bernoulli NB doğruluk oranları

Veri Kümesi	Sınıflandırma Oranı (%)	
	TopWord2vec	**Doc2vec**
1150 Haber	**87.48**	87.13
3000 Tweet	**53.91**	40.43
9127 Tweet	**76.33**	64.48
TTC-3600	**81.31**	80.89

Lojistik Regresyon algoritmasının sınıflandırma sonuçları Tablo 4.10 üzerinde verilmiştir. Tablo 4.10 incelendiğinde TopWord2vec yöntemi Doc2vec yönteminden daha iyi sonuçlar vermiştir.

Tablo 4.10.Türkçe veri kümelerinde Lojistik Regresyon doğruluk oranları

Veri Kümesi	Sınıflandırma Oranı (%)	
	TopWord2vec	**Doc2vec**
1150 Haber	**89.13**	84.09
3000 Tweet	**58.99**	47.26
9127 Tweet	**79.33**	75.41
TTC-3600	**86.50**	86.28

Tablo 4.9 ve 4.10 detaylı incelendiğinde TTC-3600 veri kümesi üzerinde TopWord2vec ve Doc2vec yöntem sonuçları Lojistik Regresyon ve Bernoulli Naif Bayes algoritmalarında yakın değerler çıkmıştır. TTC-3600 veri kümesi doküman boyutu uzun olmak ile birlikte Türkçe model eğitiminde sıklıkla tweet veri kümeleri kullanılması sonucu TTC-3600 veri kümesi üzerinde Lojistik Regresyon ve Bernoulli Naif Bayes algoritmalarının başarım oranları önermiş olduğumuz model ile %1 oranından az olmak ile birlikte yine de başarılı olmuştur.

Genel olarak sınıflandırma algoritmalarının Türkçe veri kümeleri üzerindeki sonuçları incelendiğinde Doc2vec yerine TopWord2vec yöntemi kullanıldığında daha yüksek sonuçlar çıktığı görülmüştür

Tablo 4.11.Türkçe veri kümelerinde yöntem karşılaştırması

Veri Kümesi	Yöntem	Sınıflandırma Oranı (%)				
		SVM	KNN	Rastgele Orman	Bernoulli Naif Bayes	Lojistik Regresyon
1150 Haber	TopWord2vec	**90.26**	88.52	85.04	87.48	89.13
1150 Haber	Doc2vec	**88.43**	82.96	77.22	87.13	84.09
3000 Tweet	TopWord2vec	58.79	49.36	49.23	53.91	**58.99**
3000 Tweet	Doc2vec	**47.60**	33.80	41.29	40.43	47.26
9127 Tweet	TopWord2vec	**79.84**	77.45	76.21	76.33	79.33
9127 Tweet	Doc2vec	**74.82**	64.91	70.06	64.48	75.41
TTC-3600	TopWord2vec	**87.39**	84.67	82.61	81.31	86.50
TTC-3600	Doc2vec	86.50	78.42	74.19	80.89	**86.28**

Tablo 4.11 üzerinde Türkçe veri kümeleri üzerinde yöntem karşılaştırmaları sınıflandırma oranları ile özet olarak verilmiştir. Tablo 4.11 üzerinde Türkçe veri kümeleri üzerinde en yüksek başarımları sağlayan algoritmalar SVM ve Lojistik Regresyon algoritması olduğu görülmektedir. Sınıflandırma algoritmalarından bağımsız olarak yöntem karşılaştırması yapıldığında önermiş olduğumuz Topword2vec Modeli Doc2vec modelinden başarılı olduğu görülmektedir.

Aynı sınıflandırma algoritmaları kullanılarak İngilizce veri kümeleri üzerinde testler gerçekleştirilmiştir.

Tablo 4.12 üzerinde İngilizce tweet veri kümeleri ile doğrusal SVM algoritmasının sınıflandırma sonuçları verilmiştir. Sonuçlar incelendiğinde Archeage, Sts-gold ve Hobbit tweet veri kümelerinde artış olurken, Iphone6 veri kümesinde düşüş gözlemlenmiştir.

Tablo 4.12.İngilizce veri kümelerinde SVM doğruluk oranları

Veri Kümesi	Sınıflandırma Oranı (%)	
	TopWord2vec	Doc2vec
Archeage	**77.34**	75.15
Iphone6	71.75	**73.11**
Hobbit	**77.12**	73.73
Sts-gold	**79.12**	78.17

Tablo 4.13 üzerinde İngilizce tweet veri kümeleri kullanılarak KNN algoritmasının sınıflandırma sonuçları verilmiştir. Sonuçlar incelendiğinde TopWord2vec yöntemi Doc2vec yönteminden daha iyi sonuçlar göstermiştir. Özellikle Archeage veri kümesinde %9 oranında artış görülmüştür.

Tablo 4.13.İngilizce veri kümelerinde KNN doğruluk oranları

Veri Kümesi	**Sınıflandırma Oranı (%)**	
	TopWord2vec	**Doc2vec**
Archeage	**74.80**	66.06
Iphone6	**73.08**	71.82
Hobbit	**75.05**	70.08
Sts-gold	**73.76**	69.32

Tablo 4.14 üzerinde İngilizce tweet veri kümeleri kullanılarak Rastgele Orman algoritmasının sınıflandırma sonuçları verilmiştir. TopWord2vec yöntemi Doc2vec yönteminden daha yüksek sonuçlar vermiştir. Özellikle Hobbit veri kümesi üzerinde %11 oranında artış gözlemlenmiştir.

Tablo 4.14.İngilizce veri kümelerinde Rastgele Orman doğruluk oranları

Veri Kümesi	**Sınıflandırma Oranı (%)**	
	TopWord2vec	**Doc2vec**
Archeage	**70.42**	68.05
Iphone6	**70.43**	66.16
Hobbit	**76.22**	65.13
Sts-gold	**73.11**	69.47

Tablo 4.15 üzerinde İngilizce tweet veri kümeleri kullanılarak Bernoulli NB (Naif Bayes) algoritmasının sınıflandırma sonuçları verilmiştir. TopWord2vec yöntemi Doc2vec Yönteminden daha yüksek sonuçlar vermiştir. Özellikle Iphone6 veri kümesi üzerinde %10 oranında bir artış gözlemlenmiştir.

Tablo 4.15.İngilizce veri kümelerinde Bernoulli NB doğruluk oranları

Veri Kümesi	Sınıflandırma Oranı (%)	
	TopWord2vec	**Doc2vec**
Archeage	**71.00**	63.88
Iphone6	**70.64**	61.28
Hobbit	**72.04**	62.04
Sts-gold	**73.41**	69.47

Tablo 4.16 üzerinde İngilizce tweet veri kümeleri kullanılarak Lojistik Regresyon algoritmasının sınıflandırma sonuçları verilmiştir. TopWord2vec yöntemi Doc2vec Yönteminden daha yüksek sonuçlar vermiştir. En düşük Iphone6 veri kümesi üzerinde artış gözlemlenmiştir.

Tablo 4.16.İngilizce veri kümelerinde Lojistik Regresyon doğruluk oranları

Veri Kümesi	Sınıflandırma Oranı (%)	
	TopWord2vec	**Doc2vec**
Archeage	**77.90**	75.44
Iphone6	**71.18**	70.66
Hobbit	**78.50**	71.24
Sts-gold	**79.40**	77.29

Tablo 4.17 üzerinde İngilizce veri kümeleri üzerinde yöntem karşılaştırmaları sınıflandırma oranları ile özet olarak verilmiştir. Tablo 4.17 üzerinde İngilizce veri kümeleri üzerinde en yüksek başarımları sağlayan algoritmalar SVM ve Lojistik Regresyon algoritması olduğu görülmektedir. Sınıflandırma algoritmalarından bağımsız olarak yöntem karşılaştırması yapıldığında Iphone6 veri kümesi üzerinde Doc2vec algoritması TopWord2vec modelinden %1 oranından az olmak ile birlikte başarılı olmuştur. Diğer İngilizce veri kümeleri üzerinde TopWord2vec Modeli Doc2vec modelinden başarılı olduğu görülmektedir. Iphone6 veri kümesi TopWord2vec modeli üzerinde en yüksek başarım oranına KNN algoritması ile ulaşmıştır.

Genel olarak sınıflandırma algoritmalarının İngilizce veri kümeleri üzerindeki sonuçları incelendiğinde Doc2vec yerine TopWord2vec yöntemi kullanıldığında daha doğru ve geçerli sonuçlar çıktığı görülmüştür.

Tablo 4.17.İngilizce veri kümelerinde yöntem karşılaştırması

Veri Kümesi	Yöntem	Sınıflandırma Oranı (%)				
		SVM	KNN	Rastgele Orman	Bernoulli Naif Bayes	Lojistik Regresyon
Archeage	TopWord2vec	77.34	74.80	70.42	71.00	**77.90**
Archeage	Doc2vec	75.15	66.06	68.05	63.88	**75.44**
Iphone6	TopWord2vec	71.75	**73.08**	70.43	70.64	71.18
Iphone6	Doc2vec	**73.11**	71.82	66.16	61.28	70.66
Hobbit	TopWord2vec	77.12	75.05	76.22	72.04	**78.50**
Hobbit	Doc2vec	**73.73**	70.08	65.13	62.04	71.24
Sts-gold	TopWord2vec	79.12	73.76	73.11	73.41	**79.40**
Sts-gold	Doc2vec	**78.17**	69.32	69.47	69.47	77.29

Tez çalışması içerisinde geliştirilen yöntemin bir diğer avantajı ise bilgisayarda eğitim sürelerinin daha kısa olmasıdır. Deneylerde gözlenen model eğitim süreleri Tablo 4.16 üzerinde gösterilmektedir.

Tablo 4.18.Model eğitim süreleri

Eğitilen Model	Eğitim Süresi (dak.)		
	TopWord2vec	Doc2vec	Derlem Boyutu
Türkçe Model	**25**	367	10.073.869
İngilizce Model	**3**	14	1.682.876

Tablo 4.18 incelendiğinde Türkçe model eğitiminde 10.073.869 adet veri kullanılırken, İngilizce model eğitiminde 1.682.876 adet veri kullanılmıştır. Türkçe ve İngilizce doküman uzaylarında, TopWord2vec yöntemi Doc2vec yönteminden çok daha kısa bir sürede model eğitim işlemini gerçekleştirmektedir. Model eğitimlerinde 55 GB RAM ve 16 çekirdekli Linux işletim sistemi özelliklerine sahip makine kullanılmıştır.

5. SONUÇLAR VE ÖNERİLER

Günümüzde metin verileri her geçen gün artmaktadır. Özellikle yapılandırılmamış veriler (e-posta, web sayfası vb.) daha öncesinde görülmemiş boyutlarda olmaktadır. Metin verilerinin artması koşut olarak Doğal Dil İşleme alanında yapılan çalışmalarında hızla artmasına sebebiyet vermiştir. Bu hızla birlikte her geçen gün yeni yöntemler keşfedilmektedir. Keşfedilen yeni yöntemler geleneksel makine öğrenmesinin aksine yapay sinir ağlarını kullanan yöntemler olmaktadırlar. Yapay sinir ağlarını kullanan yöntemler özellik çıkarım işlemini otomatik yapabilmektedir. Bu sayede büyük veriler üzerinde geleneksel makine öğrenmesi algoritmalarından daha hızlı ve daha iyi performanslara sahip olabilmektedir.

Makine öğrenmesi algoritmalarında metinler kelime temsil yöntemleri ile temsil edilmektedirler. Bu temsil yöntemlerini geleneksel ve modern kelime temsil yöntemleri olarak ayırabiliriz. Geleneksel metin temsil yöntemleri, sabit uzunluklu vektörler ile gösterilmektedir. Bu yöntemler veri sayısının az olması durumunda iyi sonuçlar verebilirken veri sayısının çoğalması durumunda performans sorunlarına neden olabilmektedir. Modern kelime temsil yöntemleri ise sabit uzunluk yerine istenilen uzunlukta vektörler oluşturarak temsil etme görevini üstlenmektedir. Bu sayede verimize uygun vektör boyutu ile temsil etme işlemini gerçekleştirebilmekteyiz. Modern kelime temsil yöntemleri geleneksel temsil yöntemlerinin aksine veri sayısının çoğalması durumunda iyi sonuçlar çıkarmaktadır.

Modern kelime temsil yöntemlerinden en sık kullanılanı Mikolov tarafından tanıtılan Word2vec yöntemidir [2]. Word2vec yöntemi ile istenilen uzunlukta vektörler üreterek kelimelerimizi temsil edebiliriz. Word2vec yöntemimiz 2 farklı mimariye sahiptir. Bunlar CBOW ve Skip-Gram Mimarileridir [2]. Birbirinin ters işlevi olarak düşünülebilir. CBOW mimarisi ile bağlama bağlı olarak hedef kelime bulunurken, Skip-Gram mimarisi ile hedef kelime üzerinden bağlamlar çıkarılmaktadır. Dokümanlarımız içinde Doc2vec olarak adlandırılan kalıplama temelli bir temsil yöntemi bulunmaktadır. Doc2vec yöntemi Word2vec yönteminin dokümanlar üzerinde uygulanması gibi düşünülebilir. Temel farkı, kelimeler yerine dokümanların

kullanılması ve her paragraf numaralarının algoritmaya katılmasıdır. DM ve DBOW olmak üzere 2 farklı modeli bulunmaktadır. DM modeli CBOW modeline benzerken, DBOW modeli ise DM modelinin tersi olarak düşünülebilir.

Yapılan tez çalışması içerisinde Doc2vec üzerine yapılan çalışmalarda model eğitiminde kullanılan veri sayısının artmasıyla başarım oranları da artmaktadır. Fakat belirli bir artıştan sonra veri sayısının artması modelimizin performansında etkisi görülmemiştir. Doc2vec ile eğitilen 10 Milyon ve 20 milyon adet tweet verileri üzerinde sadece %1 oranında bir başarım farkı bulunmaktadır.

Verilen bir dokümanın konu ve kelime dağılımları ile bu dokümana ait kelime vektörleri dikkate alınarak yeni bir doküman vektör temsili geliştirilmiştir. LDA ve Word2vec yöntemleri, TopWord2vec adı verilen geliştirilen doküman vektörü sadece kelimeler arasındaki ilişkileri incelemekle kalmaz aynı zamanda dokümanın temasını ve bu temaların temsil eden kelimeleri de dikkate almaktadır.

Türkçe veri kümeleri üzerinde yöntem karşılaştırmaları yapıldığında Türkçe veri kümeleri üzerinde en yüksek başarımları sağlayan algoritmalar SVM ve Lojistik Regresyon algoritması olduğu görülmüştür. Türkçe veri kümeleri üzerinde sınıflandırma algoritmalarından bağımsız olarak yöntem karşılaştırması yapıldığında önermiş olduğumuz Topword2vec Modeli Doc2vec modelinden başarılı olduğu görülmüştür.

İngilizce veri kümeleri üzerinde yöntem karşılaştırmaları yapıldığında en yüksek başarımları sağlayan algoritmalar SVM ve Lojistik Regresyon algoritması olduğu görülmüştür. Sınıflandırma algoritmalarından bağımsız olarak yöntem karşılaştırması yapıldığında Iphone6 veri kümesi üzerinde Doc2vec algoritması TopWord2vec modelinden %1 oranından az olmak ile birlikte başarılı olmuştur. Diğer İngilizce veri kümeleri üzerinde de TopWord2vec Modeli Doc2vec modelinden başarılı olduğu görülmüştür.

Günümüzde veri sayısının çoğalması ile eğitilen modellerin süreleri model kullanımları açısından önem arz etmektedir. İyi bir model kurabilmek için veri sayısının çok olmasının yanında eğitim süresinin kısa olması içinde uygun yöntemlerin kullanılması önemli bir faktördür. Türkçe eğitim modellerinin İngilizce eğitim

modellerinden daha uzun sürdüğü görülmüştür. Uzun sürmesinin temel sebebi model eğitimlerinde Türkçe veriler üzerinde haber metinleri (uzun metinler) kullanıldığı için tekil kelime sayısının çok bulunmasıdır. İngilizce eğitim modelinde ise sadece tweet verileri (kısa metinler) kullanılıp daha az tekil kelime bulunduğu için ve derlem boyutu daha az olması sebeiyle daha kısa sürmüştür. Önermiş olduğumuz TopWord2vec modeli ile Doc2vec modeli, Türkçe ve İngilizce eğitim modelleri üzerinde eğitim süreleri açısından karşılaştırılmıştır. Karşılaştırma sonucunda TopWord2vec modeli ile Türkçe model eğitimi üzerinde yaklaşık olarak 15 kat daha hızlı bir iyileştirme gerçekleşirken, İngilizce model eğitimi üzerinde yaklaşık 5 kat daha hızlı bir model gerçekleştirilmiştir.

Yapılan deney sonuçları ile hem Türkçe hem İngilizce veri kümeleri üzerinde kurulan sınıflandırma algoritmaları ile doğruluk oranı ve model eğitim süreleri açısından önerilen yöntemin üstünlüğü gösterilmiştir.

Gelecekte bir plan olarak TopWord2vec yöntemi ile dokümanlardan bilgi alma (information retrieval) performansını iyileştirip iyileştirmediğini analiz etmeyi amaçlıyoruz. Ek olarak TopWord2vec yöntemini denetimsiz makine öğrenmesi algoritmaları üzerinde uygulamak hedeflenmektedir.

KAYNAKLAR

[1] Jiang B., Li Z., Chen H., Cohn A. G., Latent Topic Text Representation Learning on Statistical Manifolds, *IEEE Transactions on Neural Networks and Learning Systems*, 2018, **15**(3), 5643–5654.

[2] Mikolov T., Chen K., Corrado G., Dean J., Efficient Estimation of Word Representations in Vector Space, *International Conference on Learning Representations*, Arizona, USA, 2-4 May 2013.

[3] Le Q., Mikolov T., Distributed Representations of Sentences and Documents, *31st International Conference on Machine Learning*, Beijing, China, 21-26 June 2014.

[4] Kim D., Seo D., Cho S., Kang P., Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec, *Information. Sciences*, DOI: 10.1016/j.ins.2018.10.006

[5] Blei D. M., Ng A.Y., Jordan M. I., Latent Dirichlet Allocation, *Journal of Machine Learning Research*, 2003, **3,** 993-1022.

[6] Liu Y., Liu Z., Chua T.-S., Sun M., Topical Word Embeddings, *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, Texas, USA, 25-30 January 2015.

[7] Jiang D., Leung K. W.-T., Ng W., Li H., Beyond Click Graph: Topic Modeling for Search Engine Query Log Analysis, *Database Systems for Advanced Applications*, 2013, **7825,** 209–223.

[8] Hinton G. E., Mcclelland J. L., Rumelhart D. E., Editors: Mcclelland J. L., Rumelhart D. E., *Parallel distributed processing: explorations in the microstructure of cognition*, MIT Press, Massachusetts, 77-109, 1986.

[9] Bengio Y., Ducharme R., Vincent P., Jauvin C., A Neural Probabilistic Language Model, *Journal of Machine Learning Research*, 2003, **3,** 1137-1155.

[10] Mikolov T. et al., Recurrent neural network based language model, *11th Annual Conference of the International Speech Communication*, Chiba, Japan, 26-30 September 2010.

[11] Pennington J., Socher R., Manning C. D., GloVe: Global Vectors for Word Representation, *Empirical Methods in Natural Language Processing*, Doha, Qatar, 26-28 October, 2014.

[12] Wang Q., Liu R., Li H., Guo W., Topical Paragraph Vector learning, *11th International Conference on Natural Computation*, Zhangjiajie, China, 15-17 Aug. 2015.

[13] Ghosh S. et al., Contextual LSTM (CLSTM) models for Large scale NLP tasks, https://arxiv.org/pdf/1602.06291.pdf (Ziyaret tarihi: 10 Mayıs 2019).

[14] Mikolov T., Zweig G., Context dependent recurrent neural network language model, *IEEE Spoken Language Technology Workshop*, Florida, USA, 2-5 December 2012.

[15] Nguyen D. Q., Billingsley R., Du L., Johnson M., Improving Topic Models with Latent Feature Word Representations, *Transactions of the Association for Computational Linguistics*, 2015, **3**, 299-313.

[16] Wang Z., Ma L., Zhang Y., A Hybrid Document Feature Extraction Method Using Latent Dirichlet Allocation and Word2Vec, *IEEE First International Conference on Data Science in Cyberspace*, Changsha, China, 13-16 June 2016.

[17] Sohangir S., Wang D., Finding Expert Authors in Financial Forum Using Deep Learning Methods, *2018 Second IEEE International Conference on Robotic Computing*, California, USA, 31 January- 2 February 2018.

[18] Zhu W. et al., A study of damp-heat syndrome classification using Word2vec and TF-IDF, *2016 IEEE International Conference on Bioinformatics and Biomedicine*, Shenzhen, China, 15-18 December, 2016.

[19] Bilgin M., Şentürk İ. F., Sentiment analysis on Twitter data with semi-supervised Doc2Vec, *2rd International Conference on Computer Science and Engineering (UBMK)*, Antalya, Türkiye, 5-8 Ekim 2017.

[20] Şahin G., Turkish document classification based on Word2Vec and SVM classifier, *25th Signal Processing and Communications Applications Conference (SIU)*, Antalya, Türkiye, 15-18 Mayıs 2017.

[21] Ma H., Wang X., Hou J., Lu Y., Course recommendation based on semantic similarity analysis, *3rd IEEE International Conference on Control Science and Systems Engineering*, Beijing, China, 17-19 August 2017.

[22] Çelenli H. İ., Özturk T. S., Şahin G., Gerek A., Ganiz M. C., Document Embedding Based Supervised Methods for Turkish Text Classification, *3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo, Bosnia-Herzegovina, 20-23 September 2018.

[23] Çelenli H. İ., Application of paragraph vectors to news and tweet data, *26th Signal Processing and Communications Applications Conference (SIU)*, İzmir, Türkiye, 2-5 Mayıs 2018.

[24] Arora S., Liang Y., Ma T., A SIMPLE BUT TOUGH-TO-BEAT BASELINE FOR SEN-TENCE EMBEDDINGS, *5th International Conference on Learning Representations*, Toulon, France, 24-26 April 2017.

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

[26] Mei Q., Shen X., Zhai C., Automatic labeling of multinomial topic models, *13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, California, USA, 12-15 August 2007.

[27] Jadhav N., Topic models for Sentiment analysis: A Literature Survey, *Center for Indian Language Technology (CFILT)*, 2014.

[28] Perez-Cruzt F., Alarcon-Dianat P. L., Navia-V A., Artes-Rodriguez A., Fast Training of Support Vector Classifiers, *Advances in neural information processing systems*, British Columbia, Canada, 3-8 December 2001.

[29] https://en.wikipedia.org/wiki/Support-vector_machine, (Ziyaret tarihi: 10 Ağustos 2019).

[30] Aha D. W., Kibler D., Albert M. K., Instance-based learning algorithms, *Machine Learning*, 1991, **6**(1) 37–66.

[31] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm, (Ziyaret tarihi: 12 Ağustos 2019).

[32] https://medium.com/@k.ulgen90/lojistik-regresyon-makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-7-c6bc685a4084, (Ziyaret tarihi: 20 Ağustos 2019).

[33] Breiman L., "Random Forests. transparencias, *Statistics (Ber).*, 2001, **45**, 1-33.

[34] https://acadgild.com/blog/random-forest, (Ziyaret tarihi: 5 Eylül 2019).

[35] "Han J., Kamber M., *Data Mining Concepts and Techniques*, 2nd Edition, Morgan Kaufmann Publishers, Burlington, 2006.

[36] http://karlrosaen.com/ml/learning-log/2016-06-20/, (Ziyaret tarihi: 11 Kasım 2019).

[37] Kılınç D., Bozyiğit F., Yıldırım P., Yücalar F., Borandağ E., TTC-3600: A new benchmark dataset for Turkish text categorization, *Article Jornal of Information. Science.*, 2017, **43** 174–185.

[38] Saif H., Fernández M., He Y., Alani H., Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold, *Emotion and Sentiment in Social and Expressive Media (ESSEM*), 2013.

[39] Pedregosa F. et al., Scikit-learn: Machine Learning in Python, *Journal of machine learning research*, 2011, **12,** 2825–2830.

[40] Rehurek R., Sojka P., Software Framework for Topic Modelling with Large Corpora, *LREC 2010 Workshop on New Challenges for NLP*, Valetta, Malta, 22 May 2010.

KİŞİSEL YAYIN VE ESERLER

[1] **Çelenli H. İ.**, Özturk T. S., Şahin G., Gerek A., Ganiz M. C., Document Embedding Based Supervised Methods for Turkish Text Classification, *3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo, Bosnia-Herzegovina, 20-23 September 2018.

[2] **Çelenli H. İ.**, Application of paragraph vectors to news and tweet data, *26th Signal Processing and Communications Applications Conference (SIU)*, İzmir, Türkiye, 2-5 Mayıs 2018.

ÖZGEÇMİŞ

Halil İbrahim Çelenli 1994'de Van'da doğdu. Lise öğrenimini Tokat'da tamamladı. 2012 yılında girdiği Cumhuriyet Üniversitesi Bilgisayar Mühendisliği Bölümü'nden 2016 yılında mezun oldu. 2017 Yılında Kocaeli Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimine başladı. Yüksek lisans eğitiminde kelime kalıplama algoritmaları konusunda çalışmaları bulunmaktadır. Yüksek lisans eğitimine devam ederken IBSS Danışmanlık firmasında Ar-Ge Mühendisi olarak çalışmaktadır.