# TRAFİK YORUMLARININ SINIFLANDIRILMASINDA NORMALİZASYONUN ETKİSİ

**2019**
**DOKTORA TEZİ**
**BİLGİSAYAR MÜHENDİSLİĞİ**

**Zeynep ÖZER**

# TRAFİK YORUMLARININ SINIFLANDIRILMASINDA NORMALİZASYONUN ETKİSİ

**Zeynep ÖZER**

**Karabük Üniversitesi**
**Fen Bilimleri Enstitüsü**
**Bilgisayar Mühendisliği Anabilim Dalında**
**Doktora Tezi**
**Olarak Hazırlanmıştır**

**KARABÜK**
**Şubat 2019**

Zeynep ÖZER tarafından hazırlanan "TRAFİK YORUMLARININ SINIFLANDIRILMASINDA NORMALİZASYONUN ETKİSİ" başlıklı bu tezin Doktora Tezi olarak uygun olduğunu onaylarım.

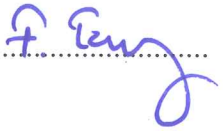
Doç. Dr. Oğuz FINDIK
Tez Danışmanı, Bilgisayar Bilimleri Anabilim Dalı ..... 

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Doktora tezi olarak kabul edilmiştir. 22/02/2019

| Ünvanı, Adı SOYADI (Kurumu) | İmzası |
|---|---|
| Başkan : Prof. Dr. Mehmet AKBABA (KBÜ) | ..... |
| Üye : Doç. Dr. Oğuz FINDIK (KBÜ) | ..... |
| Üye : Doç. Dr. Muharrem DÜĞENCİ (KBÜ) | ..... |
| Üye : Doç. Dr. Gülşen CEBİROĞLU ERYİĞİT (İTÜ) | ..... |
| Üye : Dr. Öğr. Üyesi Şafak KAYIKÇI (İBÜ) | ..... |

...../...../2019

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Doktora derecesini onamıştır.

Prof. Dr. Filiz ERSÖZ
Fen Bilimleri Enstitüsü Müdürü ..... 

*"Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."*

Zeynep ÖZER

# ÖZET

**Doktora Tezi**

## VERİ MADENCİLİĞİ ALGORİTMALARININ PARALEL VE DAĞITIK SİSTEMLERDE GERÇEKLEŞTİRİLMESİ

**Zeynep ÖZER**

**Karabük Üniversitesi**
**Fen Bilimleri Enstitüsü**
**Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı:**
**Doç. Dr. Oğuz FINDIK**
**Şubat 2019, 145 sayfa**

Trafik sıkışıklıkları, kazalar, hatalı sürücüler, araç arızaları ve yol çalışmaları gibi durumlar hem sürücüler için hem de trafik yönetim birimleri için zaman ve para kaybına neden olan ciddi problemler doğurmaktadır. Trafik olaylarının yer ve tipinin gerçek zamanlı olarak belirlenebilmesi sürücüler ve trafik yöneticilerine problemin çözülebilmesi veya alternatif güzergâh seçilebilmesi adına önemli kolaylıklar sağlayacaktır. Günümüzde trafikle ilgili olayların tespit edilmesi ve sıkışıklarının takibi için yaygın olarak kameralar ve fiziksel sensörlere dayanan sistemler kullanılmaktadır. Öte yandan sosyal medya platformlarında yapılan paylaşımlarda trafikle ilgili çok değerli bilgiler bulunmaktadır. Bu tez çalışmasında trafikle ilgili olayları sosyal medya mesajlarını (SMM) kullanarak makul bir doğruluk oranı içerisinde, maliyet etkin bir çözümle, geniş bir kapsama alanı içerisinde tespit edebilecek bir yöntem önerilmektedir. Bu doğrultuda yapılan çalışmalarda öncelikli olarak SMM'lerdeki gürültü probleminin azaltılması için kullanılan Türkçe metin

normalizasyon yaklaşımları belirsizlik durumları için kelime temsilleri ile genişletilmiştir. Ayrıca trafik alanına özel yaklaşık 1,5 M etiketsiz tweetten oluşan bir derlem hazırlanarak kelime temsilleri elde edilmiştir. Sonrasında ise etiketli veriden elde edilen kelime temsilleri ve etiketsiz veriden elde edilen alana özel kelime temsilleri kullanılarak tekrarlayan yapay sinir ağları, uzun kısa dönem bellek ağları, çift yönlü uzun kısa dönem bellek ağları, kapılı tekrarlamalı ünite ağları ve konvolüsyonel ağlar kullanılarak trafik veri setinin sınıflandırılması gerçekleştirilmiştir.

Özellikle Twitter son yıllarda kullanıcıların duyguları, düşünceleri ve olaylar hakkında bilgi paylaştıkları önemli bir platform haline geldi. Kullanıcıların gün içerisinde yaşadıkları olaylar hakkında anlık olarak bilgi paylaşımında bulunmaları bu platformdan sağlanan verileri olay tespiti açısından oldukça kıymetli bir hale getirdi. Bununla birlikte bu paylaşımlar mobil cihaz kullanımı, kullanıcı alışkanlıkları ve mesajlardaki karakter sayısı kısıtlamaları gibi sebeplerden dolayı yüksek miktarda gürültülü veri içermektedir. Bu tez çalışmasında öncelikli olarak SMM'lerdeki gürültünün azaltılabilmesi için Türkçe normalizasyon araçlarında kullanılan diyakritik, aksan ve ünlü harf restorasyon modülleri ile yazım denetimi modülü Word2vec tabanlı belirsizlik giderme modülü ile genişletilmiştir. Ayrıca normalizasyon işlemi için bütünüyle kaskad bir mimari yerine paralel ve kaskad yapıdan oluşan hibrit bir mimari kullanılmıştır. Sonuç olarak trafik veri seti üzerinde gerçekleştirilen normalizasyon işleminde güncel tekniklere kıyasla %25,95'lik bağıl hata azaltımına denk %10,41'lik bir iyileşme sağlanarak %70,29'luk bir başarım elde edilmiştir.

Trafik verilerinin sınıflandırılması işlemi hem iki sınıflı veri seti üzerinde hem de trafikli kaza, yol çalışması ve hava durumu gibi özel durumlarıda içeren 8 sınıflı veri seti üzerinde gerçekleştirildi. Ayrıca sınıflandırılma işlemindeki normalizasyonun etkinliğini değerlendirmek üzere temelde iki grup çalışma gerçekleştirildi. Bunlardan birincisinde kelime temsilleri (Word embedding) sadece sınıflandırma veri setindeki tweetler kullanılarak elde edilirken, ikinci grup çalışmada kelime temsillerinin elde edilmesi için yaklaşık 1,5 M tweetten oluşan alana özel bir derlem kullanıldı. Derlemin oluşturulurken trafik veri setinde kullanılan anahtar kelimelerin aynıları kullanıldı ve

ilave olarak doğrudan trafikle ilgili "@radyotrafik" ve "@radyotrafik06" gibi hesaplardan elde edilen tweetler kullanıldı. 2 sınıflı veri seti üzerinde, sınıflandırma işlemindeki etiketli veriden elde edilen kelime temsillerinin kullanıldığı durumda normalizasyon işlemi tüm modellerde sınıflandırma başarımını arttırmıştır. Etiketli kelime temsillerinde normalizasyon işlemi yaparak sınıflandırma yapmak en iyi durumda LSTM modeli ile %3'lük katkı sağlarken, alana özel kelime temsili kullanmak %8,9'luk çok daha iyi bir katkı sağlamıştır. 2 sınıflı veri setinde en yüksek başarım %96,15 ile alana özel kelime temsili kullanarak normalizasyon işlemi yapılmadan elde edilmiştir. Alana özel kelime temsili kullanımı çok sınıflı veri seti üzerinde tüm modeler için sınıflandırma başarımında en iyi durumda %32,08'lik bir iyileşme sağlamıştır. En iyi sınıflandırma başarımı alana özel kelime temsilllerine ilave olarak normalizasyon yapılmasıyla %89,92 GRU modeline aittir. Bununla normalizasyon yapılmadan elde edilen %88,5'lik LSTM modeline kıyasla normalizasyonun katkısı yalnızca %1,42'dir.

Genel olarak normalizasyon işlemi sınıflandırma performansını arttırmakla birlikte etkisi alana özel kelime temsili kullanılmasında kıyasla çok daha düşüktür. Bu durumda trafik veri seti için normalizasyon işlemi yapmadan alana özel kelime temsilleri ile oldukça yüksek başarımlar elde edilebilmektedir. Ayrıca hem 2 sınıflı hem çok sınıflı veri seti için etiketli veriden kelime temsili kullanılması durumunda CNN modeli diğer modellere kıyasla belirgin şekilde daha iyi sonuç vermektedir.

Bu durumda önerilen normalizasyon yaklaşımı en iyi durum için LSTM modeli ile %3'lük snıflandırma performansı artışı sağlanırken, en kötü durumda sınıflandırma performansındaki artış %0,25 ile CNN modelinde gerçekleşmiştir. Buna ilave olarak etiketli veriden elde edilen kelime temsillerinde en yüksek sınıflandırma başarım skorları CNN ile elde edilmiştir. Orjinal veri, önerilen normalizasyon yaklaşımı ve manuel normalizasyon için sırasıyla %93,05, %93,3 ve %93,35 olarak gerçekleşmiştir. Öte yandan alana özel kelime temsili kullanılmasıyla birlikte 2 sınıflı veri seti üzerinde en yüksek başarım oranı %96,15 ile orjinal veri kullanılarak LSTM modeli ile elde edilmiştir. Ayrıca bu koşul için normalizasyon işlemi LSTM modelinin sınıflandırma performansını arttırmamıştır.

**Anahtar Sözcükler :** Metin Madenciliği, twitter, tweet normalizasyonu.
**Bilim Kodu :** 924.1.014

## ABSTRACT

**Ph. D. Thesis**

**THE EFFECT OF NORMALIZATION ON THE CLASSIFICATION OF TRAFFIC COMMENTS**

**Zeynep ÖZER**

**Karabük University**
**Graduate Shool of Natural and Applied Sienes**
**Department of Computer Engineering**

**Thesis Advisor:**
**Assoc. Prof. Dr. Oğuz FINDIK**
**February 2019, 145 pages**

Situations such as traffic jams, accidents, faulty drivers, vehicle failures and roadworks lead to serious problems that cause loss of time and money for both drivers and traffic management units. The fact that the location and type of traffic events can be determined in real time will provide drivers and traffic managers with important facilities for solving the problem or choosing an alternative route. Nowadays, systems based on cameras and physical sensors are widely used for detecting traffic incidents and tracking congestion. On the other hand, there is very valuable information about traffic on social media platforms. In this thesis, we propose a method that can detect traffic related events within a wide coverage area with a cost effective solution at a reasonable rate of accuracy by using social media messages (SMM). In this respect, primarily the Turkish text normalization approaches, which are used for reducing noise problems in SMMs, are expanded with word embeddings for umbiguity situations. In addition, a collection of 1.5 M unlabeled tweets specific to the traffic area was prepared

and word representations were obtained. Then, word embeddings obtained from labeled data and domain spesific word embeddings, which are obtained from unlabeled data, are used to classification of traffic data set via recurrent neural networks, long short term term memory networks, bidirectional long short term term memory networks, gated recurrent unit networks and convolutional neural networks.

Especially in recent years, Twitter has become an important platform where users share information about emotions, thoughts and events. The fact that the users shared information about the events they experienced during the day made the data from this platform very valuable in terms of event detection. However, these shares contain a high amount of noisy data due to mobile device usage, user habits, and number of characters in messages. While users share important information about events, these shares have many noisy text problems, such as the diacritic character problem, typographical errors, laughing in random letters, and the use of acronyms and accents. In this thesis primarily, in order to reduce the noise in SMMs, the diacritic, accent and vowel letters restoration modules and spell-checking module used in Turkish normalization tools were extended with umbiguity module based on Word2vec. In addition, a hybrid architecture consisting of parallel and cascade structure is used instead of a completely cascade architecture for the normalization process. As a result, in the normalization process carried out on the traffic data set, an improvement of 10.41% corresponding to 25.95% relative error reduction was achieved compared to the state of the art techniques and a performance of 70.29% was achieved.

The classification process of the traffic data was carried out on both the two-class data set and the 8-class data set, which included special cases such as traffic accident, road work and weather. In addition, two groups of studies were conducted to assess the effectiveness of normalization in the classification process. In the first, word embedding was obtained using tweets only in the classification data set, while the second group study used a special collection of about 1,5 M tweet to obtain word embeddings. The same keywords used in the traffic data set were used to create the corpus, and also tweets obtained from directly traffic related accounts such as “@radyotrafik” and “@radotrafik06”. The normalization process increased the

classification performance on all models when word representations obtained from the labeled data in the classification process were used on the 2-class data set. In the case of normalization and labeled word embeddings, the LSTM model contributed 3% in the best case, while the use of domain-specific word embeddings provided a much better contribution of 8,9%. The highest performance in the 2-class data set was obtained without normalization by using a domain-specific word embeddings with 96,15%. The use of domain-specific word embeddings has provided a 32,08% improvement in the best case classification performance for all models on the multi-class data set. The best classification performance belongs to 89,92% GRU model by normalization in addition to domain-pecific word embeddings. With this, the contribution of normalization is only 1,42% compared to the 88.5% LSTM model obtained without normalization. However, compared to the 88.5% LSTM model without normalization, the contribution of normalization is only 1,42%. In this case, without the normalization process for the traffic data set, highly specific achievements can be obtained with domain-specific word embeddings. In addition, for both the 2-class and the multi-class data set, the CNN model is significantly better when the word embeddings is used from the labeled data.

In this case, the proposed normalization approach achieved the performance improvement of 3% with the LSTM model in the best case, whereas in the worst case, the performance in the classification performance was realized in the CNN model with 0,25%. In addition, the highest classification performance scores were obtained with CNN in word embeddings obtained from labeled data. For original data, proposed normalization approach, and manual normalization were accured 93,05%, 93,3% and 93,35%, respectively. On the other hand, with the use of domain-specific word embeddings, the highest performance rate was obtained with the LSTM model using the original data with 96,15% on the 2 class data set. Furthermore, the normalization process for this condition did not increase the classification performance of the LSTM model.

**Key Word** **:** Text Mining, twitter, tweet normalization.
**Science Code :** 924.1.014

## TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Doç. Dr. Oğuz FINDIK'a;

Tez izleme komitesinde yer alan ve değerli katkılarıyla çalışmama destek veren Sayın Prof. Dr. Mehmet AKBABA ve Sayın Doç. Dr. Muharrem DÜĞENCİ'ye;

Bu çalışma süresince paylaştığı görüşleri doğrultusunda çalışmanın her zaman daha iyiye yönlenmesindeki katkıları ve sabrı için hayat arkadaşım Dr. İlyas ÖZER'e;

Çalışmam için gerekli olanakları sağlayan kayınvalidem Hacer ÖZER ve kayınpederim Arslan ÖZER'e;

Her zaman başarılı bir araştırmacı olmam için beni teşvik eden babam Prof. Dr. Abdullah BAYRAM'a;

Maddi manevi hiçbir yardımı esirgemeden yanımda olan ve ümidimi kaybettiğim her anda beni cesaretlendiren, kendimi şanslı hissettiren ilk öğretmenim canım annem Adalet BAYRAM'a ve sevgili kardeşlerime;

Varlığı ile moral ve neşe kaynağım olan sevgili oğlum Muhammed Emir ÖZER'e teşekkürlerimi sunarım.

## İÇİNDEKİLER

**Sayfa**

**Sayfa**

**Sayfa**

**Sayfa**

## ŞEKİLLER DİZİNİ

**Sayfa**

**Sayfa**

## ÇİZELGELER DİZİNİ

**Sayfa**

**Sayfa**

## SİMGELER VE KISALTMALAR DİZİNİ

### KISALTMALAR

ASCII : American Standard Code for Information Interchange

BiLSTM: Bi-directional Long-Short Term Memory (Çift Yönlü Uzun Kısa Dönem Hafıza)

CBOW : Continuous Bag of Words

CNN : Convolution Neural Network (Konvolüsyonel Sinir Ağları)

ÇDTT : Çok Dilli Tweet Tespiti

DD : Dil Doğrulayıcı

DDİ : Doğal Dil İşleme

DN : Doğru Negatif

DP : Doğru Pozitif

DR : Diyakritik Restorasyon

GRU : Gated Recurrent Unit (Kapılı Tekrarlamalı Ünite)

HMM : Hidden Markov Model

LSTM : Long-Short Term Memory (Uzun Kısa Dönem Hafiza)

MRL : Morphologically Rich Language (Morfolojik Açıdan Zengin Dil)

NLP : Natural Language Processing (Doğal Dil İşleme)

OOV : Out of Vocabulary (Sözlük Dışı İfadeler)

ReLU : Rectified Linear Units

RHG : Rastgele Harflerle Gülme

RNN : Recurrent Neural Networks (Tekrarlayan Yapay Sinir Ağı)

SMM : Sosyal Medya Mesajı

SMS : Short Message Service (Kısa Mesaj Hizmeti)

TF-IDF : Term Frequency - Inverse Document Frequency (Terim Sıklığı - Ters Döküman Sıklığı)

URL : Uniform Resource Locator (Standart Kaynak Bulucu)

ÜHR : Ünlü Harf Restorasyonu

YP : Yanlış Pozitif

YN : Yanlış Negatif

# BÖLÜM 1

# GİRİŞ

Trafik sıkışıklıkları, trafik kazaları, kötü hava koşulları, araç arızaları ve yol çalışmaları gibi durumlar hem sürücüler için hem de trafik yönetim birimleri için ciddi problemler doğurmaktadır [1,2]. Trafik olaylarının yer ve tipinin gerçek zamanlı olarak belirlenmesi sürücüler ve trafik yöneticilerinin önlem alması ve alternatif çözümler üretmesi açısından son derece önemlidir.

İnsanlar on yıllar boyunca gerçek zamanlı olarak trafik olaylarının türünü yerini ve konumunu belirleyecek sistemler üzerine çaba sarf etmişlerdir [2]. Geliştirilen sistemlerde trafik olaylarının tespit edilmesi için yaygın olarak kullanılan yöntemler, görüntüleme sensörleri, akustik algılayıcılar, manyetik sensörler ve pasif kızılötesi sensörler gibi fiziksel algılayıcılar kullanarak trafik yoğunluğu, akış ve hız gibi parametrelerin ölçülmesine ve değerlendirilmesine dayanmaktadır. Bu sensörler genellikle sabit bir noktaya yerleştirilerek ölçümler yapılmaktadır. Veri madenciliği uygulamaları yoluyla dağınık sensörlerden gerçek zamanlı olarak elde edilen veriler büyük bir zamansal ve mekânsal spektrumda değerlendirilerek olay tespiti yapılmaya çalışılmaktadır. Bu sistemler otoyollar ve ana arterlerde genellikle başarılı bir şekilde çalışmaktadır ancak yerel arterlerdeki olayların tespiti açısından başarımları düşüktür [2]. Buna ilave olarak kullanılan sistemler genellikle yüksek yatırım maliyetleri gerektirmektedir. Ayrıca bu sistemlerin kapsama alanlarının genellikle kısıtlı olması olay tespitinde gecikmelere neden olabilmektedir [3]. Özellikle yerel arterlerdeki tıkanıkların sebepleri hatalı park, yaya trafiği ve belediye çalışmaları gibi çok çeşitli olabilir. Bu durumda sensörlerin mevcut kapsama alanları ele alındığında tespit edilmelerini güçleştirmektedir.

Bu fiziksel sensörlerin kapsama alanlarının düşük olmasının başlıca sebebi kurulum, bakım, onarım ve işletme maliyetlerinin oldukça yüksek olması nedeniyle sadece ana

arterler üzerine seyrek olarak yerleştirilmelerinden kaynaklanmaktadır. Bir kavşak noktasına yerleştirilen endüktif döngü dedektörünün kurulum ve bakım maliyeti yıllık olarak 9 500 ile 16 700 Amerikan Doları düzeyindedir [4-5]. Bu nedenle fiziksel sensörlere dayanan yöntemler yüksek maliyet dezavantajları, sınırlı uzaysal kapsama alanları gibi faktörlere bağlı olarak her zaman uygun çözüm olmayabilirler. Bununla birlikte son yıllarda çevrimiçi topluluklar tarafından üretilen çok miktarda veri geniş uzaysal kapsama alanı ve düşük maliyetleri ile trafik olaylarının tespit edilmesi ve değerlendirilmesi açısından alternatif ve mevcut fiziksel sensörleri destekleyen bir çözüm sağlayabilir [6].

Bu tez çalışmasında trafikle ilgili olayları makul bir doğruluk oranı içerisinde, maliyet etkin bir çözümle, geniş bir kapsama alanı içerisinde tespit edebilecek bir yöntem önerilmektedir. Önerilen yöntem sosyal medya mesajlarından elde edilen metin verilerinin sınıflandırılmasına ve trafikle ilgili olanların tespit edilerek araç arızası, kaza veya yol çalışması gibi olayın alt türünün belirlenmesine dayanmaktadır.

Sosyal medya kullanımının geçtiğimiz birkaç yıl içerisinde tüm dünya çapında çok ciddi bir büyüme kaydetmesi ve kullanıcıların bu platformlar üzerinden her alanda bilgi paylaşımında bulunması araştırmacıların ciddi anlamda ilgisini çekmiştir. Buna bağlı olarak bu mecralardan elde edilen veriler gerçek zamanlı olay tespiti [7], duygu tespiti [8], suç tahmini [9] ve trafik anomali tespiti [10] gibi oldukça geniş bir alanda kullanılmaya başlandı. Öte yandan sosyal ağlarda kullanılan yazım dili, mesajlardaki karakter sayısı kısıtlamaları, mobil cihaz kullanımı ve bu cihazlardaki klavye yerleşimlerinin birçok dil açısından uygun olmayışı gibi sebeplerden dolayı formal yazım diline kıyasla çok büyük farklılıklar göstermektedir. Kullanıcılar tarafından oluşturulan verinin çok yüksek miktarda gürültü barındırmasına bağlı olarak bu verilerin kullanılmadan önce normalize edilmesine yönelik çalışmalar son dönemde hız kazandı [11-12-13]. Bununla birlikte Türkçe gibi zengin morfolojik yapıya sahip diller için normalizasyon işlemi oldukça zorlu bir görevdir. Ayrıca Türkçe'nin sahip olduğu zengin ek ve kök yapısı hata formlarını oldukça genişletmekte birbirine yakın yüzey formlarında pek çok aday kelimenin oluşmasına neden olmaktadır.

Son dönemde trafik olaylarının gerçek zamanlı olarak tespit edilebilmesi üzerine de çalışmalar yapılmaktadır. Özellikle Twitter gibi platformlar üzerinde insanlar fikirleri, duygusal durumları ve yaşadıkları olaylar gibi pek çok durum ve konu hakkında paylaşımda bulunmaktalar. Trafikle ilgili yaşadıkları olaylarda bu paylaşım konularından bir tanesidir. Bu tez çalışmasında önerilen yaklaşımda Twitter kullanıcıların paylaşımları anahtar sözcük temeline dayanan bir yaklaşım ile sorgulanmakta ve elde edilen tweetlerin trafikle ilgili olup olmadığı ve ilgiliyse alt sınıfının ne olduğu belirlenmektedir. Bu noktada yaşanılan başlıca problem sosyal medya paylaşımlarındaki içeriklerin normal yazım diline kıyasla oldukça fazla yazım yanlışı ve sözlük dışı kelime içermesidir. Bu durum sınıflandırma performansını olumsuz etkilemekte ve başarımı ciddi anlamda düşürmektedir. Yazım yanlışlarının kullanıcı tarafından el yordamıyla düzeltilmesi de gerçek zamanlı çalışması beklenen sistemler için kabul edilebilir bir çözüm değildir. Bununla birlikte sosyal medya mesajlarını otomatik olarak normalize edecek ve doğru kelime formlarını belirleyecek normalizasyon yaklaşımları İngilizce gibi kısıtlı sayıda kelimeye sahip diller için yaygın olarak kullanılan yöntemlerden bir tanesidir. Fakat Türkçe için normalizasyon görevi İngilizce'ye kıyasla çok daha zordur. Birçok hatalı yazım formu için birden fazla aday kelime türetilebilmektedir. Bu doğrultuda öncelikli olarak Türkçe normalizasyon yaklaşımlarını belirsizlik durumları için kelime temsilleri ile genişleten bir yaklaşım sunulmaktadır. Bunun yanı sıra birçok çalışmada alana özel kelime temsillerinin kullanılması, kullanıcılar tarafından üretilen gürültülü metinlerin sınıflandırılması işleminde oldukça başarılı sonuçlar sağlamaktadır. Bu tez çalışmasında da Twitter verilerinden, trafik alanıyla ilgili etiketsiz veriler kullanılarak alana özel bir kelime temsili hazırlanmakta ve sınıflandırma başarımı üzerine olan etkisi değerlendirilmektedir. Sınıflandırma başarımı üzerine olan etkiyi kıyaslamak için kelime temsilleri sınıflandırma veri setindeki etiketli veriden elde edilerek ve ayrıca alana özel etiketsiz verilerden elde edilerek ayrı ayrı sınıflandırma işlemi gerçekleştirilmekte ve sonuçlar kıyaslanmaktadır. Bunun yanı sıra her iki kelime temsil yöntemi üzerinde orijinal veri, önerilen normalizasyon yaklaşımı kullanılarak elde edilmiş veri ve son olarak insan denetçiler tarafından el yordamıyla normalize edilmiş veriler kullanılarak sınıflandırma işlemi gerçekleştirilmekte normalizasyon işleminin sağladığı katkı kıyaslanmaktadır. Böylelikle Türkçe gibi normalizasyon işleminin oldukça zorlu olduğu diller için metin normalizasyonu yapmadan sadece

alana özel kelime temsili kullanımıyla yüksek sınıfılandırma başarımı elde edilip edilemeyeceği değerlendirilmektedir. Ayrıca trafik verilerinin sınıflandırılmasında RNN, LSTM, BiLSTM, GRU ve CNN gibi güncel makine öğrenmesi yöntemleri kullanılarak sınıflandırma işlemi gerçekleştirilmektedir. Buna ilave olarak bu makine öğrenmesi yöntemlerinin oluşturduğu derin ve sığ mimarilerin sınıflandırma başarımı üzerine olan etkisi değerlendirilmektedir.

## 1.1. LİTERATÜR TARAMASI

Trafik sıkışıklıkları hem sürücüler için hem de trafik sistemini idare eden yöneticiler için çok ciddi bir problemdir. Trafik sıkışıklıklarını, tekrarlayan sıkışıklıklar ve tekrarlamayan sıkışıklıklar olarak iki temel grup altında kategorize etmek mümkündür [2]. Tekrarlayan sıkışıklıkların genel kaynağı insanların günlük kullanım alışkanlıklarından kaynaklanan güzergâh seçimi gibi faktörlere dayanmaktadır ve gün be gün tekrar etmektedir. Öte yandan tekrarlamayan trafik sıkışıklıklarının temel nedeni trafik kazası, kötü hava koşulları, yol yapım ve bakım onarım çalışmaları, araç arızaları gibi özel nedenlere dayanmaktadır ve bu özel nedenler toplam trafik sıkışıklığının neredeyse yarısını oluşturmaktadır [2,14]. Dolayısıyla bu özel olayların zamanında, doğru ve maliyet etkin bir şekilde tespit edilebilmesi trafik yönetimi açısından oldukça önemli bir konudur. Bu konuya efektif bir çözüm getirilmesi özellikle büyük metropollerdeki trafik sıkışıklıklarının azaltılması ve buna bağlı olarak oluşan kayıpların azaltılması bakımından oldukça faydalı olacaktır.

Son yıllarda akıllı telefon ve mobil cihaz kullanımında yaşanan çok büyük orandaki artış, trafikle ilgili olaylara ait bilgileri toplayabilmek adına umut verici alternatif bir yaklaşım haline gelmektedir [15]. Akıllı telefonlar ile sıklıkla paylaşımda bulunulan platformlardan bir tanesi de Twitter'dır.

Twitter REST API ve Streaming API uygulamaları aracılığıyla kullanıcılarına kamuya açık tweetleri ücretsiz olarak elde etme olanağı sunmaktadır. REST API uygulaması kullanılarak son 7 gün içerisinde paylaşılmış olan mesajlar anahtar kelimeler kullanılarak temin edilebilirken, Streaming API uygulamasında gerçek zamanlı olarak paylaşılan tweetlerin elde edilebilmesine olanak sağlamaktadır. Twitter API'leri

kullanılarak alınan bilgilerin içerisinde konum, zaman, kullanıcı bilgileri ve 140 karakterle sınırlı (Kasım 2017'den sonrası için 280 karakter) metin verisi yer almaktadır. Twitter platformu üzerinden temin edilen veriler ile seçim sonuçlarını [16] ve suç tahmin etme [9], felaketlerin tespit edilmesi [17,18] ve sosyal olayların tanımlanması [19] ve buna benzer pek çok çalışma yapılmaktadır.

Trafik yönetimini kolaylaştırabilmek adına trafikle ilgili olayları gerçek zamanlı olarak tespit edebilmesi üzerine çalışmalar da yapılmaktadır [20]. Twitter mesajları kullanılarak gerçekleştirilen bu çalışmada 2 farklı veri seti hazırlanarak tweetlerden trafik olaylarının tespit edilebilmesi hedeflenmiştir. İlk veri setinde problem ikili sınıflandırma problem olarak ele alınmış ve tweetler trafikle ilgili olanlar ve trafikle ilgili olmayanlar olmak üzere iki grup altında toplanmıştır. Birinci grup veri seti toplamda 1330 tweetten oluşmaktadır. Bu tweetlerin 665 trafikle ilgili ve 665 tanesi de trafikle ilgili olmayan tweetler olmak üzere dengeli bir paylaşım gerçekleştirilmiştir. Hazırladıkları ikinci grup veri seti ise 3 farklı sınıftan meydana gelmektedir. Bu sınıflar sırasıyla harici olay (futbol maçı, konser vs.), trafik sıkışıklığı veya kaza ve son olarak trafikle ilgili olmayan tweetlerdir. Benzer şekilde bu veri setinde her biri 333 adet tweetten oluşan toplam 999 adetlik dengeli bir veri setidir. Öte yandan SMM'ler kısaltmalar, yazım yanlışları, dil bilgisel hatalar, sözlük dışı kelimeler içeren ve doğası gereği son derece kısa olan yapılandırılmamış ve düzensiz metinlerdir [20,21]. Bu nedenle SMM'lerden olay tespit edilmesi blog ve e-postalar gibi iyi biçimlendirilmiş geleneksel medya araçlarından olay tespit edilmesine kıyasla çok daha zorlu bir görevdir [20,22]. Bu sebeplerden dolayı gelen bilgileri analiz edebilmek için çalışmalarında veri madenciliği, makine öğrenmesi, istatistik ve doğal işleme alanlarından çıkarılan yöntemleri kullanan metin madenciliği tekniklerini kullanmışlardır [20]. Önerdikleri sistem üç ayrı modülden oluşmaktadır. Bu modüller sırasıyla SMM'leri alma ve ön işleme, SMM'lerin detaylandırılması, sınıflandırma modülleridir. Twitter API'si aracılığı ile alınan verilere ön işlem aşamasında düzenli ifade (Regular Expression) filtresi uygulanarak kullanıcı adı, zaman bilgisi ve hashtag gibi bilgiler temizlenmiştir. Yine aynı aşamada tweet içerisindeki tüm karakterler küçük harf ile ifade edilmiştir. Detaylandırma aşamasında ise öncelikli olarak tokenization işlemi gerçekleştirilmiştir. Sonrasında ise gereksiz kelimelerin (stop-word) temizlenmesi işlemi yapılmıştır. Bu aşamadan sonra geriye kalan kelimelerde

sırasıyla kök bulma ve kök filtreleme işlemleri yapılmıştır. Sonrasında elde edilen özelliklerin sunumu gerçekleştirilmiştir. Son aşamada ise destek vektör makineleri kullanılarak tweetlerin sınıflandırılması işlemi gerçekleştirilmiştir. Neticede ikili sınıflandırma probleminden oluşan ilk veri setinde %95,75'lik bir sınıflandırma başarımı elde edilmiştir. Üç farklı sınıftan oluşan veri setinin sınıflandırılması aşamasında ise başarı oranı yaklaşık %7'lik bir düşüş kaydederek %88,89 olarak gerçekleşmiştir.

Günümüzde sosyal medya platformları sıradan kullanıcıların yanı sıra kamu kurum ve kuruluşları, büyük medya şirketleri ve devlet büyükleri gibi pek çok kullanıcıya da ev sahipliği yapmaktadır. Bu tip kullanıcıların paylaşımları sıradan kullanıcılar ile kıyaslandığında daha az gürültülü veri içermektedir. Yani yazım ve dilbilgisi kurallarına göreceli olarak daha fazla riayet edilmektedir. Bu tip kaynaklar tarafından oluşturulan verilerin daha güvenilir olması ve çok az miktarda gürültü barındırıyor olması nedeniyle doğrudan SMM'lerdeki bu tip kullanıcı mesajlarını dikkate alan çalışmalar da gerçekleştirilmektedir [23]. Yapılan çalışmada sadece kamu kuruluşlarının ve haber sitelerinin Twitter hesapları üzerinden yaptıkları paylaşımlar dikkate alınarak trafikle ilgili tweetlerin yorumlanması için bir metodoloji önermişlerdir. Çalışmalarında trafikle ilgili olan olaylar altı farklı sınıf altında tanımlanmıştır. Bu sınıflar kısıtlama, kaza, araç arızası, trafik durumu, hava durumu ve trafikle ilgili sınıflandırılmamış diğer olaylardır. Gerçekleştirdikleri testlerde %73'lük bir başarım elde etmişlerdir. Motivasyonlarının temel kaynağı kamyon filolarını izlemek için bir araç geliştirmek ve trafikten elde edilen veriler doğrultusunda daha etkin bir filo yönetimi sağlamak, maliyetleri azaltmak ve ürünlerin zamanında müşterilere ulaştırılmasını sağlamaktır. Bu doğrultuda 500 kamyondan oluşan orta ölçekli sıvı gaz dağıtım şirketi üzerinde ve 5 000 kamyondan oluşan büyük ölçekli yakıt dağıtım şirketleri üzerinde testler gerçekleştirmişler ve önerdikleri modelin yakıt dağıtımındaki süreleri azaltarak ürünlerin müşteriye zamanında ulaşmasının sağlandığını ve müşteri memnuniyetinde artış yaşandığını kaydetmişlerdir. Kullanılan yaklaşımdaki en büyük problem sadece kurumsal Twitter hesaplarını dikkate almalarıdır. Bu nedenle gerçek kullanıcı paylaşımlarından kaynaklı veriler kullanılmamıştır. Her ne kadar sıradan kullanıcı verileri çok yüksek oranda gürültü barındırıyor olsa da gerçek zamanlı olay tespiti açısından son derece kritik bir

öneme sahiptir. Çünkü kurumsal hesaplara ulaşacak olay zamanları ile olayın gerçekte yaşandığı zaman arasında ciddi farklılıklar olabilmektedir. Bu durumda sistemin gerçek amacından uzaklaşmasına neden olabilir. Sistem kurumsal hesaplar üzerinden olayı başarılı bir şekilde tespit edebilirken olay çoktan meydana gelmiş ve bitmiş olabilir. Bu nedenden dolayı tüm kullanıcı verilerinin temin edilmesi uzaysal ve zamansal temsil kabiliyeti açısından oldukça önemli bir konudur.

Tüm trafik olaylarını tespit etmek yerine doğrudan trafik kazası veya araç arızası gibi daha özel alt bir konuyu tespit etmek de zaman zaman ön plana çıkan görevler arasında yer almaktadır. Özellikle trafik kazaları beklenmeyen bir anda trafiğin ciddi anlamda sıkışmasına neden olabilmektedir. Bu doğrultuda sosyal medya verilerinden sadece trafik kazalarına ilişkin olayların tespit edilmesine yönelik çalışmalar da yürütülmüştür [24]. Gerçekleştirdikleri çalışmada araç kazaları üç sınıf altında toplanmıştır. Bu gruplar çarpışma, hareketsiz kalmış araç ve araç yangınıdır. Çalışmalarında öncelikli olarak gerçek haber metinlerinde kaza kelimesi ile birlikte yaygın olarak kullanılan kelimelerin tespit edilmesi için bir çalışma gerçekleştirmişlerdir. Buradan polis, yol, şerit ve ölü gibi 100 farklı kelimeden oluşan bir anahtar sözcük dizisi oluşturmuşlardır. Anahtar sözcükler çıkarılırken dilsel olarak yaygın bir biçimde kullanılan kelimeler, olayın özelliklerini belirten ve coğrafi konumu hakkında bilgi veren ifadeler dahil edilmemiştir. İkinci adım olarak da bu anahtar kelimeler ile ilişkili tweetlerin tespit edilmesi olmuştur. Bu doğrultuda tweetler anahtar sözcüklere göre filtrelenmiş sonrasında filtrelen tweetler rastgele olarak seçilerek trafik kazası ile ilgili olup olmadığı el yordamıyla etiketlenmiştir. Bu etiketlemeler doğrultusunda kazayla ilgili tweetlerdeki en sık kullanılan kelimelerin neler olduğu tespit edilmiştir. Sonuç olarak kazayla ilişkili olan 900 adet tweetten oluşan bir veri seti elde etmişlerdir. 900 adet tweetten oluşan veri kümesinin tamamı yaklaşık olarak 20 bin adet tokendan oluşmaktadır. Bu tokenların bir kısmı seçilip kökleri bulunduktan sonra özellik olarak kullanılmıştır. Bu bağlamda belirgin bir dilsel anlama gelmeyen ve önemli bir olayı işaret etmeyen kelimeler çıkarılmıştır. Sonuç olarak 44 bireysel token ve 17 token çiftinden oluşan bir özellik seti ile %85'lik bir başarım elde edilmiştir.

Karayollarındaki trafik sıkışıklığı büyük şehirlerin oldukça önemli problemlerinden birisidir. Bu doğrultuda trafik sıkışıklığı düzeyinde ne boyutta olduğunun

belirlenmesinde güzergâh seçimi açısından önemlidir. Trafik sıkışıklığının değerlendirilmesi için yukarıdaki bölümlerde de belirtildiği gibi pek çok sensör kullanılabilmektedir. Bununla birlikte bir diğer yaklaşım da Twitter verileri üzerinden elde edilen veriler sıkışıklık durumunun değerlendirilip değerlendirilemeyeceğidir [25]. Trafik sıkışıklığının Twitter mesajları ile değerlendirilmesi için yapılan çalışmada [25], farklı türlerde Twitter verileri ve C4.5 karar ağaçları kullanılarak trafik sıkışıklık durumunun ne düzeyde olduğunu tespit etmek için bir model önermişlerdir. Önerilen yaklaşım ile bir cadde üzerinde gelecek 30 dakikalık süre içerisinde trafik sıkışıklık düzeyinin ne boyutta olacağını tahmin etmeye çalışmışlardır. Eğitim verilerinde kullandıkları özelliklerde sırasıyla haftanın günü, günün saatleri, dakika ve tweet yoğunluğudur. Tweet yoğunluğunu tespit edebilmek için önceden belirledikleri Twitter hesaplarını ve konum bilgisiyle etiketlenmiş tweetleri baz almışladır. Yol isimleri ve trafik durumu ile ilişkili 1 821 adet tweet toplayarak bu tweetleri zaman dilimlerine göre ayırmışlardır. Sonuç olarak trafik sıkışıklık durumunu 3 farklı sıkışıklık düzeyinde sınıflandırmışlar ve ortalama F1 skorunda 0,892'lik bir değer elde etmişlerdir.

SMM'lerden anlamlı bilgilerin tespit edilmesi büyük çoğunlukla DDİ (Doğal Dil İşleme) ve metin madenciliği konuları ile ilişkilidir. DDİ ve metin madenciliğinde kullanılan yöntemler SMM'lere uygulanarak bunlardan bilgi çıkarımı yapılmaktadır. DDİ'de yaygın olarak kullanılan yöntemlerden bir tanesi de metin özetlemedir. Bu yöntem tweetlerden trafik ile ilgili olayların tespit edilmesinde de kullanılmıştır [26]. Önerilen yöntem ilk aşamada trafikle ilgili anahtar kelimelerin türetilmesi süreciyle başlamaktadır. Sonrasında trafikle ilgili tweetleri tespit edebilmek için birliktelik kuralı tabanlı yinelemeli sorgu genişletme algoritması kullanılmaktadır. Sonrasında elde edilen tweetler metin özetlemesi yaklaşımı ile özetlenmiştir. Çalışma ile gereksiz tweet içeriklerini ayırarak anlaşılabilir özet bir tweet metni çıkarılabileceğini ortaya koymuşlardır.

Yukarıdaki bölümlerde de belirtildiği gibi sosyal medya platformlarının kullanımının son yıllarda giderek artması ve özellikle Twitter gibi platformlarda paylaşılan mesajların araştırmacılar tarafından kullanılabiliyor olması bu alanda yapılan çalışmalara ciddi anlamda ivme kazandırdı. Bu çalışmalar genel olarak DDİ ve metin

madenciliği yöntemlerine dayanmaktadır. DDİ ve metin madenciliği konuları hâlihazırda birçok zorlu görevle baş etmek zorundadır. Bu zorlukların başında gelen konulardan bir tanesi özellikle kelime çantası (bag of words) gibi yöntemlerde elde edilen özellik vektörünün son derece seyrek verilerden oluşmasıdır. Yani derlemde çok büyük sayıda kelimeler bulunurken sadece bir cümle vektör olarak temsil edilmek istendiğinde çok sayıda sıfırdan oluşan bir veri seti elde edilmektedir. Bu durum sınıflandırma işlemini halihazırda ciddi bir şekilde zorlaştırmaktadır. Bunun üzerine SMM'lerdeki yazım hataları da dahil edildiğinde seyreklik problem giderek artmaktadır. Örnek 1'deki tweetlere bakılacak olursa sadece 4 adet tweet için bile birçok hatalı yazım formunun olduğu görülmektedir. Buna ilave olarak aynı kelime için farklı yazımlar da mevcuttur. Örneğin ilk tweette "köprü" kelimesi ASCII formda "kopru" şeklinde yazılmışken üçüncü tweette doğru bir şekilde yazılmıştır. Benzer şekilde "sağ" ve "şeritte" kelimeleri son tweette doğru bir biçimde yazılmışken ikinci tweette ASCII formunda yazılmıştır. Mevcut tweetler üzerinde herhangi bir normalizasyon işlemi yapılmadan kelime çantası elde edilmek istenirse özellik vektörünün boyutu ciddi bir şekilde artacaktır. Ayrıca normalde aynı kelimeyi temsil etmelerine karşılık farklı kelimeler gibi değerlendirileceklerdir.

Örnek 1:

- @radyotrafik e5'te sefakoyde kopru yonunde radisson otele 50 m kala orta seritte 2 araclik kaza oldu simdi, cekici gerekmez, hafif hasarli.
- @radyotrafik küçükcekmece benzinlik sonrası kamyonla ufak arac e5 sag seritte bekliyo kaza olabilir avcılar yonunde bgniz olsun
- İstanbul'da yağmur yagmasın trafik felç 2 adımlık yol bile 1 saatte gidilmiyo ?? yok yol yaptık yol köprü yaptık aynı sorunlar devam
- @radyotrafik06 konya yolu g.başı yönünde trn gnş köprüsü altında sağ şeritte araç arızası! araç sahibi reflektör kutusu sallayarak uyarıyor

Kelime çantaları metin verilerinin sınıflandırılmasında özellik vektörü elde edilmesi için sıklıkla kullanılan yöntemlerden bir tanesidir. Fakat bu yöntemde kelimelerin konum bilgileri kaybolmaktadır. TF-IDF ve n-gram modelleride benzer şekilde yaygın olarak kullanılan yöntemlerden bazılarıdır. Bu yaklaşımlarda her bir veri seti çok sayıda 0'dan oluşan seyrek matrislerle temsil edilmektedirler. Bu durumda

sınıflandırma başarımlarını olumsuz etkilemektedir. Bununla birlikte son yıllarda makine öğrenmesi algoritmalarında yaşanan gelişmelerle birlikte yeni yöntemlerde kullanılmaya başlanmıştır. Son dönemde yaygın olarak kullanılmaya başlanan yöntemlerden bir tanesi kelime temsillerini kullanarak sınıflandırma işlemini gerçekleştirmektir. Bu yaklaşımda her bir metin kelime dizileri şeklinde ifade edilmektedir. Ayrıca her bir kelime de sayısal bir vektörel gösterime sahiptir. Bu yaklaşımda anlamsal olarak birbirine yakın kelimeler benzer vektörler üretmektedir. Ayrıca kelimelerin konum bilgileri korunmakta ve düşük boyutlu vektörel temsiller sayesinde veri seyrekliği probleminin de önüne geçilmektedir.

Kelime temsilleri pek çok sınıflandırma görevinde kullanılmaya başlanmıştır. Bu alanlardan bir tanesi de kullanıcı yorumların sınıflandırılmasıdır [27]. Bu doğrultuda restoran ve film yorumlarını sınıflandırmak için skip-gram modeli ile eğittikleri 200 boyutlu kelime temsillerini kullanarak duygu sınıflandırması gerçekleştirmişlerdir. Sınıflandırma işleminde 3 farklı restoran yorumları veri seti ve 1 tane de film yorumları veri seti kullanmışlardır ve kelime temsilleri her bir veri seti için ayrı ayrı eğitilmiştir. Sonuç olarak kullandıkları LSTM modelinde CNN modeline göre daha yüksek başarım elde etmişlerdir. Twitter mesajlarından duygu analizi üzerine yapılan bir çalışmada [28] benzer şekilde Word2vec kullanılarak kelime temsilleri elde edilmiştir. Ön eğitim aşamasında parametreler bütünüyle rastgele başlatılarak, Word2vec aracı ile büyük boyutlu etiketsiz derlem üzerinden eğitilerek ve son olarak önerdikleri modelde de bir önceki aşamada eğittikleri kelime temsillerini etiketli veri üzerinde tekrardan ayarlayarak kullanmışlardır. Önerdikleri modelde birçok yaklaşıma kıyasla oldukça yüksek skorlar elde etmişlerdir. Kelime temsillerinin elde edilmesi için kullanılan kaynaklar farklılık gösterebilmektedir. Doğrudan alana özel sınıflandırma verisinden kelime temsilleri elde edilebildiği gibi [27], büyük boyutlu etiketsiz derlemler kullanılarak da kelime temsilleri elde edilebilmektedir [28]. Her iki yöntemin bir birleşimi olarak etiketsiz verilerden elde edilen kelime temsilleri etiketli veri ile tekrar eğitilerek daha güçlü temsiller sağlanabilmektedir [28]. Bunun yanı sıra kamuya açık genel kelime temsilleri kullanılarak da sınıflandırma işlemi gerçekleştirilebilmektedir [29]. İlaç reaksiyonları üzerine yapılan bir diğer çalışmada da [30] jenerik tweetlerin yanı sıra ilaçla ilgili alana özel tweetler bir arada kullanılarak kelime temsilleri elde edilmiştir. Bir diğer çalışmada da [31] benzer şekilde jenerik ve

alana özel kelime temsillerinin bir arada kullanılmasına yönelik bir yaklaşım sunulmuştur. Jenerik kelime temsilleri pek çok çalışmada kullanılmakla birçok alan kendine özgü kelimeler bulundurmaktadır. Dolayısıyla zengin alan bilgisi gerektiren durumlarda doğrudan alana özel kelime temsilleri kullanmak jenerik temsillere göre daha iyi sonuçlar sağlayabilmektedir. Bu doğrultuda biyomedikal alanında metin madenciliği üzerine sunulan çalışmada [32] alana özel kelime temsillerinin genel temsillerine kıyasla daha iyi sonuçlar sağladığı görülmüştür. Benzer şekilde yine biyomedikal alanında yapılan çalışmada alana özel kelime temsillerinin yeterli miktarda veriden elde edilmesi durumunda sınıflandırma performansını yukarı taşıdığı görülmüştür [33]. Bununla birlikte yeterli miktarda alana özel kelime temsilinin elde edilemediği durumlarda genel kelime temsillerinin de alana özel temsillere kıyasla bir miktar daha düşük olmakla birlikte yeterince başarılı skorlar sağladığı değerlendirilmiştir [33].

Özellikle alana özel kelime temsilleri pek çok çalışmada kullanıcılar tarafından üretilen gürültülü metinler üzerinde oldukça başarılı sonuçlar vermektedir. Twitter platformundaki mesajlarda kriz durumlarıyla ilişkili verileri sınıflandırmak için yapılan çalışmada [34] kriz durumlarıyla ilgili etiketsiz veriler kullanılarak başarılı sonuçlar elde etmişlerdir. Sosyal medya platformlarından paylaşılan metinlerin sınıflandırılmasının yanı sıra insan makine etkileşimi uygulamaları, akıllı kişisel asistanlar birçok uygulamada da alana özel kelime temsilleri olumlu katkılar sağlamaktadır. Bu doğrultuda anlamsal ifade sınıflandırmak için yapılan çalışmada [35] Word2vec kullanılarak elde edilen alana özel kelime temsilleri kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. RNN ve CNN modellerinin kullanıldığı çalışmada sığ mimariler ile daha iyi sonuçlar elde edilmiştir.

Büyük etiketsiz verilerden elde edilen kelime temsilleri pek çok uygulamada oldukça başarılı sonuçlar sağlamaktadır. Buna ilave olarak özellikle belirli alanlar için hazırlanmış kelime temsillerinin de genel temsillere kıyasla çok daha iyi sonuçlar sağladığı görülmektedir. Alana özel kelime temsillerinin kullanımının yanı sıra metin normalizasyon işlemlerinin de birçok çalışmada sınıflandırma başarımını yukarı taşıdığı bilinmektedir.

Çizelge 1.1. “aramak” fiilinin Twitter’daki bazı yanlış kullanımları.

| Hatalı Yazım | Doğru Yazım |
|---|---|
| arıyo | arıyor |
| ariyo | arıyor |
| ariyor | arıyor |
| arıyooo | arıyor |
| ariym | arayım |
| arıyom | arıyorum |
| ariyorum | arıyorum |
| arıyon | arıyorsun |
| ariyosun | arıyorsun |
| arıyolar | arıyorlar |
| arıyozz | arıyoruz |
| ariyouz | arıyoruz |
| ariyonuz | arıyorsunuz |
| ariyoken | arıyorken |
| arıyodum | arıyordum |
| arıcam | arayacağım |
| arıcak | arayacak |
| arıyımda | arayayım da |
| arıyomuşum | arıyormuşum |
| arıyomuşsunuz | arıyormuşsunuz |
| arıyomuşcasına | arıyormuşçasına |

Sosyal medya mesajları (SMM’ler) yaygın olarak kısaltmalar, harf tekrarları, yeni sözcük türevleri, emojiler gibi hataları içerirler, bu hatalar genellikle istatistiksel yöntemler kullanılarak giderilerek ilgili metnin formal hali elde edilir. Ancak bu durum Türkçe, Fince ve Korece gibi MRL’ler ve eklemeli diller için kendine has birtakım zorlukları barındırmaktadır. Bu tip dillerde farklı köklere yeni ekler getirilerek oldukça fazla sayıda yeni kelime ve anlam elde etmek mümkündür. MRL’lerde ve eklemeli olan dillerde her bir isim kökünden binlerce yüzey formu türetilebilirken her bir fiil kökünden ise milyonlarca yüzey formu üretilebilmektedir [36]. Buna bağlı olarak sosyal medya mesajları içerindeki hatalı yüzey formları da ciddi anlamda çeşitlilik göstermektedir. Çizelge 1.1’de “aramak” fiilinin Twitter’daki bazı yanlış kullanımları görülmektedir. Ayrıca yüzey formlarındaki bu çeşitlilik sözlük boyutunun ciddi anlamda arttırdığı için doğru yüzey formunu sözlükte arama temeline

dayalı metotlar çok yüksek bellek gereksinimleri ve zaman kısıtları nedeniyle verimli olmamaktadır. Bu sebeplerden dolayı normalizasyon işlemi alana özel kelime temsili kullanımına kıyasla çok daha zorlu bir görevdir. Ayrıca sık karşılaşılan yazım yanlışları alan bazında farklılık gösterebilmektedir. Genel anlamda sık karşılaşılmayan bir ifadeyle özel bir alanda sıklıkla karşılabilmekte veya aynı kısaltmalar farklı anlamlar taşıyabilmektedir. Bu durumda bir normalizasyon aracının başka bir alan için hatalı sonuçlar türetmesine neden olabilmektedir.

Normalizasyon işleminin genel zorluklarının yanı sıra Türkçe gibi zengin morfolojik yapısı ve sahip olduğu diyakritik karakterler gibi birçok problem mevcuttur. Diyakritik karakterlerin ASCII (American Standard Code for Information Interchange) eşleniklerinin oldukça yaygın olarak kullanılmasıdır. Diyakritik, harflere eklenen ve onların sesini değiştiren birtakım işaretlerdir ve Türkçe, Fransızca, Yunanca, Macarca ve İspanyolca gibi pek çok dilde kullanılmaktadırlar. Türk alfabesi (ç,ı,İ,ğ,ö,ş,ü) harflerinden oluşan 7 adet diyakritik karaktere sahiptir ve bu karakterlerin ASCII eşlenikleri Çizelge 1.2'de de görüldüğü gibi sırasıyla (c,i,I,g,o,s,u) harfleridir. Deasciification veya Diacritization olarak da adlandırılan Diyakritik Restorasyon kısmen veya tamamen ASCII formda yazılmış kelimelerin tekrardan doğru bir biçimde yazılmasıdır. Buradaki temel problem diyakritik karakterlerin ASCII eşleniklerinin de Türk alfabesinde kullanılan geçerli harfler olmasına bağlı olarak diyakritik restorasyon sonrasında birden fazla geçerli Türkçe kelime oluşmasıdır. Örneğin "göl" kelimesinin ASCII formu "gol" kelimesidir ve her iki kelime de Türkçe'de yaygın olarak kullanılan geçerli sözcüklerdir. Bu durum diyakritik restorasyon işleminde belirsizliklerin oluşmasına neden olmaktadır. Bunlara ilave olarak ayırt edici herhangi bir işaret bırakılmadan bitişik yazılan kelimeler, çok dilli metinlerin kullanılması, ünlü harflerin yazılmaması, rastgele harflerle gülme, aksanlı yazılan ifadelerle de Türkçe SMM'lerde karşılaşılmaktadır.

Çizelge 1.2. Türk alfabesindeki diyakritik karakterler.

| | | | | | | |
|---|---|---|---|---|---|---|
| Türkçe | ç | ğ | ı | ö | ş | ü |
| | Ç | Ğ | İ | Ö | Ş | Ü |
| ASCII | c | g | i | o | s | u |
| | C | G | I | O | S | U |

Bugüne kadar yapılan normalizasyon çalışmalarının büyük kısmı İngilizce gibi kısıtlı sayıda kelimeye sahip dilleri hedef aldığı için birçok çalışmada yaygın olarak istatistiksel yöntemler veya sözlükte arama metotları kullanılmıştır. Ayrıca, ilk çalışmalarda genellikle kısa mesaj metinlerinin normalize edilmesi hedeflenmiştir [37,38]. Bu doğrultuda İngilizce kısa mesajları normalizasyonu için farklı yaklaşımlar denenmiştir. Bu yaklaşımlardan bir tanesinde istatistiksel makine çevirisine dayanan [37] bir yöntem kullanılarak mesaj içerikleri formal yazım diline çevrilmiştir. Bu yaklaşımın en önemli problemi sadece eğitim setinde karşılaştığı problemler için çözüm üretebilmesidir. Bu problemin üstesinden gelmek için benzer şekilde kısa mesaj içerikleri üzerinde yapılan normalizasyon işlemi için denetimsiz gürültülü kanal metodu önerilmiştir [38]. Öte yandan kısa mesaj içerisindeki kasıtlı yazım hatalarının varlığı standart yazım hatası düzeltme tekniklerinin başarımını ciddi bir biçimde azaltmaktadır [39]. Kasıtlı yazım hatalarından kaynaklanan normalizasyon problemini gidermek için basit bir bigram dil modelini ve Hidden Markov Model (HMM) tabanlı kelime hata modelini kullanan bir yaklaşım önerilmiştir [39]. Bigram dil modelini kullandıkları modelde kelime seviyesinde hata oranlarının %35 oranında azaltılması sağlanmıştır. Sosyal medya platformlarının yaygınlaşmasıyla, metin normalleştirme alanındaki çalışmalar SMM'lerin normalleşmesine odaklanmış ve bu alanda birçok yöntem önerilmiştir [40,41]. Bu yöntemlerden bir tanesinde mesajlar gürültü azaltma ön işlemi ve makine çevirisi modelinden oluşan iki aşamalı bir yaklaşım kullanılarak normalize edilmiş [40] ve BLEU skorlaması kullanılarak sistem performansı kıyaslanmıştır. Hatalı kelimelerin düzeltilmesi kadar önemli olan bir diğer konu da hatanın türünün tespit edilmesidir [41]. Bu doğrultuda yapılan çalışmada öncelikle hatalı formlarda yazılan kelimeler bir sınıflandırıcı kullanılarak tanımlanmıştır [41] ve daha sonra aday kelime, sözcük benzerliği ve içeriğe uygun olarak sözlüğe bakma yöntemi ile önerilir [41]. İngilizce'nin Türkçe gibi dillere kıyasla az sayıda kelimeye sahip olması nedeni ile sözlüğe bakma temeline dayanan kural tabanlı yaklaşımlar birçok çalışmada kullanılmıştır [41-42]. Bunun yanı sıra kural tabanlı yaklaşımların aksine SMM'lerin normalizasyonu için denetimsiz istatiksel model de önerilmiştir [43]. Önerilen modelde standart olan ve olmayan tokenlar arası ilişki log-lineer bir model ile temsil edilmiştir. Bu çalışmaya benzer şekilde bir diğer çalışmada da SMS'lerin kendilerine özgü kısaltmalar ve stenografi barındırmaları ve bu tip yazımların genellikle statik sözlük tabanlı metin benzerlik teknikleri veya kural tabanlı

normalizasyon teknikleri ile düzeltilemeyeceği fikrinden hareketle bağlamsal (içeriklerin) benzerlik kavramını yakalamak için sözcüklerin dağınık temsillerini öğrenen denetimsiz (unsupervised) bir model önermişlerdir [44]. Sinir ağlarından ve log-lineer modellerden elde edilen lineer ve non-lineer dağıtık gösterimleri kullanarak yaptıkları testleri Twitter veri seti üzerinde Microsoft Word ve Aspell ile kıyaslamışlar ve daha iyi sonuçlar elde etmişlerdir. [45]'deki çalışmada ise denetimsiz modellerin aksine ilk aşamada Koşullu Rastgele Alanlar (Conditonal Random Field) temeline dayanan denetimli (supervised) bir yaklaşım ve ikinci aşamasında ise aday kelime formlarına bir dizi sezgisel (heuristics) kuralların uygulandığı iki aşamalı hibrit bir model önerilmiştir. Bu çalışmalardan farklı olarak [46]'daki çalışmada geleneksel gürültülü kanal modeli kelime-kelime geçişlerini elde etmek için hece seviyesinde genişletilmiştir. İngilizce SMM'lerin normalleştirilmesi üzerine yapılan çalışmaların büyük çoğunluğu en uygun aday kelimenin düzeltme sözlüğünde sunulan karşılığının olduğunu varsaymaktadır. Fakat kelimenin kullanıldığı zaman aralığı ve metnin içeriği gibi faktörlere bağlı olarak herhangi bir gürültülü formda yazılmış kelimeye karşılık gelen birden fazla doğru formda kelime bulanabilir. Bu fikirden hareketle [38]'deki çalışmada gürültülü kelimelerin üretildiği zaman dilimini, bulunduğu metnin içeriğini ve gürültülü kelimenin ilgi alanını baz alarak en uygun kelimeyi öneren bir model sunulmuştur. Bu çalışmanın aksine [48]'deki çalışma SMM'lerin normalleştirilmesinin aslında bir eşleştirme problemi olduğunu varsayarak Adaptive Similarity Function ve En Yakın Komşu eşleşmesinin (matching) normalleştirme probleminin çözümü için en doğrudan prosedür olduğunu önermişlerdir. Önerdikleri benzerlik fonksiyonu Bağlamsal, Fonetik Benzerlik Ve String olmak üzere 3 farklı parametrenin ağırlıklandırılmış değerlerini içerirken, en yakın komşu eşleşme algoritması minimum bir benzerlik eşik değerine sahiptir. Bir diğer çalışmada [49] ise Levenshtein Uzaklığı, Sözlük Haritaları ve Demetaphone algoritmasına dayanan modüler bir yaklaşım Lexnorm 1.2 isimli İngilizce tweetlerden oluşan bir veri seti üzerinde test edilerek elde edilen sonuçlar mevcut denetimsiz yaklaşımlar ile kıyaslanmış ve diğer yöntemlerden daha iyi sonuç elde edilmiştir.

İngilizce üzerine yapılan çalışmalara kıyasla daha az olmakla birlikte Flemenkçe [50], Malayca [11], İspanyolca [51], Vietnemca [52] ve Çince [53] diğer diller için de birtakım SMM'lerin normalizasyon çalışmaları yapılmıştır. Öte yandan Türkçe,

Macarca, Fince, Japonca ve Korece gibi MRL'ler için SMM normalizasyonu oldukça zorlu bir görevdir ve yazarın bilgisine göre Türkçe üzerine yapılan sadece 1 adet komple SMM normalizasyon çalışması [12] bulunmaktadır. Bu çalışmada [41]'deki çalışmaya benzer şekilde normalizasyon problemi hatalı kelimenin tespiti ve aday kelime üretilmesi olmak üzere 2 aşamada ele alınmıştır. Önerdikleri modelde hatalı kelimeler bir morfolojik analizör kullanılarak tespit edilmiştir. Aday kelime üretme sürecinde ise Harf Durumu Dönüştürme, Değiştirme Kuralları ve Sözlükte Arama, Özel İsim Tespiti, Diyakritik Restorasyon, Ünlü Harf Restorasyonu, Aksan Normalizasyonu ve Yazım Hatası Düzeltme olmak üzere 7 katmanlı kaskad bir mimari kullanılmıştır. Bu kaskad mimari yaklaşımı ile tek bir aşamada çözülemeyen kompleks problemlerin çözümü hedeflenmiştir. Önerdikleri modelin performansını Türkçe NLP çerçevesi [54] MS Word gibi araçlarla kıyaslamışlar ve bunlara göre belirgin bir performans artışı sağlamışlardır. Öte yandan önerdikleri modelin en büyük dezavantajlarından bir tanesi çok dilli tweetler, Sözcük Ayırma ve Rastgele Harflerle Gülme gibi sosyal medya yazımlarında sıkılıkla karşılaşılan birtakım problemlerin dikkate alınmamasıdır. Ayrıca bütünüyle kaskad bir mimari kullanılması birden fazla modülün aynı anda sonuç üretebileceği durumlar için her zaman daha önceki katmandaki sonucun geçerli kelime olarak alınmasıdır. Bunlara ilave olarak Türk alfabesinde bulunmayan "w" ve "x" gibi harflerin doğrudan değiştirilmesi bu kelimelerin gerçekten yabancı kelimeler olabileceği ihtimalini bütünüyle yok saymaktadır. Dahası Türkçe'nin zengin morfolojik yapısı birbirine yakın yüzey formlarında anlamsal olarak birbirinden çok uzak kelimelerin oluşmasına imkân tanımaktadır. Örneğin Çizelge 1.3'de de görüldüğü gibi "anadın" şeklinde hatalı bir biçimde yazılmış yüzey formu için bir düzenleme mesafesi (edit distance) uzaklığında "anaydın", "anardın", "kanadın" gibi 23 farklı aday kelime mevcuttur. Bu durumda en uygun kelimenin seçilebilmesi için tüm cümlenin anlamsal olarak değerlendirilmesi oldukça önemli bir gerekliliktir. Fakat önerilen modelde bu tip problemler için tam anlamıyla bir çözüm sunulmamaktadır.

Çizelge 1.3. "anadın" şeklinde yanlış yazılmış kelime için bir düzenleme mesafesi uzaklığındaki bazı aday kelimeler.

| Aday Kelimeler |
|---|
| inadın |
| onadın |
| adadın |
| aradın |
| atadın |
| ananın |
| kanadın |
| anladın |
| anardın |
| anaydın |

Ayrıca SMM normalizasyon problemine komple bir çözüm sunmamakla birlikte Türkçe sosyal medya metinlerinde sıklıkla karşılaşılan diyakritik restorasyon ve Ünlü Harf Restorasyonu (ÜHR) gibi problemleri ele alan bir dizi çalışma da yapılmıştır. [55]'deki çalışmada Türkçe DR problemi ele alınmış ve 18 milyon kelimelik bir derlem kullanılarak karakter tabanlı bir Hidden Markov Model inşa edilmiştir. Değişik n-gram dil modelleri kullanarak yapılan testlerde 4-gram modeliyle 3-gram modeline göre belirgin bir hata azaltımı sağlanmıştır. Türkçe üzerine yapılan çalışmalarda yaygın olarak kullanılan NLP çerçevesi Zemberek [54] ise diğer birçok modelin aksine her bir token için birden fazla aday kelime üreterek, bu kelimeleri köklerinin derlemde geçme sıklığına göre sıralamaktadır. Emacs Turkish Mode [56] çalışmasında ise [55]'deki çalışmadan esinlenilerek 1 milyon kelimelik bir derlem ile bir çeşit Karar Listesi (Decision List) algoritması olan Greedy Prepend Algoritmasını kullanarak Karar Listesi oluşturulmuştur. [57] ve [58]'deki çalışmalar ise DR performansının Bilgi Getirimi (İnformation Retrieval) uygulamaları üzerindeki etkilerini ele almışlardır. [13]'deki çalışmada Koşullu Rastgele Alanlar (Conditonal Random Field) ve Dil Doğrulayıcı (Language Validator) tabanlı iki katmanlı bir yaklaşım ile Ünlü Harf Restorasyonu ve Diyakritik Restorasyon problemi üzerine bir çalışma yapılmıştır. Bunlara ilave olarak DR ve ÜHR problemi ile Türkçe'nin yanı sıra Arapça [59-61], Hırvatça [62], Vietnamca [63], Romence [64] gibi pek çok dilde karşılaşılmaktadır.

## 1.2. ÇALIŞMANIN TEMEL AMACI VE LİTERATÜRE KATKILARI

Son yıllarda SMM'ler DDİ ve metin madenciliği uygulamalarında, metinden duygu çıkarımı, olay tespiti ve suç tahmini, sosyal olay analizi gibi pek çok alanda kullanılmaya başlanmıştır. Trafik olaylarının tespit edilmeye çalışılması da bu kullanım alalanlarından biridir. Bununla birlikte yukarıdaki bölümlerde de belirtildiği gibi SMM'ler yüksek miktarda gürültü barındırmaktadır. Bu gürültü de nihai sistem performansını son derece olumsuz etkilemektedir.

Metinlerdeki gürültüyü azaltmak için kullanılan yöntemlerden bir tanesi normalizasyon yapmaktır. Fakat Türkçe için normalizasyon işlemi oldukça zorlu bir görevdir. Ayrıca normalizasyon işleminin pek çok adımında hatalı bir kelime için birden fazla aday kelime üretilebilmektedir. Bu doğrultuda ilk olarak, Türkçe için yapılan çalışmalarda normalizasyon işlemi için kullanılan diyakritik, aksan ve ünlü harf resterasyon modülleri ile yazım denetimi modülü Word2vec tabanlı belirsizlik giderme modülü ile genişletilmiştir. Buna ilave olarak normalizasyon işlemi için bütünüyle kaskad bir mimari yerine kaskad ve paralel mimariden olauşan hibrit bir yapı kullanılarak normalizasyon işleminin başarımının arttırılması hedeflenmiştir.

Bununla birlikte normalizasyon işlemi oldukça zorlu bir görevdir ve aynı model farklı veri setleri üzerinde birbirinden farklı skorlar sağlayabilmektedir. Bunun yanı sıra yaygın olarak karşılaşılan hata türleri alandan alana farklılık gösterebilmektedir. Bu durumlarda özellikle Türkçe gibi diller için normalizasyon işleminin etkinliğini oldukça kısıtlamaktadır.

Öte yandan son yıllarda alana özel kelime temsillerinin kullanımı kullanıcılar tarafından oluşturulan gürültülü metinlerin sınıflandırılmasında oldukça başarılı sonuçlar sağlamaktadır. Bu doğrultuda trafikle ilgili Twitter hesaplarından ve anahtar kelimeye dayalı bir sorgulama işlemi gerçekleştirilerek yaklaşık 1,5 M adet etiketsiz tweetten oluşan trafik alanına özel bir kelime temsili oluşturulmuştur.

Trafik verilerinin sınıflandırılmasında doğrudan etiketli sınıflandırma verisinden elde edilen kelime temsilleri ve etiketsiz verilerden elde edilen trafik alanına özel kelime

temsilleri ayrı ayrı kullanılarak sınıflandırma başarımları üzerinde olan etkileri değerlendirilmiştir. Bunun yanı sıra sınıflandırma işlemi orijinal trafik verisi, önerilen normalizasyon yaklaşımı ile normalize edilmiş trafik verisi ve son olarak el yordamıyla normalize edilmiş trafik verisi kullanılarak her iki kelime temsili üzerinde testler gerçekleştirildi. Böylelikle hem alana özel kelime temsili kullanımının sağlamış olduğu katkı hemde normalizasyon işleminin sağlamış olduğu katkı değerlendirildi.

Bunlara ilave olarak LSTM, GRU ve CNN gibi güncel makine öğrenmesi yöntemlerinin başarımları kıyaslandı. Ayrıca bu yöntemlerin oluşturuduğu sığ ve derin mimarilerin sınıflandırma performansı üzerine olan etkisi değerlendirildi. Bunların yanı sıra alana özel kelime temsilleri ile genel amaçlı jenerik kelime temsillerinin trafik verilerinin sınıflandırmasına olan etkisi ortaya kondu.

Nihai olarak Türkçe metin normalizasyon yaklaşımı kelime temsilleri ile genişletildi ve ayrıca alana özel kelime temsilleri ile normalizasyon işleminin sınıflandırma performansı ortaya konarak Türkçe metin normalizasyonun zorlukları ve hataya meyilli olması nedeniyle normalizasyon işlemi yapmadan başarılı sınıflandırma performansının elde edilemeyeceği üzerine bir değerlendirme sunuldu.

# BÖLÜM 2

# TÜRKÇE

Türkçe dünyanın en yaygın olarak konuşulan ilk 7 dili arasında yer alan [57,58], Ural-Altay dil ailesinin Altay koluna mensup bir dildir. Zengin morfolojik içeriği ve eklemeli yapısı sayesinde bir kök veya gövdeden birçok yeni kelime ve anlam türetmek mümkündür. Türkçe'de bir isim kökünden binlerce kelime türetilebilir [54]. Öte yandan bir fiil kökünden türetilebilecek kelime sayısı ise bunun çok çok ötesindedir ve milyonlarca kelime türetilebilmektedir [54]. Ek A'da "anlamak" fiilinin farklı zaman ve kişi ekleri gibi durumlar altında bazı örnek çekimleri görülmektedir. Bu durum İngilizce gibi kısıtlı sayıda kelimeye sahip diller ile kıyaslandığında sözlük boyutunun aşırı şekilde büyümesine neden olmaktadır ve pek çok durumda kelime yazımlarının doğruluğunu bir sözlükten kontrol etme temeline dayanan uygulamalar Türkçe gibi morfolojik açıdan zengin diller için uygun olmamaktadır. Bu durumda kullanılabilecek alternatif bir yöntem morfolojik çözümleme yapmaktır. Bu sayede kelimenin geçerli sözcük olup olmadığı kontrol edilebilmektedir.

Bu bölüme morfoloji ve morfolojik çözümleme kavramları tanıtılarak Türkçe'nin morfolojik yapısı incelenecektir.

## 2.1. MORFOLOJİ

Morfoloji, dilbiliminde kelimelerin iç yapısını ve bu yapıyı oluşturan kuralları inceleyen bir bilimdir. Morfemlerin kurallı bir şekilde bir araya gelmesi ile dilin anlamlı en küçük parçası olan sözcükler meydana gelir [65]. Örnek 2'de görüldüğü gibi "göz" köküne belirli bir kural dahilinde yeni ekler getirilerek "gözcü" ve "gözcülük" kelimeleri türetilmiştir. Aynı kural doğrultusunda ve yine aynı ekler kullanılarak "söz" kökünden "sözcü" ve "sözcülük" kelimeleri türetilebilmiştir. Benzer şekilde yine "göz" kökünden "gözlük", "gözlükçü" ve "gözlükçülük" kelimelerinin türetilebildiği görülmektedir ve "göz" kelimesinin yerine aynı gruptan

farklı bir kök kullanılarak aynı kural doğrultusunda farklı kelimelerin türetilmesi mümkündür. Morfoloji temelde dildeki bu kuralları anlamaya ve ortaya çıkarmaya çalışan bir bilimdir.

Morfemler, sırasıyla serbest ve bağımlı morfem olmak üzere iki farklı gruba ayrılmaktadırlar. Serbest morfemler tek başlarına sözcük olarak kullanılabilirlerken, bağımlı morfemler tek başlarına kullanılamamaktadırlar. Sözcük kökleri genellikle serbest morfemlerdir. Öte yandan ekler ise bağımlı morfemlerdir ve sözcüklere eklenmek suretiyle kullanılabilmektedirler [65].

Örnek 2:

- Göz +cü + lük : gözcülük
- At +çı +lık : atçılık
- Söz + cü +lük : sözcülük
- Göz +lük + çü +lük : gözlükçülük
- Kulak + lık +çı +lık : kulaklıkçılık

Türkçe'de kullanılan kelimeler en az bir kökten meydana gelmektedirler. Bununla birlikte teorik olarak sınırsız sayıda ek alabilmektedirler. Türkçe sondan eklemeli bir olmakla birlikte az da olsa genellikle yabancı dillerden giren bazı ön ekler de bulunmaktadır [65-66]. Örnek 3'te Türkçe'de kullanılan bazı önekler ile türetilmiş kelimeler görülmektedir.

Örnek 3:

- Namert
- Namüsait
- Biçare
- Asosyal

Morfoloji, çekimsel, türetimsel ve bileşik morfoloji olmak üzere genellikle üç ana grup altında ele alınmaktadır [67]. Çekimsel morfolojide kelimenin türünde bir değişiklik yaşanmamaktadır. Sadece zaman ve şahıs ekleri gibi ekler gelmektedir. Öte yandan

türetimsel morfolojide ise yapım ekleri ile yeni bir kelime meydana getirilmektedir. Son olarak bileşik morfolojide iki farklı kelimenin birleşmesi ile yeni bir kelime oluşmaktadır. Örnek 4'te her üç morfoloji türü için de birer örnek görülmektedir.

Örnek 4:

- Ağaçlarımız – Çekimsel Morfoloji
- Yolcu – Türetimsel Morfoloji
- Ayakkabı – Bileşik Morfoloji

## 2.2. MORFOLOJİK ANALİZ

Morfolojik analiz işleminde esasında yapılan işlem bir kelimeyi meydana getiren kök ve eklerin neler olduğunu ortaya çıkarmaktır. Analiz edilen sözcük geçerli bir Türkçe sözcük ise analiz işleminin sonucunda bir çıktı üretilebilmektedir. Aksi durumda önceden belirlenmiş morfolojik kurallara uyan bir kök ve ek dizilimin olmadığı anlamına gelir. Dolayısıyla analizi yapılan kelime geçerli bir Türkçe kelime değildir. Örnek 5'te "geliyorlardı" ve "okutturmak" kelimelerinin morfolojik analizleri görülmektedir.

Örnek 5:

- Gel / kök_fiil
- + (i)yor / zaman_şimdiki
- + lar / fiil_şahıs_Onlar
- + dı / fiil_hikaye

- Oku / kök_fiil
- + t / fiil_oldurgan
- + tur / fiil_ettirgen
- + mak / fiil_mastar

Yukarıda da belirtildiği gibi DDİ uygulamalarında Türkçe gibi eklemeli ve zengin morfolojiye sahip diller için sözlük tabanlı yaklaşımlar kullanmak sözlük boyutunun

aşırı şekilde büyümesine bağlı olarak çok uygun değildir. Bununla birlikte morfolojik analiz kullanmanın olumsuz yanlarından bir tanesi de günlük hayatta kullanılmayan bazı kelime formlarının da morfolojik açıdan doğru olabilmesidir. Örneğin "bardaklıkçılık" gibi bir kelime günlük hayatta kullanılmamakla birlikte morfolojik açıdan geçerli bir dizilime sahiptir.

## 2.3. MORFOLOJİK BELİRSİZLİK

Morfolojik analiz işleminin bir diğer problemi de morfolojik belirsizliktir. Türkçe'nin sahip olduğu geniş ek kümesi ve zengin morfolojik yapısı nedeniyle morfolojik analiz işleminden sonra farklı anlamlara gelebilen olası kök ve ek dizilimlerinin oluşması mümkündür. Bu durum morfolojik bir belirsizliğe neden olmaktadır. Örnek 6'da "yollar" kelimesi için bazı olası morfolojik çözümler görülmektedir.

Örnek 6:

- Yol / kök_isim
- +lar / çoğul

- Yolla / kök_fiil
- + r / zaman_geniş

- Yol / kök_isim
- +lar / isim_şahıs_onlar

# BÖLÜM 3

# DOĞAL DİL İŞLEME

Doğal dil işleme (DDİ) insan dilinin hesaplamalı işleyişini inceleyen yani bilgisayar ile günlük hayatta kullanılan dillerin etkileşimini araştıran bir alandır. Bir başka ifade ile açıklamak gerekirse bilgisayarların insan dilini nasıl anlayabileceğini ve üretebileceğini ortaya koymaya çalışan, dilbilimi, matematik, bilgisayar bilimi, istatistik ve yapay zekâ gibi pek çok alan ile ilişkili çok disiplinli bir çalışma sahasıdır [68]. 20. Yüzyılın ortalarında yapay zekanın bir alt alanı olarak görülen DDİ uygulamaları günümüzde kişisel asistanlar (Apple Siri), çeviri sistemleri (Google Translate), arama motorları (Google, Yahoo, Yandex) ve otomatik deprem raporlama (LA Times) gibi pek çok alanda yaygın bir şekilde kullanılan bir çalışma sahası haline gelmiştir [68,69].

Günümüzde internet kullanımının çok büyük bir şekilde yaygınlaşması ve buna bağlı olarak kullanıcılar tarafından üretilen yapılandırılmamış verilerin muazzam bir şekilde artmış olması nedeniyle DDİ uygulamaları son yıllarda artan bir ivme ile gelişim göstermektedir. İnternet kullanımının yaygın olmadığı dönemlerde verileri veri tabanlarında tutmak ve ihtiyaç duyulan bilgileri Yapılandırılmış Sorgulama Dilini (Structured Query Language - SQL) kullanarak elde etmek mümkündü. Fakat artan yapılandırılmamış veri boyutu nedeniyle bu verileri işleyip anlam çıkarabilen sistemlere olan ihtiyaç da ciddi bir şekilde artış gösterdi.

DDİ uygulamalarının temel amaçlarından bir tanesi günlük hayatta kullanılan konuşma ve yazı dilinin bilgisayarlar tarafından işlenmesini sağlayarak bilgisayarların bu verileri anlamlandırmasını sağlayabilmektir.

DDİ'nin tarihçesi incelendiğinde ilk çalışmaların başladığı tarih olarak genellikle 1950'ler kabul edilmektedir. Alan Turing "Computing Machinery And Intelligence"

isimli makalesinde kendisinin “Taklit Oyunu (Imitation Game)” olarak adlandırdığı daha sonralarda ise kendi ismi ile anılan ve “Turing Testi” olarak isimlendirilen bir ölçüt ortaya koymuştur. Bu ölçüt ile bir makinenin bir insanı taklit edebilme yeteneğini ortaya koyar [70].

1954 yılında Georgetown Üniversitesi ve IBM’in gerçekleştirdiği bir gösterimde ise 60’tan fazla Rusça cümle İngilizce’ye çevrilmiştir. Bu gösterim için oldukça kısıtlı sayıda kelime ve dilbilgisi kuralı kullanılmasına rağmen uluslararası çapta makine çevirisi çalışmalarının yaygınlaşması açısından oldukça ilham verici neticeleri olmuştur [71]. Ayrıca yazarlar makine çevirisi probleminin 3-5 yıllık bir süreç zarfında çözülmüş bir problem olacağını iddia etmişlerdir. Bununla birlikte gerçek ilerleme önerilenin çok çok gerisinde kalmış ve bu alanda yapılan fonlamalar ciddi anlamda azaltılmıştır [72].

Öte yandan 1960’lı yıllarda DDİ işleme alanında bazı önemli kazanımlar elde edilmiştir. Bu yıllarda sınırlı sayıda kelime ve durumu işleyebilme kapasitesine sahip “SHRDLU” yazılımı geliştirilmiştir. Benzer bir diğer çalışma da 1964-1966 yılları arasında Joseph Weizenbaum tarafından yazılmış olan kişi merkezli terapi (Rogerian Terapi) simülasyonu olan ELIZA yazılımıdır. ELIZA insan ile etkileşime geçebiliyordu ve hasta “başım ağrıyor” dediğinde “neden başım ağrıyor dedin” gibi jenerik cümleler kurabiliyordu.

DDİ uygulamaları 1980’lere kadar devam eden süreçte genellikle el ile yazılmış karmaşık kurallara dayanmaktaydı. Bununla birlikte 1980’lerin sonlarına doğru dil işleme uygulamalarında makine öğrenmesi algoritmalarının kullanılmaya başlanmasıyla birlikte DDİ alanında bir devrim yaşandı. Hesaplama gücünün de artmasıyla birlikte DDİ alanındaki çalışmalar artış kaydetti.

Bir bilgisayarın insanların konuştuklarını anlayabilmesi veya bir insanın anlayabileceği şekilde konuşabilmesi için dilin özelliklerine hakkında bilgi sahibi olması gerekir [73]. Bu doğrultuda dilin bilgisayarlar tarafından anlaşılabilir olması için yapılan çalışmalar şu dört grup altında ağırlık kazanmıştır [73]:

- Söz dizimi
- Anlam bilimi
- Ses bilimi
- Biçim bilimi

Dilbilimciler yukarıda belirtilen dört ana başlık altında dilin özelliklerini ortaya koymaya çalışırken DDİ araştırmacıları da elde edilen özellikleri bilgisayarların kullanabileceği bir şekle dönüştürmeye çalışırlar.

DDİ çalışmaları halen gelişmekte olan bir çalışma alanıdır ve birçok araştırmacı bu alanda çalışmalar yürütmeye devam etmektedir.

## 3.1. ÇALIŞMA ALANLARI

DDİ otomatik özetleme, makine çevrimi, metni konuşmaya çevirme, konuşma tanıma, biçimsel bölütleme gibi oldukça geniş birçok çalışma alanına sahiptir. DDİ bu alanlardan bazılarında doğrudan bir gerçek dünya problemi çözülmeye çalışılırken bazılarında ise daha büyük problemlerin çözüme kavuşturulabilmesi adına alt görev olarak kullanılmaktadırlar. Bu bölümde yaygın kullanılan bazı DDİ alanları hakkında kısa bir bilgilendirme yapılmaktadır.

### 3.1.1. Söz Dizimi

Bu başlık altında yapılan en önemli çalışmalardan bir tanesi morfolojik analiz çalışmalarıdır. Morfolojik analiz bu tez çalışmasında da yaygın olarak kullanılan bir görevdir. Bu görev bütünüyle dilde kullanılan kelimelerin karmaşıklığıyla ilgilidir. Örnek vermek gerekirse İngilizce oldukça basit bir morfolojik yapıya sahiptir ve kelimelerin alabildiği ekler oldukça limitlidir. Öte yandan Bölüm 2'de de daha detaylı bir şekilde anlatıldığı gibi Türkçe morfolojik açıdan oldukça zengin eklemeli bir dildir. Bu durumun bir neticesi olarak morfolojik olarak analiz edilmesi oldukça zorlu bir görevdir. Benzer şekilde Fince ve Korece gibi pek çok dil de zengin bir morfolojiye sahiptir.

Basit morfolojiye sahip diller için bir kelimenin olası tüm formlarını ayrı kelime olarak modellemek mümkün olabilirken Türkçe gibi diller için veri boyutunu çok büyük miktarda arttırdığı için kullanışlı değildir. Bu durumda morfolojik analiz yaparak kelimeyi morfemlerine ayırmak ve her bir morfemin ait olduğu grubu tespit etmek gerekir. Böylelikle kelimenin en temel anlamı tespit edilebilir.

Söz dizimi başlığı altında sayılabilecek bir diğer görevde kelime segmantasyonudur. Yani sürekli bir metinin kelimelere bölünmesi işlemidir. Halihazırda kelimelerin boşluklarla ayrıldığı İngilizce ve Türkçe gibi diller için bu görev önemsiz olabilirken kelime sınırlarının tam olarak belirtilmediği Çince, Japonca ve Tayca gibi dillerde metnin içerisindeki kelimeleri tespit etmek açısından oldukça önemli bir görevdir.

Söz dizimi başlığı altında söylenebilecek bir diğer DDİ görevi de Konuşma Parçası Etiketleme (Part-of-Speech Tagger) işlemidir. Esasında bu görev kelimenin tipini belirleme görevidir. Yani verilen bir kelimenin isim, bağlaç, fiil veya sıfat gibi gruplardan hangisine ait olduğunu tespit etme görevidir. İlk bakışta basit bir görev gibi değerlendirilmekle birlikte özellikle MRL'ler için kendine has birtakım zorlukları bulunmaktadır. Örnek 7'deki "yollar" kelimesi incelenecek olursa kelime için birkaç farklı olası durum olabileceği açıkça görülecektir. Birinci durumda kelime "yol" isim kökünden türemiş ve "-lar" çoğul ekini almış olabilir veya "yollamak" fiil kökünden türeyerek "-r" geniş zaman ekini almış olabilir. Örnekte kelimenin her iki durumunu da gösteren iki basit cümle görülmektedir. Birinci cümlede kelime fiil olarak kullanılmışken ikinci cümlede isim olarak kullanılmıştır. Bu durumda konuşma parçası etiketleme görevinde sadece kelimeye bakmak yeterli olmayacaktır. Tüm cümleyi değerlendirmeye alarak kelimenin sınıfı hakkında bir karar vermek gerekli olacaktır.

Örnek 7:

- Kitapları bu hafta yollar.
- Çok geniş yollar yapılıyor.

Bir diğer önemli DDİ görevi de kök çözümleme (lemmatization) işlemidir. Kök çözümleme kelimenin çekimli hallerini bir araya getirme sürecidir. Bu sayede kök ve

kelimenin sözlük formu bir arada analiz edilebilmektedir. Kök bulma (stemming) işleminden farklı olarak kök çözümleme işleminde cümlenin bir kısmını, tamamını veya tüm metni kullanarak kelimenin tam olarak hangi anlamda kullanıldığının tespit edilmesi hedeflenir [74,75].

Yukarıda sayılan görevlerin bir arada yine söz dizimi başlığı altında sayılabilecek cümle sınırlarının tespit edilmesi, terminoloji çıkarımı, kök bulma, cümlelerin ayrıştırma ağacının oluşturulması gibi pek çok DDİ görevi bulunmaktadır.

### 3.1.2. Anlamsal

Sözdizimi temel alanının yanı sıra yine DDİ görev başlığı altında sayılabilecek bir diğer temel görevde anlamsal değerlendirme başlığıdır. Anlamsal değerlendirme konusu oldukça geniş bir kapsama alanına sahip çok geniş bir konudur. Anlamsal değerlendirme başlığı altında sayılabilecek konulardan bazıları şunlardır:

- Optik karakter tanıma
- Soru cevaplama
- Makine çevirisi
- Varlık ismi tanıma
- Doğal dil üretme
- Doğal dil anlama
- İlişki çıkarma
- Duygu analizi

Optik karakter tanıma görevi taranmış bir belge, basılı bir doküman veya fotoğraf gibi ortamlardaki metinlerin tespit edilmesini sağlar. Böylelikle bu karakter bilgisayarlar tarafından düzenlenebilir ve aranabilir hale gelir.

Soru cevaplama uygulamaları DDİ alanları arasından günümüzde en yaygın olarak kullanılanlardan bir tanesidir. Bu uygulama alanı için verilebilecek en güzel örneklerden bir tanesi günümüzde adeta hayatın bir vazgeçilmezi haline gelen arama motorlarıdır. Arama motorları kullanıcının girdiği soruya göre milyarlarca belki daha

fazla olası sonuç arasından en uygun sonucu getirmeye çalışırlar. Bu doğrultuda arama motorunun kullanıcının sorusunu en doğru bir şekilde anlayarak bu soruya en uygun cevabı tespit edip getirmesi gerekmektedir. Bu noktada DDİ teknikleri devreye girmekte ve bu tekniklerden yararlanılarak en uygun sonuçlar tespit edilmeye çalışılmaktadır [76].

Soru cevaplama uygulamaları sorgu yerine kullanıcılar tarafından doğal dilde sorulan soruları anlayarak doğru çözümü getirme konusunda ideal bir çözüm sağlar [76]. Soru cevaplama sistemlerinin temel amacı tüm belgeyi getirmek yerine belgenin içerisindeki cevapları almaktır [76]. Kapalı alan ve açık alan olmak üzere iki tip belge soru cevap sistemi bulunmaktadır. Kapalı alan sistemlerinde çalışma alanları belirli özel konularla sınırlandırılmıştır ve bu konular dahilindeki soruların anlaşılması ve cevaplanması hedeflenmektedir [76]. Açık alan sistemleri ise genellikle web tabanlı sistemlerdir ve herhangi bir özel konu kısıtlaması bulunmamaktadır.

Makine çevirisi de anlamsal olarak değerlendirmeyi gerektiren bir diğer önemli DDİ alanıdır. Yukarıdaki bölümlerde de belirtildiği gibi bu alanda yapılan çalışmalar 1950'lere kadar uzanmaktadır. Zaman zaman makine yardımıyla insan çevirisi uygulamalarıyla karıştırılmaktadır. Fakat makine çevirisi sistemlerinin görevi herhangi bir insan yardımı olmaksızın bir dildeki metni başka bir dile çevirme işlemidir.

Bu alanda kullanılan birden fazla yaklaşım bulunmaktadır. Bu yaklaşımlardan bazıları şöyle sıralanabilir:

- Kural tabanlı çeviri genellikle sözlüklerin ve dilbilgisi problemlerin oluşturulmasında kullanılmaktadır. Diğer yaklaşımlardan farklı olarak hedef ve kaynak dillerin anlamsal, morfolojik ve söz dizimsel yapıları hakkında daha fazla bilgiye ihtiyaç duyan bir yaklaşımdır.

- İnterlingua yaklaşımında kaynak dil önce bir aracı dile dönüştürülmektedir. Sonrasında ise bu aracı dilden hedef dile bir çeviri gerçekleştirilmektedir.

- Transfer tabanlı makine çevirisi yaklaşımı İnterlingua yaklaşımı ile benzerlikler taşımaktadır. Orijinal cümlenin anlamını simüle eden bir ara sunumdan çeviri gerçekleştirilmektedir. Fakat İnterlingua yaklaşımından farklı olarak ara sunum kısmen çeviri yapılan dil çiftine bağlıdır.

- Sözlük tabanlı yaklaşımda da kelimeler bir sözlükteki anlamına bağlı kalınarak doğrudan çevrilirler.

- İstatiksel makine çevirisinde iki dilli derlemlere dayalı olarak istatiksel yöntemler ile çeviriyi üretirler.

- Nöral makine çevirisi yaklaşımında son yıllarda derin öğrenme tabanlı uygulamalar dikkat çekmektedir. Google istatiksel makine çevirisi yerine derin öğrenme tabanlı makine çevirisi kullanacağını açıklamıştır.

- Örnek tabanlı makine çevirisi yaklaşımında da daha önceden yapılan çeviri örneklerine bakılarak yeni bir çeviri yapılmaya çalışılmaktadır.

Varlık ismi tanımada anlamsal başlığı altındaki bir diğer önemli ve yaygın olarak çalışılan DDİ görevidir. Varlık ismi tanıma görevinde doğal dilde yazılmış metinlerin içerisindeki kurum, kuruluş, insan ve yer gibi varlıkların isimlerini tespit edip bunları türlerine göre sınıflandırmak hedeflenmektedir [77,78].

DDİ'nin bir diğer çalışma alanı da doğal dil üretmedir. Bu alan da yapılan çalışmalar ile makinelerin insan gibi konuşması veya doğal dilde metinler üretmesi örnek verilebilir. Apple firmasına ait Siri uygulaması da bunun başarılı örneklerinden bir tanesidir. Metin üretme çalışmalarında ise bir metinden yararlanarak farklı yeni metinler üretme veya farklı konularda yeni metin üretme gibi faaliyetler yürütülmektedir. Otomatik haber üretme [79] veya finansal raporların incelenmesi ve anomalilerin raporlanması [80] gibi farklı konular doğal dil üretmenin ilgi alanları içerisinde bulunmaktadır.

Doğal dil anlamada geniş ölçekli içeriğe sahip, zorlu bir DDİ uygulama alanıdır. Bu başlık altından yürütülen faaliyetler insan robot etkileşiminden gazete makalelerinin anlaşılmasına kadar uzanan geniş bir yelpazeyi kapsamaktadır. Bir sistemin anlayışının derinliği sistemin daha karmaşık problemleri ele alabilmesini sağlamaktadır. Bu durumda sistemin zengin bir kelime ve dilbilgisi kapasitesine sahip olmasına bağlı olarak basit komutların anlayıp işlemesinden öteye geçerek akıcı bir konuşmayı da anlayabilir bir hale getirecektir.

İlişki çıkarma görevi ham metin verileri gibi yapılandırılmamış bilgi kaynakları üzerindeki bilgilerin ortaya çıkartılmasında önemli bir rol oynar. İlişki çıkarma işlemi bilgi çıkarma görevine oldukça benzemektedir fakat ilave olarak tekrarlanan ilişkilerin kaldırılmasını gerektirir.

DDİ çalışmalarının yoğunlaştığı bir diğer faaliyet alanı da duygu analizi konusudur ve bu konu literatürde sıklıkla çalışılan konulardan bir tanesidir. Duygu analizi doğal diller ile yazılmış metinlerin içerisindeki duygusal durumları tespit etmeyi amaçlamaktadır. Tespit edilen duygusal durum olumlu, olumsuz veya nötr şeklinde etiketlenmektedir. Kişinin metnin içerisindeki görüşü anlamsal olarak çok geniş olabilmektedir fakat bunlar genel olarak olumlu, olumsuz veya nötr ifadeler barındırmaktadır [81]. Duygu analizi metinlerin içerisindeki bu ifadeleri tespit etmeyi amaçlamaktadır.

Pek çok alanda duygu analizi uygulamaları yapılmış olmakla birlikte son yıllarda sosyal medya platformları duygu analizi çalışmaları için vazgeçilmez bir veri kaynağı haline gelmiştir.

Yukarıda belirtilen görevlerin yanı sıra DDİ çalışmalarında anlamsal ana başlığı altında sayılabilecek başlık segmantasyonu, sözcüksel anlabilim gibi pek çok alt kol bulunmaktadır.

### 3.1.3. Söylem

Söylem başlığı altında yürütülen DDİ çalışmalarından başlıcaları otomatik metin özetleme, eş atıf çözümleme ve külliyat analizi gibi çalışmalar yer almaktadır. Otomatik özetlemeyi bir veya daha fazla dokümanın içerisindeki önemli bilgileri tutarak dokümanın boyutunun azaltılması faaliyeti olarak tanımlamak da mümkündür [82,83]. Otomatik özetleme çalışmaları çıkarıcı ve yorumlayıcı olmak üzere iki farklı kola ayrılmaktadır [83]. Çıkarıcı işlemlerde istatistiksel analiz yöntemleri uygulanarak dokümanın içerisindeki önemli cümleler tespit edilerek bu cümleler değiştirmeden seçilmektedir [83]. Yorumlayıcı temeline dayanan özetleme çalışmaları ise dokümanın içerisindeki cümleleri dilbilimsel olarak işleyerek bu cümlelere anlamca benzer yeni cümleler kurar [83]. Dolayısıyla çıkarıcı temeline dayanan çalışmalarda dokümanın içerisindeki cümlelerin haricinde yeni bir cümle bulunmazken, yorumlayıcı temeline dayanan çalışmalarda özet metin içerisinde farklı cümleler bulunabilmektedir.

Eş atıf çözümleme çalışmaları da bir metnin içerisindeki aynı kelimeye atıf yapan birden fazla kelimeyi tespit etme çalışmalarıdır. Dönüştü (Anaphora) çözümü bu konuya verilebilecek spesifik örneklerden bir tanesidir. Örnek 8'de dönüştü durumu görülmektedir. İlk cümledeki "İlyas" kelimesi ikinci cümlede zamir olan "O"'ya dönüşmüştür. İlk cümledeki "İlyas", "araba" ve "ev" kelimeleri isim olduğu için "O" zamirinin cümle içerinde tam olarak hangisini işaret ettiğinin tespit edilmesi görevi dönüştü çözümleme görevidir. Daha genel ifadesi ile eş atıf çözümleme görevidir. Buradaki ikinci ifade her zaman zamir olmak zorunda değildir. Cümlede gizli öznenin bulunması gibi durumlarda da benzer sonuçlar ortaya çıkmaktadır.

Örnek 8:

- İlyas arabasını evde bıraktı. O işe taksi ile gitti.

Söylem analizi çalışmalarının amacı ise verilen metin ile ilgili karmaşık sorulara cevaplar bulabilmek ve metinde doğrudan ifade edilmeyen fakat dolaylı bir şekilde anlatılmış olan konuların tespit edilmesidir [84,85].

### 3.1.4. Konuşma

Konuşma tanıma ve metni konuşmaya çevirme gibi uygulamalar konuşma alanındaki DDİ faaliyetlerinin başlıcalarıdır. Bu alanda yapılan çalışmalara çağrı merkezi hizmetleri, kişisel asistanlar ve insan robot etkileşimi pek çok uygulamada karşılaşılmaktadır.

Otomatik konuşma tanımayı, konuşmaların bilgisayarlar ve makineler tarafından metne çevrilmesi olarak tanımlamak mümkündür [86]. Literatürde otomatik konuşma tanıma sistemleri için genellikle istatiksel ve bilgi tabanlı olmak üzere iki farklı yaklaşım kullanılmaktadır [86]. İstatistik tabanlı sistemler veri güdümlü sistemlerdir ve akustik-fonetik bilgiyi farklı makine öğrenmesi algoritmaları ile örtülü olarak kullanırlar. Bilgi tabanlı yaklaşımda ise akustik-fonetik özellikler doğrudan kullanılır. Bu yaklaşımda ayırt edici akustik fonetik özelliklerin konuşmanın tam olarak hangi bölümünde olduğu tespit edilmeye çalışılır [86].

Konuşma alanında yapılan bir diğer DDİ görevi de otomatik konuşma tanımanın tam tersi niteliğindeki metinden konuşma üretme sistemleridir. İnsan konuşmasının yapay olarak üretilmesi işlemi konuşma sentezi olarak adlandırılır. Metinden konuşma üretme sistemleri de doğal dildeki metinleri sentezleyerek konuşmaya çevirirler. Metinden konuşma üretme sistemleri insan - teknoloji arayüzü uygulamaları, görme engelliler için medya ve eğlence sistemleri gibi pek çok alanda kullanılabilmektedir. Tipik olarak bu sistemler metni kompakt bir ses gösterimine dönüştürürler, sonrasında ise bu gösterimi bir sentezleyici kullanarak konuşmaya çevirirler [80].

Yukarıdaki bölümlerde de belirtildiği DDİ sistemleri çok geniş bir çalışma alanına sahiptir. Ayrıca birçok çalışma alanında birden fazla disiplinin bir arada çalışmasına ihtiyaç duyulan zorlu bir görevdir. Bununla birlikte günümüzde DDİ sistemlerinin yaygınlığı da giderek artmakta ve gündelik yaşantıda sıklıkla kullanılır bir hale gelmektedir.

## 3.2. METİN MADENCİLİĞİ

Veri madenciliğinin bir alt kolu olan metin madenciliği doğal dilde üretilmiş metinlerden kullanılabilir yüksek kaliteli bilgilerin çıkarılmasını hedeflemektedir [87]. Metin madenciliği genellikle metin verileri üzerinde yapılan veri madenciliği uygulamaları olarak tanımlanmaktadır [87]. Veri ve metin madenciliklerinin en temel farkları ise veri madenciliği genellikle yapılandırılmış veriler üzerinde çalışırken metin madenciliğinin kullandığı veriler çoğunlukla yapılandırılamamış verilerden oluşmaktadır [87-89]. Bazı araştırmacılar ise metin madenciliğine yapılandırılmamış metin verilerinin veri madenciliği teknikleri uygulanabilmesi için yapılandırılmış hale getirilmesi işlemi olarak yaklaşmaktadırlar [89-92].

Metin madenciliği ile ilişkili bir diğer alanda bilgi getirimi (Information Retrieval) sistemleridir ve bu iki alan birbirleriyle oldukça yakından ilişkilidir. Bu iki terim zaman zaman birbirinin eşdeğeri terimler gibi kullanılmakla birlikte aralarında birtakım farklılıklar bulunmaktadır [87]. Metin madenciliği sistemleri doğal dilde üretilmiş büyük veri koleksiyonları üzerinde çalışır. Ayrıca amaç tüm dokümanı getirmek yerine dokümalardaki örüntüleri ve dokümanların birbirleri arasındaki ilişkileri çıkarmaktır ve metin madenciliği metindeki daha önceden bilinmeyen bilgiyi çıkarma ihtiyacından kaynaklanmaktadır [87].

Metin madenciliği, DDİ, istatistik, makine öğrenmesi ve hesaplamalı dilbilimi gibi pek çok alanın bir arada kullanıldığı çok disiplinli bir çalışma alanıdır. Günümüzde birçok şirket yüksek miktarda metin verisi depolamaktadır. Bunun yanı sıra özellikle Twitter gibi sosyal medya platformlarında her gün oldukça yüksek miktarda yeni metin verisi üretilmektedir. Bu verilerden anlamlı bilgilerin elde edilebilmesi için metin madenciliği ve DDİ çalışmalarına olan ihtiyaç her geçen gün katlanarak artmaktadır. Ayrıca yapılandırılmamış metin verilerinin otomatik olarak anlaşılması ve yorumlanması oldukça zorlu bir görevdir. Bu nedenle bu çalışma alanı akademik anlamda giderek ilgi çekici hale gelmektedir. Ayrıca sosyal medya platformları üzerinde üretilen verinin son derece gürültülü olması bu görevi daha da zorlu hale getirmektedir. Metin madenciliğinin kapsamını anlayabilmek adına bu alanda yapılan çalışmaların incelenmesi yararlı olacaktır.

Metin madenciliğinin çalışma alanlarından bir tanesi metin kategorizasyonu konusudur. Bu alanda yapılan çalışmalar bir metnin konusunun veya kategorisinin ne olduğunu belirlemeyi amaçlamaktadır. Metin kategorizasyonu esasında bir sınıflandırma problemidir ve bu problemin başarılı bir şekilde çözülebilmesi için bir metaforlar, yazım çeşitlilikleri ve eşanlamlılar gibi bir dizi probleminde çözülmesi gerekmektedir. Metin sınıflandırması işleminde bir metin belgesi genellikle kelime çantası (bag-of-words) olarak kabul edilmektedir ve vektör uzay modeli ile gösterilmektedir. Buradaki en önemli problemlerden bir tanesi vektör uzay gösteriminin genellikle son derece seyrek bir yapıya sahip olmasıdır. Ayrıca sosyal medya metinlerinde her bir kelimenin onlarca farklı yazımının bulunuyor olması bu ayrıklığı giderek arttırmaktadır. Bu nedenle kelime çantasını elde etme işleminden önce normalizasyon işleminin yapılması bir gereklilik halini almıştır. Normalizasyon işleminden sonra elde edilen vektör uzay gösterimine makine öğrenmesi algoritmaları uygulanarak metnin ait olduğu sınıf veya sınıflar tespit edilmeye çalışılır.

Benzer nitelikteki belgelerin gruplandırılması olarak tanımlanan metin kümeleme işlemi de metin madenciliğinin çalışma alanlarından bir tanesidir. Sınıflandırma ve kümeleme işlemi tanımsal olarak birbirine benzemekle birlikte esasında farklı görevlerdir. Sınıflandırma işlemi esnasında belgeler önceden belirlenmiş gruplara ayrılırken kümeleme işleminde gruplar önceden belirlenmemektedir.

Günümüzde üretilen veri miktarı her geçen gün giderek büyüyen bir oranda artmaktadır. Bu duruma bağlı olarak bu veriler üzerinden bir karar verilmeye çalışılması durumunda veya dokümanlar arasındaki bağıntılar gösterilmek istediğinde bu işlemi gerçekleştirmek oldukça zorlu olabilmektedir. Bu problemin üstesinden gelebilmek için kullanılan yöntemlerden biri metin görselleştirmedir. Birçok metin madenciliği sistemi kullanıcılarına metinleri görsel olarak ele alma ve aralarındaki bağıntıları keşfetme olanağı tanımaktadır. Metin görselleştirme ile çok büyük boyutlu metin tabanlı veri kaynaklarına görsel anlamda hiyerarşik bir yapı kazandırılmakta, tarama olanakları artırılmakta ve belgelerin haritalanması sağlanmaktadır [93]. Kullanıcı yakınlaştırma, ölçekleme ve alt haritalar oluşturma gibi farklı özelliklerden yararlanmak suretiyle oluşturulan belge haritası ile etkileşime geçebilmektedir.

Veri görselleştirme işlemi genel olarak üç aşamada gerçekleştirilebilmektedir [93,94]. Bu aşamalar sırasıyla:

- Veri hazırlama
- Veri analizi ve çıkarım
- Harita oluşturulması

İlk aşamada görselleştirilecek verinin temin edilmesi veya konunun belirlenmesi gibi işlemler gerçekleştirilir. Sonrasında veriler analiz edilir ve bilgi çıkarımı yapılır. En son olarak da elde edilen analiz sonuçları doğrultusunda verinin görselleştirilmesi yapılır. Görselleştirme işleminin neticesi olarak özellikle çok büyük boyutlu dokümanlarda ilk bakışta görülemeyen yapıların daha kolay bir şekilde keşfedilmesi hedeflenir.

Daha önce DDİ konusu altında detayları verilen metin özetleme konusu da metin madenciliğinin çalışma sahalarından bir tanesidir. Esasında veri madenciliği, metin madenciliği, bilgi çıkarımı ve DDİ gibi konuları birbirinden kesin çizgilerle ayırmak pek mümkün değildir. Çalışma sahaları pek çok noktada kesişmektedir.

Metin madenciliği ve DDİ konuları oldukça geniş alt sahalara sahip konulardır ve müşteri ilişkileri yönetimi, doğru reklam gösteriminin sağlanması, metin kaynaklarının ulusal güvenlik amacıyla izlenmesi [95], biyomedikal alanında protein-hastalık ilişkilerinin belirlenmesi [96], protein etkileşimlerinin ortaya çıkarılması [97,98] veya film yorumlarının değerlendirilmesi [99] gibi çok sayıda uygulama alanına sahiptirler.

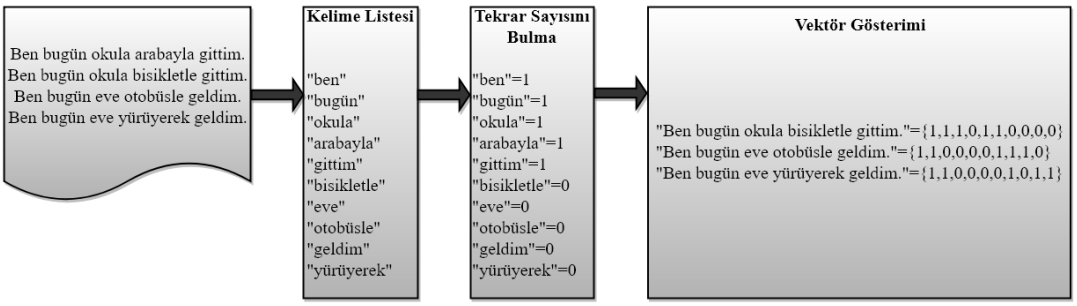
# BÖLÜM 4

# TEZ ÇALIŞMASINDA KULLANILAN METOTLAR

Bu bölümde tez çalışmasında kullanılan kelime temsilleri, metinlerin vektörel ifadesi ve kullanılan yapay sinir ağı teknikleri gibi kavramlar hakkında kısaca açıklamalar yapılacaktır.

## 4.1. METİNLERİN VEKTÖREL İFADESİ

Kelime çantaları ve n-gramlar bir metni vektörel olarak ifade etmek için yaygın olarak kullanılan yöntemlerdendir. Kelime çantası yaklaşımında dokümanların içerisindeki kelimelerin her biri benzersiz bir özellik olarak ele alınır ve bu benzersiz özelliklerin her birinin ilgili doküman veya SMM içerisindeki tekrar sayısı bulunur. Sonuç olarak dokümanın vektörel bir temsili elde edilmiş olur. Şekil 4.1'de kelime çantası yaklaşımı 4 farklı cümlenin vektörel temsillerinin elde edilişi görülmektedir.




Şekil 4.1. Kelime çantası yöntemi örneği.

Kelime çantası modeli kelimelerin sırasız bir doküman gösterimi yaklaşımıdır yani bu yaklaşım ile elde edilen vektörel gösterimde kelimelerin cümle içerisinde hangi sıra ile bulunduğu hakkında bir bilgi saklanmaz. Önemli olan kelimelerin doküman içerisinde kaç kez bulunduğudur. Öte yandan n-gram modelinde kelimelerin mekânsal

konumunda saklanmış olsa da yüksek boyutluluk ve veri seyrekliği gibi problemlerle karşılaşılmaktadır.

Bunlara ilave olarak metin formatındaki verileri sayısal temsiller şeklinde ifade etmek için farklı yöntemler mevcuttur. Metin kelime dizileri şeklinde ifade edilerek her bir kelime bir vektörle temsil edilebileceği gibi metni karakter dizileri şeklinde ifade ederek her bir karakteri bir vektör ile temsil etmekte mümkündür veya benzer şeklinde kelime n-gramlarından oluşan dizileri de vektörel olarak temsil etmek mümkündür.

Trafik verilerinin sınıflandırılması işleminde her bir tweet kelime dizisi şeklinde ifade edilmekte ve her bir kelimede kelime temsil modeli ile sayısal bir vektör olarak gösterilmektedir. Kelime temsil modelleri kelime çantası ve n-gram modellerinde karşılaşılan problemlerin üstesinden gelmek için kullanılmaya başlanmıştır. Kelime temsil modellerinde her bir kelime bir vektör ile temsil edilmektedir. Bu yaklaşımda birbirine yakın anlama gelen kelimeler birbirine benzer vektörler ile ifade edilmektedirler. Ayrıca her bir kelimenin düşük boyutlu vektörler ile ifade edilmesi nedeniyle boyutsallık ve veri seyrekliği problemi büyük ölçüde ortadan kaldırılmaktadır.

## 4.2. KELİME TEMSİLİ (WORD EMBEDDING)

Kelime temsili, kelimelerin anlamsal özelliklerini yakalamayı hedefleyen düşük boyutlu, yoğun vektörel temsillerdir [100]. Ayrıca DDİ'de dil modelleme ve özellik öğrenme tekniklerinin genel ismi olup, kelimeler gerçek sayılardan oluşan vektörlerle eşleştirilmektedirler. Böylelikle kelimeler düşük boyutlu vektörlerle temsil edilebilmekte ve kelimelerin arasındaki anlamsal ilişki vektörel benzerlik ölçümleri ile kıyaslanabilmektedir.

Kelimelerin vektörel temsillerini oluşturmanın çeşitli yöntemleri bulunmaktadır. Bunlardan başlıcaları, yapay sinir ağları kullanmak [101], eş-oluşum (co-occurrence) matrisleri üzerinde boyut azaltma [102-104], olasılıksal modeller [105], bilgi tabanlı yöntemler [106] ve açık vektör uzay gösterimidir [107].

Kelime temsil yaklaşımının söz dizimsel ayrıştırma [108], duygusal analiz [109,110], metin benzerliği tespiti [111], konuşma parçası etiketleme [112] ve makine çevirisi [113] gibi pek çok DDİ görevinde performansı arttıran bir etkisinin olduğu ortaya konulmuştur.

Kelime temsilleri bu tez çalışmasında iki farklı şekilde kullanılmaktadır. İlk kullanım yerinde normalizasyon modüllerinin belirsizlik durumları için genişletilmesi işlemi gerçekleştirilmektedir. Bu durumda üretilen aday kelime ile mevcut kelimelerin vektörel benzerlik değerleri kıyaslanarak en uygun kelimenin seçimi yapılmaktadır. Bir diğer kullanım yeri de sınıflandırma işlemi esnasında tweet içerisindeki her bir kelimenin vektörel temsilini sağlamak içindir.

Kelime temsil modeli olarak Word2vec kullanılmıştır. Word2vec bir derlemi girdi olarak alan ve buna karşılık bir vektör setini çıktı olarak üreten sığ bir nöral networktür [114]. Word2vec kelimeleri sabit uzunluklu vektörler ile temsil etmektedir. Bu vektörlerin en temel özelliği, ilgili kelime hakkında anlamsal bilgiler taşımasıdır [101,115,116]. Word2vec modeli verilen derlem için kelimelerin global eş-oluşum istatistiklerini göz ardı etmektedir. Bunun yerine sadece sözcüklerin bulundukları bağlam pencerelerini tüm derlem boyunca ele alır [117,118]. Nöral networkler pek çok alanda yaygın bir şekilde kullanılmaktadır [119-122]. Word2vec temelde Skip-Gram ve Continuous bag of words (CBOW) olmak üzere 2 farklı nöral modeli kullanarak kelimeleri bulundukları bağlama göre tahmin ederler.

Word2vec yöntemi sayesinde derlemde bulunan kelimeler arasında anlamsal ve yapısal bir ilişki kurulabilmektedir [118]. Ayrıca sözcükler arasındaki anlamsal ve yapısal ilişki vektörler arasındaki mesafeye aktarılarak sözcüklerin birbiriyle ilintili olma durumu cebirsel olarak kontrol edilebilmektedir [118]. Yani iki vektör arasındaki uzaklığa veya kosinüs benzerliğine bakarak sözcüklerin bir arada bulunabilme durumları veya anlamsal benzerlikleri kıyaslanabilmektedir.

CBOW ve Skip-gram modelleri birbirine zıt iki temel yaklaşım sergilemektedirler. CBOW modeli kelimeyi tahmin etmek için içerisinde bulunduğu bağlamı esas alırken Skip-gram modeli bunun aksine mevcut kelimeyi kullanarak bu kelimenin etrafında

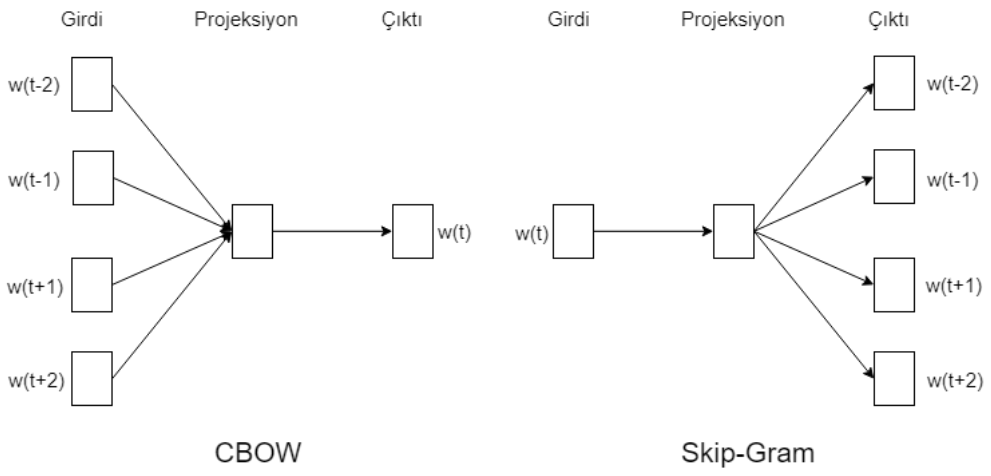
bulunabilecek diğer kelimeleri tahmin etmeye çalışır. CBOW modeli hata gradyanının geri yayılımına bağlı olarak kelime ve çıktıyı karşılaştırıp kelimenin temsilini (representation) düzeltir. Esasında CBOW aşağıdaki denklemi maksimize etmeye çalışır:

$$\frac{1}{V}\sum_{t=1}^{V}\log = p(m_t \mid m_{t-\frac{c}{2}}...m_{t+\frac{c}{2}}) \tag{4.2}$$

Öte yandan Skip-gram modeli bir kelimeyi esas alarak bu kelimenin etrafında bulunması muhtemel olan diğer kelimeleri bulmaya çalışır ve temelde aşağıda verilen denklemi maksimize etmeye amaçlar:

$$\frac{1}{V}\sum_{t=1}^{V}\sum_{j=t-c, j\neq t}^{t+c}\log p(m_j \mid m_t) \tag{4.3}$$

Şekil 4.2'te CBOW ve Skip-gram modellerinin görünümleri gösterilmektedir. Kelimelerin vektörel temsilleri elde edildikten sonra normalizasyon işlemi için Bölüm 4.3'te detaylı verilen kosinüs benzerliği ve diğer vektörel uzaklık ölçüm yöntemleri ile kelimelerin arasındaki ilişki değerlendirilir. Elde edilen kelime temsillerinin trafik verilerinin sınıflandırılması işleminde kullanımı da Bölüm 7.1'de tanıtılmaktadır.




Şekil 4.2. CBOW ve Skip-gram modellerinin gösterimi.

## 4.3. VEKTÖREL BENZERLİK

Bu tez çalışmasında Word2vec yöntemiyle elde edilen kelimelerin vektörel gösterimleri arasındaki ilişki aşağıdaki bölümlerde detayları verilen Kosinüs Benzerliği, Öklid Uzaklığı, Manhattan Uzaklığı, Minkowski Uzaklığı ve Chebyshev Uzaklığı yöntemleri kullanarak hesaplanmıştır.

### 4.3.1. Kosinüs Benzerliği

Bu tez çalışmasında kullanılan vektörel benzerlik ölçümlerinden bir tanesi kosinüs benzerliğidir. Kosinüs benzerliği bu tez çalışmasında kullanılan diğer vektörel uzaklık ölçümlerinden farklıdır. Şöyle ki kosinüs benzerliği iki ayrı vektörün arasındaki mesafenin büyüklüğü değil birbirlerine göre göreceli oryantasyonlarıdır. Dolayısıyla diğer yöntemlerde vektörler arasındaki uzaklık değerlendirilirken, kosinüs benzerliğinde birbirlerine göre olan yönelimleri değerlendirilmektedir. Şekil 4.3'te kosinüs benzerliğinin gösterimi görülmektedir.

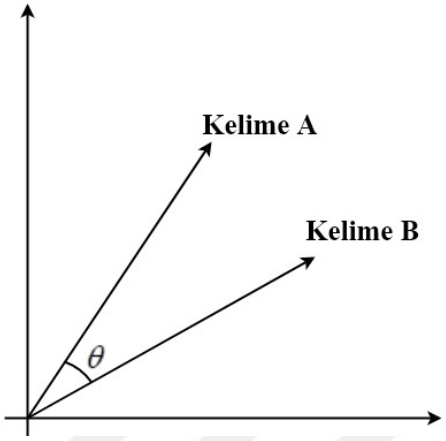
$a(x_1, y_1)$ ve $b(x_2, y_2)$ iki boyutlu uzayda verilen 2 nokta olmak üzere bu iki nokta arasındaki kosinüs benzerliği şu şekilde yazılabilir:

$$\cos\theta = \cos(a,b) = \frac{a \cdot b}{\|a\|\|b\|} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + x_2^2} \times \sqrt{y_1^2 + y_2^2}} \tag{4.4}$$

Öte yandan boyutu artıracak olursak $a$ ve $b$ vektörlerini $a(a_1, a_2, a_3 ... a_n)$ ve $b(b_1, b_2, b_3 ... b_n)$ şeklinde gösterebiliriz. Bu durumda Eşitlik (4.4)'ü şu şekilde tekrar yazabiliriz:

$$\cos\theta = \cos(a,b) = \frac{\sum_1^n (a_i \times b_i)}{\sqrt{\sum_1^n a_i^2} \times \sqrt{\sum_1^n b_i^2}} \tag{4.5}$$

Burada $\cos\theta$ değeri [0,1] aralığındadır ve 0 iki kelime arasında hiçbir anlamsal ilişki bulunmadığını ifade ederken 1 değeri ise kelimenin aynı anlama geldiğini ifade eder.




Şekil 4.3. Kosinüs benzerliği gösterimi.

### 4.3.2. Öklit Uzaklığı

Öklit uzaklığında iki nokta arasındaki mesafe Pisagor bağıntısı kullanılarak hesaplanmaktadır. Şekil 4.4'te Öklid uzaklığı görülmektedir.

$a(x_1, y_1)$ ve $b(x_2, y_2)$ yine iki boyutlu uzayda verilen 2 nokta olmak üzere bu iki nokta arasındaki Öklid uzaklığı şu şekilde hesaplanabilir:

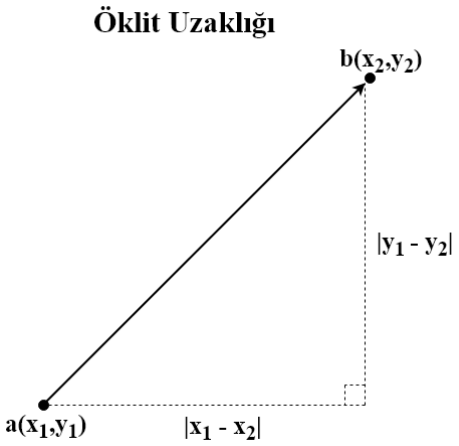
$$d(a,b) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4.6}$$

Benzer şekilde çok boyutlu $a(a_1, a_2, a_3 ... a_n)$ ve $b(b_1, b_2, b_3 ... b_n)$ vektörleri için iki vektör arasındaki uzaklık *d(a,b)* şu şekilde ifade edilebilir:

$$d(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2} \tag{4.7}$$

Bu durumda eşitlik şu şekilde tekrar yazılabilir:

$$d(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{4.8}$$

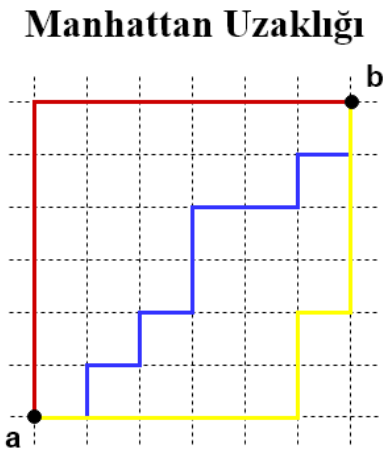



Şekil 4.4. Öklid uzaklığı gösterimi.

### 4.3.3. Manhattan Uzaklığı

Manhattan uzaklığı iki nokta arasında ızgara benzeri bir yol izler. Gözlemler arasındaki toplam mutlak uzaklık alınarak hesaplanmaktadır. Manhattan uzaklığı idealize edilmiş bir şehri simgelemektedir ve Şekil 4.5'da görüleceği üzere 2 nokta arasında uzanan üç farklı renkteki yolların tamamı aynı uzunluktadır.

Yukarıdaki bölümlerde belirtilen *a* ve *b* çok boyutlu vektörleri için Manhattan uzaklığı şu şekilde yazılabilir:




Şekil 4.5. Manhattan uzaklığı gösterimi.

$$d(a,b) = \sum_{i=1}^{n} |a_i - b_i| \tag{4.9}$$

### 4.3.4. Minkowski Uzaklığı

Minkowski uzaklığı, Manhattan ve Öklid uzaklıklarının genelleştirilmiş bir formu olarak düşünülebilir.

Benzer şekilde yukarıdaki bölümlerde belirtilen *a* ve *b* çok boyutlu vektörleri için *p* dereceyi ifade etmek üzere Minkowski uzaklığı şu şekilde yazılabilir:

$$d(a,b) = (\sum_{i=1}^{n} | a_i - b_i |^{p})^{1/p} \tag{4.10}$$

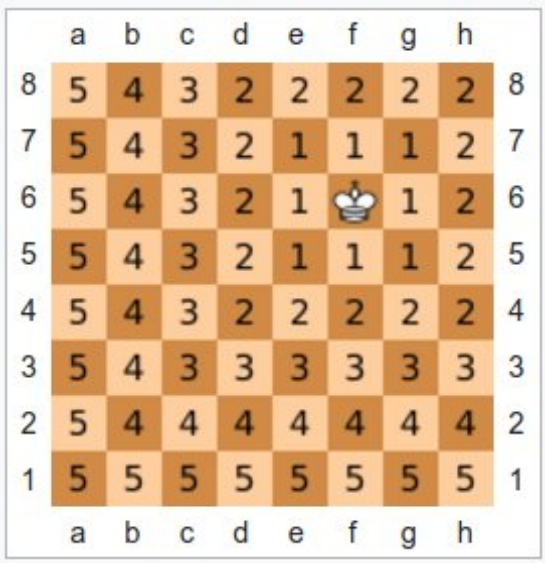
Burada *p* değerinin 1 olması durumunda Manhattan uzaklığı bulunmuş olur. Öte yandan *p* değerinin 2'ye eşit olması durumunda ise Öklid uzaklığı bulunmuş olacaktır. Son olarak *p*'nin limit değerinin ($+\infty$) yaklaşması durumunda ise Chebyshev uzaklığı elde edilmiş olacaktır.

### 4.3.5. Chebyshev Uzaklığı

Chebyshev uzaklığı, satranç tahtası uzaklığı olarak da bilinmektedir. Bu uzaklık ölçüsü Şekil 4.6'da görüleceği üzere satranç tahtası üzerindeki bir şahın hareketine benzemektedir ve şahın 2 farklı nokta arasında hareket edebilmesi için gerekli olan minimum mesafeyi ifade eden bir ölçüttür.

Aynı a ve b çok boyutlu vektörleri için Chebyshev uzaklığını şu şekilde ifade etmek mümkündür:

$$d(a,b) = \max_{i}(| a_i - b_i |) \tag{4.11}$$




Şekil 4.6. Chebyshev uzaklığı gösterimi.

### 4.4. N-GRAM

N-gram DDİ ve metin madenciliği uygulamalarında yaygın olarak kullanılan bir yöntemdir ve belirli bir metin veya konuşmadaki n adet elemandan oluşan bitişik dizileri ifade etmek için kullanılır. N-gram değeri bulunmak istenen öğe kelime, hece veya harf gibi herhangi bir öğe olabilir. Metin kategorizasyonu [123], otomatik özet değerlendirme [124], dil modelleme [125] ve kötü amaçlı yazılım sınıflandırması [126] gibi pek çok alanda kullanılmaktadır.

Çizelge 4.1. "günaydın" kelimesinin karakter seviyesi bigram örnekleri.

| **günaydın** | |
|---|---|
| **a** | **b** |
| gü | _g |
| ün | gü |
| na | ün |
| ay | na |
| yd | ay |
| dı | yd |
| ın | dı |
| | ın |
| | n_ |

Kullanılan “N” parametresinin ilk üç değeri için özel olarak sırasıyla unigram, bigram ve trigram olarak da isimlendirilmektedir. N’nin üçten büyük değerleri için yalnızca N-gram olarak isimlendirilmektedir. Çizelge 4.1’de “günaydın” kelimesinin iki farklı şekilde elde edilen bigram modeli görülmektedir. Çizelge 4.1’in a sütunundaki bigram modelinde kelime doğrudan çözümlenirken, b sütunundaki örnekte karakterin kelimenin başında veya sonunda bulunma olasılıklarını da tespit edebilmektedir. Aynı durumu kelime seviyesinde bir N-gram modeli elde ederken kullanmak da mümkündür. Çizelge 4.2’de N’nin 1’den 5’e kadar olan değerleri için elde edilmiş kelime seviyesinde N-gram modeli görülmektedir.

Çizelge 4.2. 1'den 5'e kadar kelime seviyesi n-gram örnekleri.

| Cümle | Van Gölü Türkiye’nin en büyük gölüdür. |
|---|---|
| Unigrams | {Van}, {Gölü}, {Türkiye’nin}, {en}, {büyük}, {gölüdür} |
| Bigrams | {Van Gölü}, {Gölü Türkiye’nin}, {Türkiye’nin en}, {en büyük}, {büyük gölüdür} |
| Trigrams | {Van Gölü Türkiye’nin}, {Gölü Türkiye’nin en}, {Türkiye’nin en büyük}, {en büyük gölüdür} |
| Four-grams | {Van Gölü Türkiye’nin en}, {Gölü Türkiye’nin en büyük}, {Türkiye’nin en büyük gölüdür} |
| Five-grams | {Van Gölü Türkiye’nin en büyük}, {Gölü Türkiye’nin en büyük gölüdür} |

Öte yandan N-gram kullanımında karşılaşılan bazı kısıtlamalar da bulunmaktadır. Bunlardan bir tanesi bağımlılık aralığının (N-1) olmasıdır. Dolayısıyla tüm metnin uzun bağımlılıklarını bulmaktan yoksundur. Dilbilimsel bilgiyi modellemek için tasarlanmamıştır fakat pratik kullanımlar için uygundur. Bir diğer konu da özellikle Türkçe gibi kelimelerin çok fazla sayıda yüzey formuna sahip olduğu dillerde derlemin yeterince güçlü bir temsile sahip olamaması ve aranılan kelimenin derlemde bulunmamasıdır.

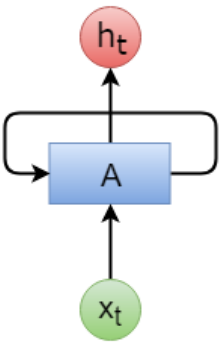
## 4.5. TEKRARLAYAN YAPAY SİNİR AĞLARI

Yapay sinir ağları pek çok makine öğrenmesi uygulamasında yaygın olarak kullanılmaktadır. Zaman içerisinde farklı yapay sinir ağı modelleri farklı veri tiplerini

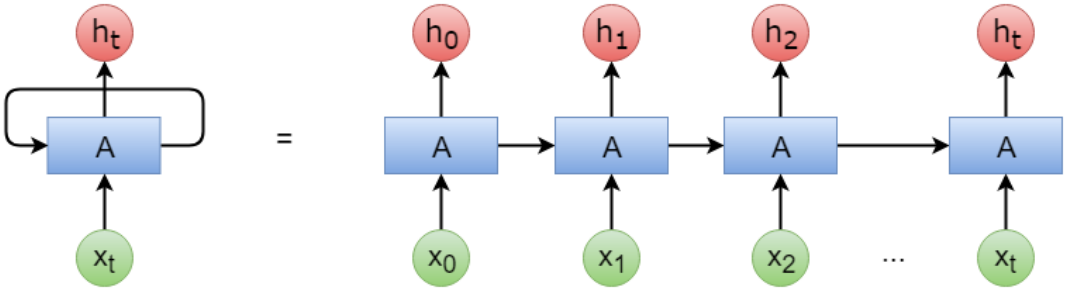
işlemek üzere özelleşmiştir. Örneğin konvolüsyonel yapay sinir ağları resim verisi gibi matris tarzı verilerin işlenmesi için özelleşmişken tekrarlayan yapay sinir ağları (Recurrent Neural Networks - RNN) da dizi verilerini işlemek üzere geliştirilmiştir [127-128]. Geleneksel ileri beslemeli yapay sinir ağlarının dikkate aldığı tek girdi maruz kaldığı mevcut örneklerken, RNN'ler bundan farklı olarak mevcut örneklerin yanı sıra zaman içerisinde algıladıklarını da girdi olarak uygularlar.

$x_i \in \square^d$ olmak üzere $[x_1, x_2, ..., x_k]$ şeklinde verilen bir girdi dizisi verilmiş olsun. Burada farklı örnekler farklı dizi uzunluklarına sahip olabilirler ve dolayısıyla $k$ değeri değişkenlik gösterebilir. RNN modelinin her bir adımında $[h_1, h_2, ..., h_k]$ dizisi şeklinde bir gizli durum üretilir. t zaman adımındaki gizli durumun aktivasyonu mevcut girdi $x_t$'nin ve önceki gizli durum $h_{t-1}$'in bir fonksiyonu olarak şu şekilde hesaplanır:

$$h_t = f(x_t, h_{t-1}) \tag{4.12}$$




Şekil 4.7. Tekrarlayan yapay sinir ağı (kapalı).

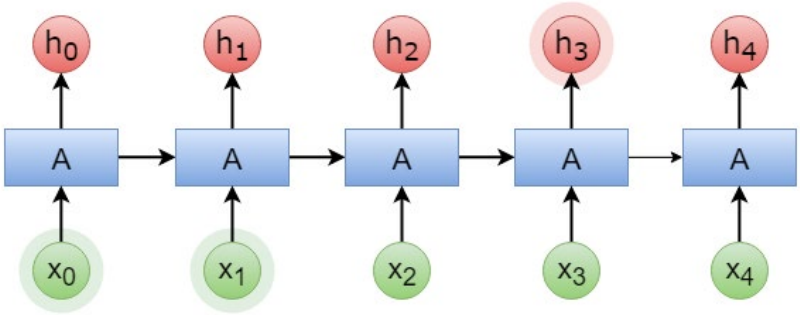



Şekil 4.8. Tekrarlayan yapay sinir ağı (açık).

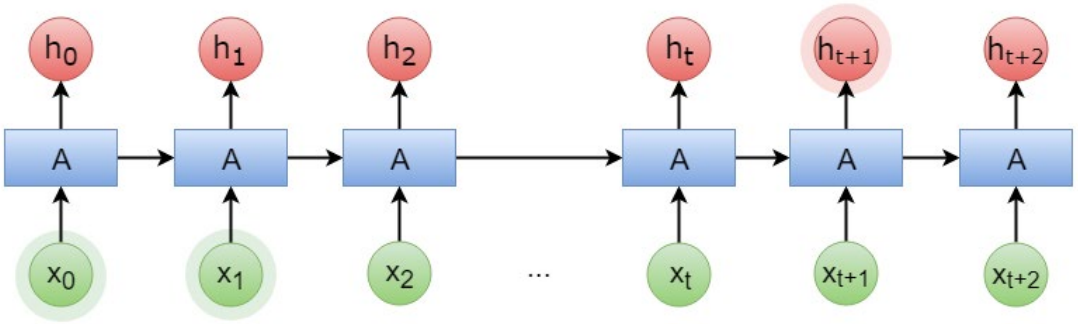
RNN'lerde geleneksel ileri beslemeli yapay sinir ağlarından farklı olarak bir tekrarlama katmanı mevcuttur. Bu katman vasıtasıyla ileri beslemeli ağın ürettiği durum bilgisi saklanarak girdi bilgisiyle birlikte ağa yeniden uygulanmaktadır. Bir başka ifadeyle RNN'ler mevcut ana kadar neyin hesaplandığını tutan bir belleğe sahiptir. Şekil 4.7 ve Şekil 4.8'de sırasıyla örnek bir RNN ağı ve bu ağın açılmış ileri beslemeli hali görülmektedir.

## 4.6. UZUN KISA DÖNEM HAFIZA

Genellikle LSTM (Long-Short Term Memory) olarak isimlendirilen uzun - kısa dönem hafıza ağları, uzun dönemli bağımlılıkları öğrenme kabiliyetine sahip özel bir RNN türüdür. İlk kez 90'ların ortasında önerilen [129] bu model günümüzde yaygın olarak kullanılmaktadır. RNN'lerde diziler üzerinde işlem yaparken yapay sinir ağının durum bilgisinin saklanması ve aktarılması hedeflenmiş olsa da durum bilgisi üzerinde sürekli olarak işlem yapılarak aktarılmasının bir neticesi olarak uzun süreli bağımlılıklar bozulmadan aktarılması pek mümkün değildir. Yani dizi içerisindeki kısa süreli bağımlılıklar oldukça başarılı bir şekilde aktarılırken uzun dönemli bağımlılıkların aktarılmasında problem yaşanmaktadır. LSTM'ler ise uzun vadeli bağımlılık problemini gidermek amacıyla tasarlanmıştır. Şekil 4.9'da kısa süreli bağımlılıklar görünürken Şekil 4.10'da uzun süreli bağımlılıklar görülmektedir.

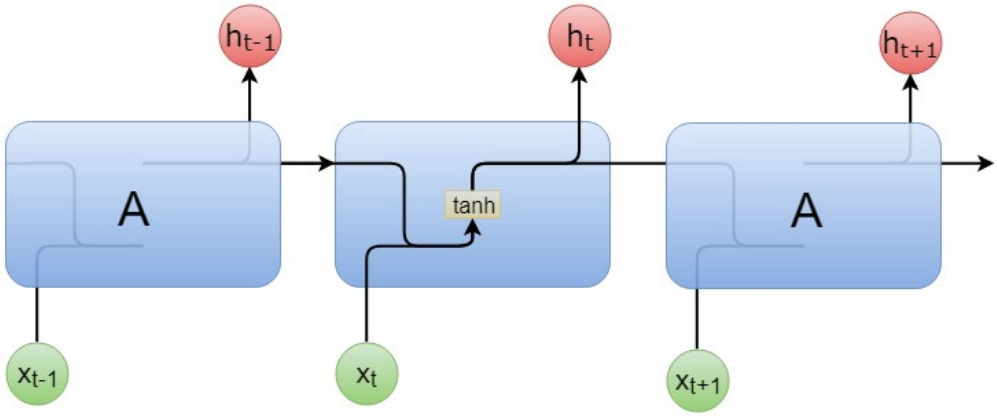



Şekil 4.9. Kısa süreli bağımlılıklar.

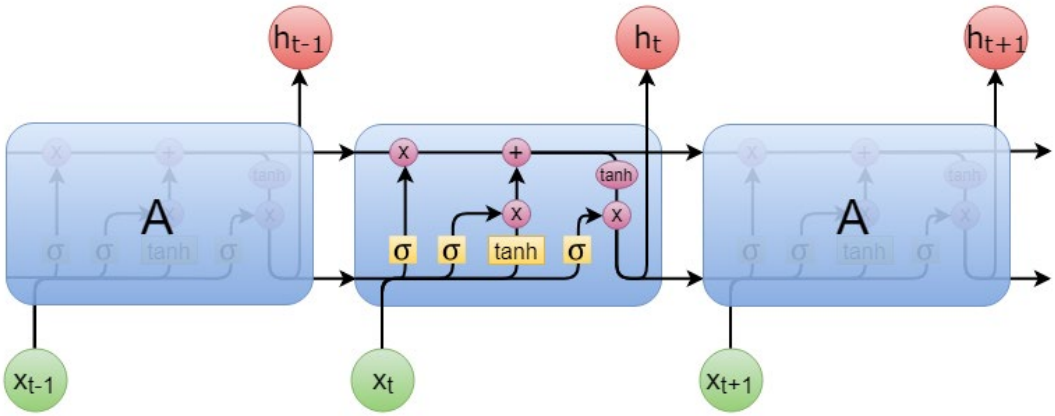



Şekil 4.10. Uzun süreli bağımlılıklar.

RNN ağlarının tümü zincir şeklinde tekrarlayan modüllerden meydana gelmektedir. Standart RNN'lerde bu modüllerden her biri genellikle tanh katmanı veya benzer bir tek katmandan oluşmaktadır. Şekil 4.11'de standart bir RNN'ye ait modül yapısı görülmektedir. LSTM'leri standart RNN'lerden ayıran özellik ise bu modülün iç yapısının Şekil 4.12'de görüldüğü gibi birbirleriyle etkileşim halinde olan 4 ayrı yapıdan meydana gelmesidir.



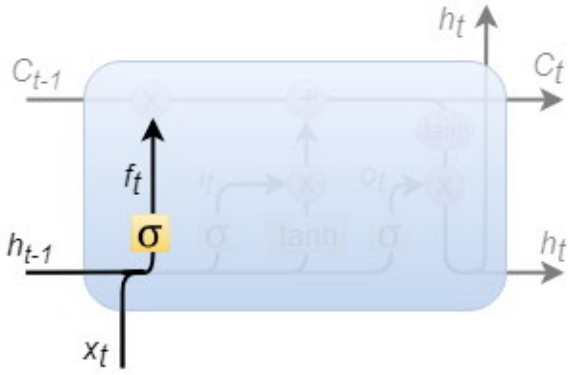

Şekil 4.11. Standart bir RNN'deki yinelenen modül, tek bir katman içerir.




Şekil 4.12. Bir LSTM'deki yinelenen modül, dört etkileşimli katman içerir.

LSTM modülü yapısı itibariyle 3 ayrı kapıdan oluşmaktadır. Bu kapıların isimleri sırasıyla unutma kapısı, girdi kapısı ve çıktı kapısıdır. Unutma kapsısı adı verilen ilk kapıda bilginin ne kadarının unutulacağı ve ne kadarlık bir kısmının bir sonraki safhaya aktarılması gerektiğine karar verir. Bu işlem için 0 ile 1 arasında bir değer üreten sigmoid katmanına sahiptir. 0 bilginin hiçbir kısmının iletilmeyeceği anlamına gelirken 1 ise tamamının iletilmesi gerektiği anlamını taşımaktadır. Şekil 4.13'de görülen unutma kapsının matematiksel modelini şu şekilde ifade etmek mümkündür:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4.13}$$




Şekil 4.13. LSTM unutma operasyonu.

Bir sonraki adımda yapılması gereken işlem ise hangi bilgilerin depolanması gerektiğine karar vermektir. Bu aşamada öncelikli olarak 2. Sigmoid katmanı yani girdi katmanı olarak isimlendirilen katman hangi değerlerin güncellenmesi gerektiğine

karar vermektedir. Sonrasındaki tanh katmanı $\tilde{C}_t$ olarak ifade edilen yeni aday değerlerin bir vektörünü oluşturmaktadır ve sonrasında bu iki işlem birleştirilmektedir. Bu işlem matematiksel olarak şu şekilde ifade edilmektedir:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4.14}$$

$$\tilde{C} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4.15}$$

Bu işlemin sonrasında ise hafıza hücresinin yeni durum bilgisinin hesaplanması gerekmektedir. Bu durumda yeni durum bilgisi şu şekilde hesaplanır:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4.16}$$

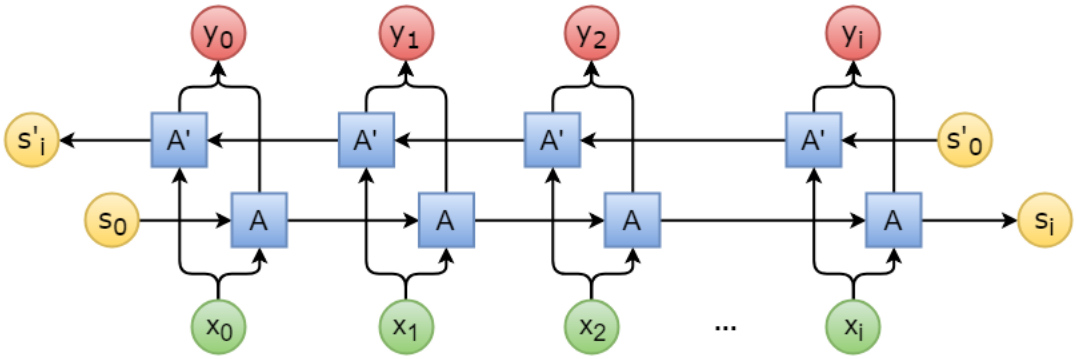
En son aşamada ise sistemin çıktısı $h_t$ hesaplanır. Bu işlem çıktı kapısında gerçekleştirilmektedir ve sitemin çıktısı $h_t$ şu şekilde hesaplanabilir:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{4.17}$$

$$h_t = o_t * \tanh(C_t) \tag{4.18}$$

## 4.7. ÇİFT YÖNLÜ UZUN KISA DÖNEM HAFIZA

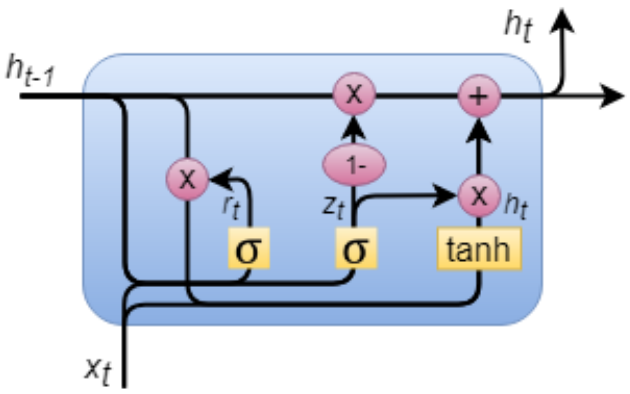
Standart RNN, GRU ve LSTM modellerinde önceki zaman adımlarının oluşturduğu temsillerin öğrenilmesidir. Burada önemli bir konu bazen NLP uygulamalarında içeriğin daha iyi kavranabilmesi ve belirsizliklerin ortadan kaldırılabilmesi için gelecekteki temsillerinde öğrenilmesi gerekebilir. Çift yönlü uzun kısa dönem hafızayı meydana getiren modüller Şekil 4.14'deki gibidir. Buradaki temel farklılık ilerim yayılım işleminin iki aşamada hesaplanmasıdır. Öncelikli olarak ilk zaman adımdan başlanarak son zaman adımına kadar değerler hesaplanmaktadır. Sonrasında ise son zaman adımından başlanarak ilk zaman adımına kadar değerler hesaplanır.




Şekil 4.14. Çift yönlü RNN ileri-geri yayılımı şematik gösterimi.

## 4.8. KAPILI TEKRARLAMALI ÜNİTE - GRU AĞI

Kapılı tekrarlamalı ünite (Gated Recurrent Unit - GRU) ağının LSTM ağından temel farkı Şekil 4.15'te de görüldüğü gibi her bir modülün 3 yerine 2 kapıdan meydana gelmesidir[130]. Bir GRU modülü güncelleme kapısı ve sıfırlama (reset) kapısından meydana gelir. Güncelleme kapısı geçmiş bilgilerin ne kadarının geçmesi gerektiğine karar verirken sıfırlama kapsısı ise bunun tam aksine geçmişten gelen bilgilerin ne kadarının atılması gerektiğine karar verir.




Şekil 4.15. Kapılı tekrarlamalı ünite.

$z_t$ güncelleme geçidini temsil eden sigmoid işlemi ve $\tilde{h}_t$ sıfırlama işlemi olmak üzere GRU matematiksel olarak şu şekilde ifade edilebilir:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{4.19}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{4.20}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{4.21}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{4.22}$$

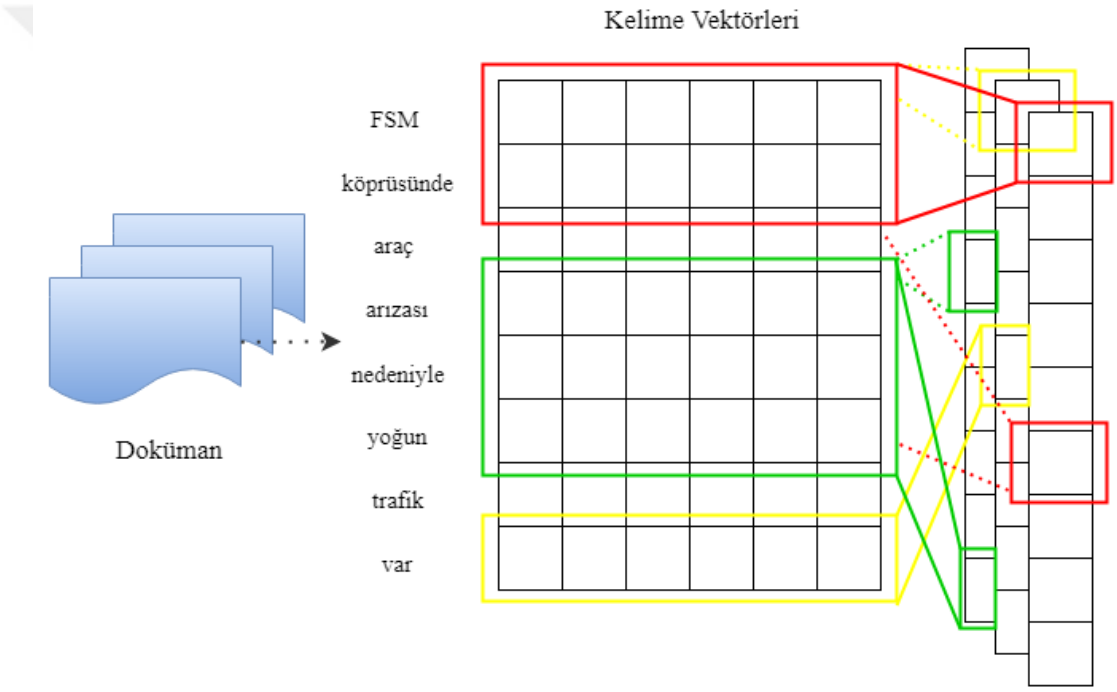
### 4.9. CNN

Esasında tipik birçok katmanlı yapay sinir ağı modeli olan CNN, ilk olarak bilgisayar görmesi problemleri için önerilmiştir [131] ve resim ile ilgili birçok uygulamada başarılı bir şekilde kullanılmaktadır. CNN modelleri yaygın olarak konvolüsyon, havuzlama ve tam bağlı katmanların bir araya getirilmesi ile oluşturulur.

Konvolüsyon katmanının tanımlanması için filtre boyutu ve üretilen haritaların sayısı kullanılır. Bu katman CNN'yi oluşturan en temel ve en önemli birimdir. Konvolüsyon katmanının temel hareket noktası bir nesneye ait görüntünün resim üzerinde nerede olduğundan bağımsız olmasıdır ve bu fikir doğrultusunda nöronlar girişin yalnızca küçük bir kısmına bağlanmakta ve girişe uygulanan verinin tüm derinliği boyunca da uzanmaktadır. Sonrasında ise ileri yayılım safhasında giriş verisi ile filtreler arasında nokta çarpımı işlemi gerçekleştirilerek 2 boyutlu aktivasyon haritası meydana getirilmektedir. Bununla birlikte CNN modelinin öğrenmesi istenilen bilgi genellikle doğrusal olmayan bilgilerdir fakat konvolüsyon işlemi matris çarpımı ve toplaması gibi lineer işlemleri içeren bir yöntemdir. Bu sebepten dolayı ReLU (Rectified Linear Units) ile f(x)=max(0,x) doyma olmayan bir aktivasyon fonksiyonu uygulanmak suretiyle CNN modeline doğrusal olmama özelliği kazandırılır. Bunun yanı sıra bir diğer önemli katmanda aşağı örnekleme işlemelerinin gerçekleştirildiği havuzlama katmanıdır. Literatürde kullanılan birçok havuzlama yöntemi bulunmaktadır fakat en yaygın olarak iki yöntem sırasıyla maksimum havuzlama ve ortalama havuzlama yöntemleridir. Bu çalışmada maksimum havuzlama yöntemi kullanılmıştır. Bu yöntemde giriş örtüşmeyen dikdörtgenlere ayrılmakta ve her bir alt parça arasından

yalnızca en büyük değere sahip olan alınmaktadır. Tam bağlı katmanda bulunan nöronlar ise geleneksel yapay sinir ağlarında olduğu gibi hem kendisinden önceki hem de kendisinden sonraki tüm katmanlara bağlıdırlar ve çıkış katmanında katmanında softmax veya başka bir sınıflandırıcı ile kullanılabilirler.

Farklı katmanlar bir araya getirilmesi ile elde çok katmanlı yapı ağın daha soyut özellikleri algılmasını, daha karmaşık yapıları keşfetmesini sağlamaktadır. Bu sayede elde edilen yapı CNN'in resim sınıflandırma [132-135], yüz ifadesi tanıma [136] ve sahne etiketleme [137] gibi birçok uygulamada başarılı olmasını sağlamaktadır.




Şekil 4.16. CNN'nin metin sınıflandırılmasında kullanımı.

CNN modelleri resim ile ilgili uygulamaların yanı sıra ses sınıflandırması [138] ve metin sınıflandırması [34] uygulamalarında da kullanılabilmektedir. Şekil 4.16'te de görüldüğü gibi CNN'ler metin sınıflandırması işleminde örnek bir kullanımı görülmektedir. Burada tweet içerisindeki her bir kelime bir satırı oluştururken, bu kelimelerin her biri de kelime temsilleri ile elde vektörel gösterimlerden meydana gelmektedir. CNN'lerin önemli kısıtlamalarından bir tanesi sabit boyutlu girdilere ihtiyaç duymalarıdır. Bu noktada kelime temsilleri halihazırda sabit boyutlu olduğu

için problem oluşturmamaktadır. Öte yandan her bir tweet farklı sayıda kelimeye sahip olabilir. Bu nedenle her bir tweet için sabit bir uzunluk değeri belirlenmekte ve tweet bu uzunluktan daha az sayıda kelimeye sahip ise 0 vektörleri ile temsil edilmektedir.

# BÖLÜM 5

# DERLEMLER VE VERİ SETLERİ

Bu bölümde ilk olarak kullanılan derlemler ve veri seti hakkında bilgi verilmektedir. Sonrasında ise kıyaslama metodolojisi açıklanarak, son olarak da elde edilen sonuçların değerlendirmesini yapılmaktadır.

## 5.1. TRAFİK VERİ SETLERİ

Günümüzde özellikle metropollerde karayolu trafiğinin yönetilmesi başlı başına bir problem haline gelmiştir. Bölüm 1.1'de belirtildiği gibi trafik sıkışıklıklarının nerdeyse yarısı kaza, hava durumu veya yol çalışması gibi özel bir olaydan kaynaklı tekrarlamayan sıkışıklıklardır. Bu sıkışıklıkların kaynağının hızlı ve maliyet etkin bir çözümle tespit edilebilmesi trafik otoritelerine büyük kolaylıklar sağlayacaktır. Son yıllarda mobil cihaz kullanımının yaygınlaşmasıyla birlikte trafikle ilişkili olaylarda sosyal medya platformları üzerinden paylaşılır hale gelmiştir. Bu durumun bir neticesi olarak bu platformlar üzerinden paylaşılan mesajlar tarfik olaylarının tespit edilebilmesi için ucuz ve etkin bir çözüm olabilir. Öte yandan SMM'lerdeki yüksek miktardaki gürültülü metinler bu platformlardan bilgi edilmesini kısıtlayan en büyük etkenlerden biridir. Literatürde gürültüyle başa çıkmak için en yaygın olarak kullanılan yöntemlerden bir tanesi metin normalizasyonudur. Fakat bu yöntem Türkçe gibi zengin morfolojik yapıya sahip diller için oldukça zorludur. Bununla birlikte son yıllarda öne çıkan yaklaşımlardan bir tanesi de alana özel kelime temsillerinin kullanılmasıdır. Bu tez çalışması kapsamında normalizasyon işleminin ve etiketsiz verilerden elde edilen alana özel kelime temsillerinin trafik verilerini sınıflandırma başarımı üzerine olan etkileri değerlendirilmektedir.

Bu tez çalışmasında trafikle ilgili tweetlerin sınıflandırılmasında iki farklı yaklaşım kullanılmaktadır. Bunlardan birincisinde problem ikili sınıflandırma problemi olarak

ele alınmaktadır ve tweetler trafikle ilgili olanlar ve trafikle ilgili olmayanlar şeklinde iki farklı sınıfa ayrılmaktadır. Bir diğer yaklaşımda da konu çok sınıflı sınıflandırma problemi olarak ele alınmaktadır. İkinci grup yaklaşımda sekiz farklı başlık altında sınıflandırma yapılarak araç arızası, hava koşulları ve trafik kazası gibi özel durumlarından belirlenmesi hedeflenmiştir.

Çizelge 5.1. Trafikle ilgili tweetlerin elde edilmesinde kullanılan anahtar kelimeler.

| | | |
|---|---|---|
| sürücü | kaza | yol |
| otoban | otoyol | kavşak |
| köprü | arıza | tünel |
| trafik | araç | şerit |
| yoğunluk | araba | çalışma |
| yanyol | konvoy | hasarlı |
| kuyruk | kaldırım | gişe |

Veri setinin oluşturulması aşamasında Twitter REST API kullanılmıştır ve anahtar kelimeye dayanan bir sorgulama işlemi gerçekleştirilmiştir. Anahtar kelimeler trafikle ilgili haberlerde sıklıkla geçen sözcüklerden oluşturulmuştur. Çizelge 5.1'de sorgulamalarda kullanılan anahtar kelimelerin listesi bulunmaktadır. Toplam 21 kelimenin tamamını içeren bir veya sorgusu kullanılmıştır. Dolayısıyla veri setindeki her bir tweet bu kelimelerden en az bir tanesini içermektedir. Aşağıdaki örneklerde trafikle ilgili olan tweetler ve trafikle ilgili olmayan tweetlerin bazıları görülmektedir.

Trafikle ilgili olmayan tweetler:

- Aşk dediğin buca ' nın yol çalışması gibi olmalı hiç bitmemeli.
- Ay'a Dört şeritli yol çalışması ilk denemede başarısızlıkla sonuçlandı!
- İlk yapılan yanlışa kaza, İkincisine hata, Üçüncüsüne ise tercih denir. Dostoyevski
- Tenha bir caddenin trafik lambası gibi hissediyorum bazen kendimi, kırmızılarıma duran yok, yeşillerimi gören.

Trafikle ilgili olan tweetler:

- #O3 #atisalani çıkışı #bayrampasa yönü sağ seritteki #**aracarizasi** #trafikyogunlugu nu artırıyor.#yolyardim #cekici #istanbul @radyotrafik
- @muratkazanasmaz Bogazici Koprusu Anadolu-Avrupa yonu, kopru cikisindaki Besiktas sapaginin ilerisinde arac arizasi, hem de orta seritte
- İstanbul'da sağanak yağış sonucu #Ataşehir ve #Kadıköy'de yollar göle döndü trafik felç oldu http://bit.ly/1p8BQAi
- Tem camlica ayrimi öncesi 2 ayri arac serit ihlali ve tayiz 34 GL 4xx8 kamyonet ve 16 jxx 70 kamyon @radyotrafik

İkili sınıflandırma görevinde kullanılan veri seti 1 000 tanesi trafikle ilgili 1 000 tanesi de trafikle ilgili olmayan olmak üzere toplam 2 000 adet tweetten oluşan dengeli bir veri setidir.

Öte yandan çoklu sınıflandırma görevi bir tanesi trafikle ilgili olmayan tweetler olmak üzere sekiz farklı gruptan oluşmaktadır. Trafikle ilgili olan tweetler ise yedi ayrı alt gruba ayrılarak sınıflandırılmıştır. Bu gruplar şunlardır:

- Yol Yapım Bakım Onarım Çalışmaları
- Trafiğe Neden Olan Harici Olaylar
- Hava Durumu
- Araç Arızası
- Sürücü Hatası
- Kaza
- Genel

Yol yapım bakım onarım çalışmaları grubu altındaki sınıflandırma işleminde planlı plansız yol çalışmaları ile ilişkili olarak bildirilen tweetler dahil edilmiştir. Trafiğe neden harici olaylar başlığında ise miting, konser ve maç gibi harici olaylarla ilişkili trafik tweetleri ilave edilmiştir. Hava durumu grubu ise sis, yağmur ve kar yağışı gibi meteorolojik koşulları içeren trafik tweetlerini içermektedir. Araç arızası başlığında ise kullanıcılar tarafından bildirilen araç arızalarını ilgilendiren tweetler

bulunmaktadır. Sürücü hatası grubunda şerit ihlali, hatalı park ve hız ihlali gibi konularda paylaşılmış tweetler yer almaktadır. Kaza grubu tweetleri ise trafik kazaları ile ilişkili tweetleri içermektedir. Son olarak genel sınıflandırmasında bulunan tweetler ise özel bir duruma işaret etmeyen genel olarak trafik sıkışıklığı ilgili paylaşımları içeren tweetler bulunmaktadır. Bu veri setinde bulunan trafikle ilgili tweetler ikili sınıflandırma probleminde kullanılan tweetlerin aynısıdır. Olayların daha ayrıntılı olarak tespit edilebilmesi için alt sınıflara ayrılmış halinden oluşmaktadır. Öte yandan bu veri setine trafikle ilgili olmayan tweetlerden sadece 200 tanesi dahil edilmiştir. Bunun nedeni sınıflar arasında aşırı şekilde dengesiz bir veri oluşmasını önlemektir. Çizelge 5.2'de kullanılan her bir sınıftan kaç adet tweet bulunduğu görülmektedir. Ayrıca veri seti her bir sınıftan en az 100 adet tweet olacak şekilde oluşturulmuştur.

Çizelge 5.2. Çok sınıflı trafik veri setinin sınıf bazında tweet sayısı dağılımı.

| **Veri Sınıfı** | **Tweet Sayısı** |
|---|---|
| Yol Çalışması | 100 |
| Harici Olay | 100 |
| Hava Durumu | 100 |
| Araç Arızası | 280 |
| Sürücü Hatası | 100 |
| Kaza | 127 |
| Genel | 193 |
| Trafikle İlgili Olmayan | 200 |

## 5.2. DERLEMLER

Kelimelerin vektörel temsillerinin elde edebilmek için derlemlerden yararlanılmaktadır. Derlem kavramı Türk Dil Kurumu tarafından Genel Türkçe Sözlük'te şu şekilde tanımlanmıştır:

"*Bir dilin türlü kullanım alanlarından derlenmiş örneklerinin dil bilgisi ve kuramsal dil bilimi araştırmalarında kullanılmak üzere bilgisayar tarafından okunabilecek biçimde bir araya getirilmiş kümesi.*"

Word2vec yöntemi derlemi girdi olarak kullanarak kelimelerin vektörel temsillerini elde etmektedir. Bu doğrultuda kelimelerin derlem içerisindeki bir arada

bulunabilirlik durumlarına göre vektörel benzerlikleri artmakta veya azalmaktadır. Dolayısıyla doğru bir temsil kabiliyetinin sağlanabilmesi açısından derlem oldukça ön plana çıkmaktadır.

Trafik verilerinin sınıflandırılması işleminde kelime temsilleri öncelikli olarak veri setindeki etiketli tweetlerden elde edilmektedir. Bunun yanı sıra alana özel kelime temsillerinin etkisini değerlendirmek için trafik alanınıyla ilgili tweetler toplanarak bir derlem oluşturulmuştur. Trafik alanına özel derlem oluşturuluruken trafik veri setine benzer bir yöntem takip edilmiştir ve Çizelge 5.2'de bulunan anahtar kelimeleri içeren tweetler toplanmıştır. Bunun yanı sıra "@ibbtrafikradyo", "@radyotrafik" ve "@radyotrafik06" gibi doğrudan trafikle ilgili Twitter hesaplarındanda anahtar kelime olmadan veri kazıma işlemi yapılmıştır. Veri kazıma işleminde trafik veri setinden farklı olarak Twitter API'leri yerine Python dilinde geliştirilmiş olan Scrapy veri kazıma aracı kullanılmıştır. Sonuç olarak trafikli ilgili anahtar kelimeleri içeren veya doğrudan trafikle ilgili hesapların paylaştığı toplam 1,5 M tweet elde edilmiştir. Elde edilen trafik alanına özel tweetler Bölüm 7.1'de anlatılan ön işleme adımlarına tabi tutulduktan sonra Word2vec CBOW yöntemi ile kullanılarak kelimelerin vektörel temsilleri elde edilmiştir. Ön işlem adımlarında işlemleri esnasında tüm Uniform Resource Locator (URL) ifadeleri url şeklinde metin ifadelerine dönüştürülmüştür. Bunun haricinde tüm metinler küçük harfe dönüştürüldükten sonra noktalama işaretlerinin tamamı temizlenmiştir. Son olarak ise sayısal karakter içeren ifadeler çıkartılmıştır.

Ayrıca normalizasyon işlemlerindeki belirsizlik durumlarını gidermek için biri İngilizce bir diğeri de Türkçe olmak üzere 2 farklı derlem daha kullanılmaktadır. İngilizce ve Türkçe derlemlerin kelime seviyesinde unigram modelleri kullanılarak normalizasyon işlemi esnasında kelimenin Türkçe veya İngilizce olup olmadığının değerlendirilmesi yapılmaktadır. Buna ilave olarak kullanılan Türkçe derlemden kelime temsilleri elde edilerek Türkçe kelimelerin normalizasyonu esnasında oluşan belirsizlik durumları giderilmektedir. Buradaki Türkçe derlem trafik derleminden farklı olarak alana özel değil genel bir derlemdir.

Genel Türkçe derlem oluşturmak için Hürriyet gazetesi arşivlerinden oluşturulan [139] derlem ve [58]'deki çalışmada da kullanılan Turkish Wikipedia Dump (trwiki-20150121-pages-meta-current) veri seti derlem olarak kullanıldı. Bunlara ilave olarak oluşturulan derlemin temsil gücünü artırabilmek adına bazı e-kitaplar da derleme dahil edildi. Derlem oluşturulurken tüm noktalama işaretleri ve sayısal değerler temizlendi. Sonuç olarak 1 249 914 farklı kelimeden oluşan ve toplamda 200 822 716 kelime içeren bir derlem oluşturuldu. Derlemde "üzengilerin", "bostanların" gibi en az görülen kelimeler sadece bir defa tekrar etmiştir. Öte yandan en fazla bulunan "bir" kelimesi 3 503 469 kez tekrar etmiştir. Ayrıca derlemde bulunan yabancı kelimeleri ve yanlış yazımları eleyebilmek adına tüm derlem morfolojik analizden geçirildi. Morfolojik analiz işlemi için Zemberek NLP kütüphanesinin [54] morfolojik analiz aracını ve bu tez çalışmasında kullanılan morfolojik analiz aracı kullanıldı ve yalnızca her iki analiz aracının da geçerli Türkçe kelime olarak kabul ettiği kelimeler ikinci derleme dahil edildi. Nihai olarak 185,5 M kelimeden oluşan Türkçe bir derlem elde edilmiş oldu.

Yukarıdaki bölümlerde de belirtildiği gibi Türkçe'nin zengin ek yapısı çok fazla sayıda yeni kelime oluşturulmasına olanak sağlamaktadır. Eklerin peşi sıra gelerek oluşturduğu kelimeler çok uzun yüzey formlarına sahip olabilmektedir. Bunun örneklerinden bir tanesi oluşturulan Türkçe derlemdeki "muvaffakiyetsizleştiricileştiriveremeyebileceklerimizdenmişçesine" kelimesidir. Toplam 65 karakterden oluşan sözcük Türkçe'nin geniş kelime hazinesini ortaya koymaktadır. Ayrıca kelime yazımının doğruluğunu denetlemek için kullanılan sözlük tabanlı yaklaşımların kısıtlamalarını da göstermektedir. Bununla birlikte morfolojik analiz araçlarının da birtakım dezavantajları bulunmaktadır. Bunlardan bir tanesi bazı durumlarda günlük hayatta kullanılmayan kelimelerin de doğru kök ve ek dizilimine sahip geçerli kelimeler olabilmesidir. Örneğin "şilence" kelimesi günlük hayatta kullanılmazken geçerli bir morfolojik çıktıya sahiptir. Bu durumda normalizasyon işlemini zorlaştırmaktadır. Şöyle ki İngilizce "silence" kelimesine diyakritik restorasyon işlemi uygulandığında geçerli morfolojik forma sahip "şilence" kelimesi elde edilmektedir. Bu doğrultuda bu tip belirsizliklerin önüne geçebilmek adına normalizasyon işlemi sonrasında olasılık tabanlı bir belirsizlik giderme işlemi uygulanmaktadır. Bunun için hazırlanan Türkçe derlemden ve İngilizce derlemden

yararlanılmaktadır. Türkçe sosyal medya metinlerinde en fazla rastlanılan yabancı dilin İngilizce olması nedeniyle bu çalışmada sadece İngilizce'ye odaklanılmıştır.

Bu tez çalışmasında oluşturulan İngilizce derlem için Türkçe'ye benzer şekilde Wikipedia'dan dosyalarından yararlanıldı. İngilizce üzerine yapılan çalışmaların Türkçe'ye kıyasla çok daha fazla olması nedeniyle İngilizce için çok büyük boyutlu derlemler elde mümkündür. Fakat Türkçe ile benzer kelime sayıları olması için sadece "enwiki-20180120-pages-meta-current14.xml-p6197598p7697598" dosyası kullanılarak yaklaşık 204 M kelimeden oluşan bir derlem oluşturulmuştur.

# BÖLÜM 6

# SOSYAL MEDYA MESAJLARININ NORMALİZASYONU

## 6.1. ÖNERİLEN NORMALİZAYON YAKLAŞIMI

Bu bölümde trafikle ilgili Twitter mesajlarının normalizasyonunda kullanılan yaklaşım tanıtılmaktadır. Türkçe'nin zengin morfolojik yapısı nedeniyle birçok durumda bir OOV token için birden fazla aday kelime üretilebilmektedir. Oluşan belirsizlik durumunun Word2vec tabanlı bir yaklaşım kullanılarak giderilmesi hedeflenmektedir. Bu doğrultuda yapılan çalışmalarda normalizasyon işlemi için kullanılan diyakritik, aksan ve ünlü harf restorasyon modülleri ile yazım denetimi modülü Word2vec tabanlı belirsizlik giderme modülü ile genişletilmiştir.

MRL kavramı söz dizimsel birimler ve ilişkiler arasındaki önemli bilgilerin kelime düzeyinde ifade edildiği diller olarak tanımlanmıştır [140]. Daha önceki bölümlerde de belirtildiği gibi Türkçe MRL'lerin güçlü bir temsilcisidir. MRL için karşılaşılan problemler daha önceki çalışmalarda hızlı sözlük büyümesi, zayıf dil modeli ve olasılık tahmini, çok yüksek OOV oranları, çekimli yapısı ve makine çevirisi için bileşik sözcükler olmak üzere 4 temel başlık altında ele alınmıştır [12,141]. Ayrıca Türkçe'nin diyakritik karakterler içermesi, birçok OOV kelime için birden fazla aday kelimenin bulunması gibi faktörler ilave birtakım zorluklar getirmektedir. Bu nedenle sözlükte arama, salt istatistiksel modeller veya benzerlik fonksiyonları gibi birçok geleneksel metot Türkçe SMM normalizasyon probleminin çözümü için tek başına yeterli değildir.

Bu çalışmada Türkçe SMM normalizasyon probleminin çözümü için paralel ve kaskad mimariden oluşan hibrit bir yaklaşım kullanılmıştır. İlk aşamada gelişmiş bir tokenization işlemi uygulanarak Twitter'a ait "@", "#" gibi özel işaretlerin bulunduğu kelimeleri ayrılmaktadır. Sonrasında [12]'deki çalışmaya benzer şekilde bir morfolojik

analizör kullanarak OOV kelimeleri tespit edilir ve aday kelime üretme aşamasına gönderilir. 10 farklı modülden oluşan aday kelime üretme aşamasında her bir OOV kelime için olası aday kelimeler üretilir. Bu aşamada aynı anda aday kelime üretme ihtimaline sahip birimler paralel olarak çalışırken diğer birimler kaskad olarak çalışır. Bir örnekle açıklamak gerekirse "gun" formunda yazılmış bir OOV kelime için çok dilli tweet tespit modülü bu kelimeyi İngilizce bir kelime olarak tespit ederken DR birimi bu kelime için "gün" aday kelimesini üretmektedir. Bu durum belirsizliğe neden olmaktadır ve bu birimlerin kaskad çalışması durumunda her zaman üst katmandaki sonuç doğru olarak kabul edileceği için hatalı kelime üretme ihtimali oluşacaktır. Bu nedenle bu tip birimler paralel çalıştırılarak her iki modülünde aday kelime üretmesi sağlanmakta sonrasında belirsizlik giderici ile en uygun kelime seçilmektedir. Benzer şekilde "gol" - "göl" ve Çizelge 3'teki örneklerde olduğu gibi herhangi bir modülünde birden fazla kelime üretmesi durumunda da yine belirsizlik gidericiye gönderilmekte ve en uygun kelime seçilmektedir. Her bir bölümün detaylı anlatımı aşağıdaki alt bölümlerde mevcuttur.

### 6.1.1. Tokenization & OOV Tespiti

Yukarıda ki bölümlerde de belirtildiği gibi MRL ve eklemeli yapısı nedeniyle Türkçe gibi diller için sözlük dışı kelimeleri (OOV'leri) bir sözlüğe bakarak tespit etmek yüksek boyut ve zaman gereksinimleri nedeniyle verimli bir yöntem değildir. Bir morfolojik analizör kullanılarak bu problemin üstesinden gelinebilir. Morfolojik analizör köklerin ve eklerin bulunduğu sözlükleri ve dile ilişkin kuralları alarak verilen yüzey formunu analiz edip olası kök ve ek dizilimlerini ortaya çıkarır. Eğer girdi olarak uygulanan yüzey formu için bir çıktı üretilebilmişse bu durum ilgili yüzey formunun geçerli bir sözcük olduğunu gösterir. Tam tersi durumda yani geçerli bir çıktı üretilememişse bu kelime OOV olarak kabul edilir ve aday üretme aşamasına gönderilir. Şekil 6.1'de morfolojik analiz işleminde kullanılan kök bulma sürecine ait kod parçası görülmektedir. Burada analiz edilecek parçanın ilk iki karakterinden başlayarak kök tablosunda bulunup bulunmadığına bakılmaktadır. Sonuç kök tablosunda mevcut ise isim veya fiil gibi kökün türünün ne olduğuyla birlikte sonuç döndürülmektedir. Bu noktada her bir kök birden fazla olası çözüme sahip olabilmektedir. Bu durumla ilgili olarak "yüz" kökü incelenecek olursa, kelime kök

tablosunda bir sayıyı ifade eden isim formuyla bulunduğu gibi "yüzmek" kelimesini ifade eden fiil şekliyle de bulunmaktadır. Bu durumda "yüz" kelimesi için iki ayrı kök sonucu elde edilecektir. Dolayısıyla kelimenin geri kalan kısmındaki ekler bu köklerden herhangi birine veya ikisine de uygun olup olmadıklarına göre değerlendirilerek analiz işlemi tamamlanacaktır.

Sosyal medya metinlerinde büyük küçük harf kurallarına genellikle riayet edilmemektedir. Bu nedenle normalizasyon aracı büyük ve küçük harflere duyarsız olacak şekilde geliştirilmiştir. Örneğin büyük harflerle yazılması gereken "TBMM (Türkiye Büyük Millet Meclisi)" ifadesi için "tbmm" şeklinde yapılan yazımda morfolojik analiz aşamasında doğru kabul edilerek aday kelime üretme sürecine gönderilmemektedir.

```
private IList<Tuple<string,string>> searchRoots(string word)
{

    List<Tuple<string,string>> couple = new List<Tuple<string,string>>();
    for (var i = 0; i < word.Length; i++)
    {
        string possibleRoot = word.Substring(0, i + 1);

        List<string> roots = searchRootTable(possibleRoot);

        foreach (var root in roots)
        {

            couple.Add(new Tuple<string, string>(possibleRoot, root));
        }
    }
    return couple;
}
```

Şekil 6.1. Morfolojik analiz aracının kök arama süreci kod parçası.

Öte yandan Twitter "#" ve "@" gibi birtakım özel işaretlerin yanı sıra URL'ler ve emojiler gibi birçok yazım türünü de bünyesinde barındırır. Ayrıca tweetlerin içerisinde normal kelimelerin bir arada zaman ifadeleri (tarih, saat), sayılar ve e-posta adresleri gibi ifadelerde bulunmaktadır. Bu tip yazımlar OOV tespit adımına göndermeden tokenization işlemi esnasında etiketlenip ayrılmaktadır. Bu işlem için düzenli ifadelerden (Regular Expressions) yararlanılmaktadır. Tokenization işlemi esnasında toplam 12 farklı düzenli ifadeden yararlanılmaktadır. Aşağıdaki iki örnekte sırasıyla URL ve Twitter'a özel karakterler ile başlayan tokenları tespit etmek için

kullanılan düzenli ifadeler görülmektedir. Burada URL ifadeleri için hem kısa hem normal URL'lerin tespit edilmesi hedeflenmiştir. Ayrıca Twitter'a özel ifadelerde de mention ve hashtag'lerin tespit edilmesi amaçlanmıştır.

- public const string URL = @"(http://|https://)([a-zA-Z0-9]+\.[a-zA-Z0-9\-]+|[a-zA-Z0-9\-]+)\.[a-zA-Z\.]{2,6}(/[a-zA-Z0-9\.\?=/#%&\+-]+|/|)";

- public const string TwitterSpecial = @"(@|#)\w+";

Çizelge 6.1. Tokenization & OOV Tespiti.

| Tokenization | | OOV Tespiti | |
|---|---|---|---|
| Token | Etiket | Token | Etiket |
| #DünyaGülümsemeGünü | Hashtag | hergun | CG |
| hergun | OOV-D | bir | Correct |
| bir | OOV-D | oncekinden | CG |
| oncekinden | OOV-D | daha | Correct |
| daha | OOV-D | cok | CG |
| cok | OOV-D | gulmek | CG |
| gulmek | OOV-D | | |
| https://t.co/1hpfMmMNi | URL | | |

Çizelge 6.1'deki örnekte de görüldüğü gibi:

- #DünyaGülümsemeGünü hergun bir oncekinden daha cok gulmek https://t.co/1hpfMmMNi

şeklinde yazılmış bir tweet için öncelikli olarak tokenization işlemi uygulanarak hashtag ve URL etiketlenir sonrasında kalan tokenlar OOV tespit işlemine tabi tutulur ve OOV olarak tespit edilenler aday kelime üretme adımına gönderilir. Tüm normalizasyon işleminden sonra beklenen çıktı:

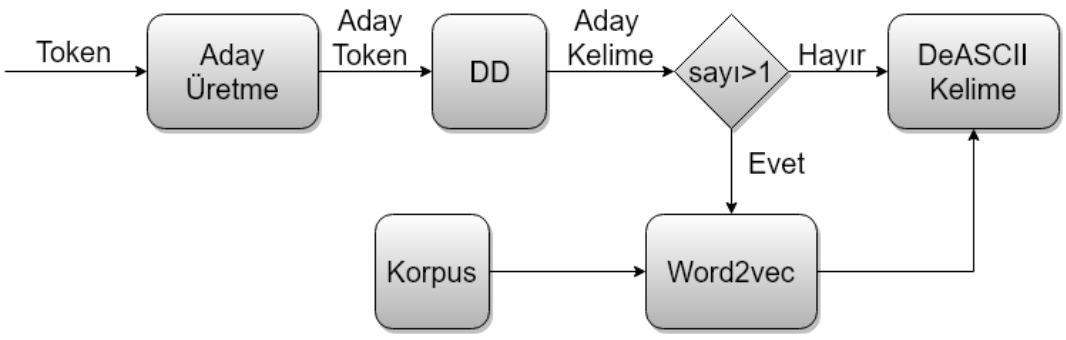
- "twitterözel[#DünyaGülümsemeGünü] her gün bir öncekinden daha çok gülmek url[https://t.co/1hpfMmMNi]" şeklindedir.

### 6.1.2. Diyakritik Restorasyon

Diyakritik karakter problemi Türkçe sosyal medya metinlerinde en fazla karşılaşılan problemlerden bir tanesidir. Trafik veri setindeki OOV kelimelerin neredeyse yarısına yakın bir kısmı diyakritik karakter problemi içermektedir. Bunun başlıca sebebi diyakritik karakterlerin birçok mobil cihazda ikincil karakter olması ve kullanıcı alışkanlıkları nedeniyle bu karakterler yerine ASCII eşleniklerinin kullanılmasıdır.

DR modülünde aday kelime üretme işlemi iki aşamalı olarak gerçekleştirilmektedir. Bu işlemin birinci aşamasında [56]'daki çalışmaya benzer şekilde karar listesi kullanılmaktadır ve bu işlemin sonucunda bir çözüm üretilmişse bu çözümler dil doğrulama (DD) katmanına gönderilmektedir. Eğer hiç aday kelime üretilememişse kural tabanlı bir yaklaşım uygulayarak olası tüm çözümler elde edip DD katmanına gönderilmektedir. DD katmanı üretilen yüzey formunun morfolojik analizini yaparak sadece geçerli Türkçe kelimeleri seçmektedir. Morfolojik analiz sonrasında herhangi bir çözüm üretilebilmişse bu sözcüğün geçerli bir Türkçe sözcük olduğunu ifade eder. Aksi takdirde sözcük geçerli olmadığından listeden çıkarılır.

Yukarıdaki bölümlerde de belirtildiği gibi DR işleminden sonra birden fazla aday kelime oluşabilmektedir. Şekil 6.2'de de görüldüğü gibi birden fazla aday kelimenin oluşması durumunda Word2vec tabanlı bir belirsizlik giderme işlemi uygulanmaktadır.




Şekil 6.2. Önerilen DR yaklaşımı.

### 6.1.3. Aksan Normalizasyonu

SMM'ler bünyesinde çok çeşitli hatalı yazım stilleri barındırmaktadır. Bu hatalı yazım stillerinden bir tanesi "wuz up bro (what is up brother)" örneğinde olduğu gibi fonetik heceleme veya yerel aksan kullanılmasıdır [142]. Fonetik heceleme problemi Tekrar Eden Harfler & Sözlükte Arama başlığı altında ele alınmaktadır ve bu bölümde Türkçe SMM'de yaygın olarak kullanılan aksan problemlerine odaklanılmaktadır. Türkçe genel itibariyle yazıldığı gibi telaffuz edilen bir dil olmakla birlikte SMM içerisinde genellikle "gelcem – geleceğim", "geliyom – geliyorum" gibi kelimelerin eklerin kısaltılarak yazılması şeklinde yaygın bir aksanlı kullanım vardır.

Öte yandan buradaki en büyük problem kullanılan olumsuzluk ekleri, soru ekleri, zaman ekleri ve şahıs ekleri gibi faktörlerin de dahil edilmesiyle olası yüzey formu çeşitliliğinin artması ve ilave olarak [12]'deki çalışmada da belirtildiği gibi aksan normalizasyonunda uyulması gereken bir dizi ünlü-ünsüz uyumu kurallarının bulunmasıdır. Örneğin doğru yazımı "gelmiyor musun" şeklinde verilen kelime doğru yazım formunda 2 token ile ifade edilirken hatalı yazım formlarında "gelmıyonmu" veya "gelmiyonmu" tek token ile ve farklı yüzey formları ile kullanım söz konusudur. Benzer şekilde "gelecek misin" ifadesi için "gelcenmi" veya "gelicenmi" gibi kullanımlar söz konusu iken "gelmeyecek misin" örneği için "gelmicenmi", "gelmiycenmi" veya "gelmeyecenmi" şeklinde kullanımlar mevcuttur.

Çizelge 6.2. Bazı aksanlı yüzeyler ve aday formları.

| Aksanlı Ek | Aday Ekler |
|---|---|
| _icem | _eceğim, _eyeceğim, _iyeceğim |
| _iyom | _ıyorum, _iyorum |
| _iyon | _ıyorsun, _iyorsun |
| _iyo | _iyor, _ıyor |
| _mıcağınıza | _mayacağınıza |
| _maycam | _mayacağım |

Bu problemin çözümü için öncelikli olarak normalize edilecek tokenın Çizelge 6.2'de örnekleri görülen aksan formları ile bitip bitmediğini kontrol edilmektedir. Ayrıca bu işlem için kullanılan sözlükte olası aday yüzey formlarını da bulundurulmaktadır. Eğer sözcük aksanlı bir sona sahipse aksanlı kısmı sözcükteki aday kısımların her biri ile değiştirilip dil doğrulayıcı (DD) ile morfolojik analiz yaparak geçerli bir yazım olup olmadığı kontrol edilmektedir. Eğer geçerli bir kelime elde edilmişse normalizasyon işlemini tamamlanır, aksi taktirde orijinal girdi ile normalizasyon adımlarına devam eder.

### 6.1.4. Tekrar Eden Harfler & Sözlükte Arama

Bu bölümde Çizelge 6.3'te örnekleri görülen fonetik yazım, jargonlar, kısaltmalar, argo kelimeler, vokatif ve karakter tekrarları gibi yazım hatalarının düzeltilmesine odaklanılmaktadır.

Çizelge 6.3. Türkçe SMM'lerdeki bazı OOV örnekleri.

| Örnekler | |
|---|---|
| Kib (Kendine iyi bak - Take Care) | Taym (Time) |
| Pls (Please) | Yutuber (Youtuber) |
| Unf (Unfollow) | Menşın (Mention) |
| Fav (Adding Favorite) | Onlayn (Online) |
| Insta (Instagram) | Hahahahah (Vocative) |
| Tivit (Tweet) | |

Karakter tekrarları için öncelikli olarak tokenın "seviyorummmmm – seviyorum" örneğindeki gibi 2'den fazla peş peşe tekrar karakteri olup olmadığını kontrol edilir. Eğer varsa doğrudan tek karakterle değiştirilir. "Seviyyorum" şeklinde sadece iki tane tekrar eden karakter varsa tek karakterle değiştirilip DD ile geçerli sözcük olup olmadığını kontrol edilmektedir. Eğer geçerli bir sözcük ise normalizasyon işlemi tamamlanır aksi taktirde orijinal girdi ile işleme devam edilir.

Yukarıda da belirtildiği gibi Türkçe geneli itibariyle yazımı ve telaffuzu aynı olan bir dil olmak birlikte SMM'lerde "taym – time", "feys – Facebook", "vatsap – WhatsApp"

veya “yutuber – YouTuber” gibi özellikle yabancı kelimelerin yazımında fonetik yazım ile karşılaşılmaktadır. Buradaki temel problem bu tip yazılara Türkçe ekler getirilerek “yutuberlar”, “yutuberın”, “feysde” gibi değişik yüzey formlarının elde edilmesidir. Benzer şekilde doğrudan yabancı kelimelere Türkçe ekler getirilerek kullanımlar da mevcuttur. Örneğin “stalk” kelimesine Türkçe ekler getirilerek “stalkladım” biçiminde kullanımlar da bulunmaktadır. Bir diğer kullanım biçimi de “slm – selam” tarzında kısaltmaların veya “unf – unfollow” gibi sosyal medyaya has jargonların kullanılmasıdır.

Bu hataların düzeltilebilmesi için öncelikli olarak yaygın kullanılan fonetik yazım, jargon ve kısaltmaları tespit edildi. Sonrasında “slm” şeklindeki kullanımlar için tokenın tamamının eşleşip eşleşmediğini kontrol ederek sözlükteki karşılığı ile değiştirildi. Öte yandan Türkçe ekler ile uzatılabilen yüzey formları için tokenın baş kısmının eşleşip eşleşmediği kontrol edilerek kelimeleri türlerine göre jargon, argo (örn: jargon[stalkladım] ) formunda etiketlendi.

Vokatiflerde SMM metinlerinde karşılaşılan problemlerden bir tanesi ve Türkçe için “hehehe” ve “hııı” gibi çeşitli kullanımları bulunmaktadır. Bu tip yazımlar da düzenli ifade paternleri kullanılarak tespit edilip vokatif olarak etiketlendi.

Ayrıca [12]’daki çalışmada örnekleri verilen “$-ş”, “@-a” logogramların ve “w” ile “q” gibi Türk alfabesinde olmayan harfleri benzerleri olan “v” - “k” gibi harfler ile değiştirilmektedir. Fakat diğer değişim işlemlerinden farklı olarak bu işlem çok dilli tweet tespit adımından sonra gerçekleştirilmektedir.

Bu bölümde ayrıca trafik alanına özel olarak yaygın kullanılan kısaltmalar ve hatalı yazım stilleri de düzeltilmektedir. Genel kullanımda çok fazla karşılaşılmayan fakat SMM’lerde trafik alanına özgü metinlerde sıklıkla karşılaşılan “yanyol (yan yol)” ve “FSM (Fatih Sultan Mehmet) köprüsü” gibi yaygın kullanımlarda bu bölümde düzeltilmektedir.

### 6.1.5. Çok Dilli Tweet Tespiti

Çok dilli metinler SMM içeriklerinde sıklıkla karşılaşılan problemlerden bir tanesidir. Türkçe SMM'lerde en yaygın olarak kullanılan yabancı dil İngilizce olduğu için bu bölümde sadece İngilizce kelimeleri hedef alan bir çalışma yapılmaktadır.

Bu modül İngilizce kelimeleri bir sözlüğe bakarak tespit eder. Fakat aynı anda birden fazla modülün aday sonuç üretme ihtimali bulunabilir. Aşağıdaki örnekte de görüldüğü gibi "gun" tokenı için iki olası çözüm vardır. Birincisi bu kelimenin İngilizce bir sözcük olması bir diğeri de DR işlemi sonrasında önerilen "gün" kelimesidir. Benzer şekilde "one" girdisi için DR modülü "öne" aday kelimesini üretecektir. Bu nedenle aynı anda çözüm üretme ihtimaline sahip bu modülleri kaskad yerine paralel olarak çalıştırılmaktadır.

Öte yandan oluşan bu belirsizliği gidermek için yukarıdaki bölümlerde detayları verilen İngilizce ve Türkçe olmak üzere 2 farklı derlem kullanarak kelimelerin sözlükte geçme frekanslarına bakılmaktadır ve Eşitlik (6.1)'in sonucuna göre aday kelimeyi seçilmektedir. Kelimelerin frekanslarının kontrol edilmesinin temel sebebi DR işlemi sonrasında elde edilen sözcüklerden bazılarının morfolojik olarak geçerli olmakla birlikte günlük hayatta çok nadiren kullanılan veya hiç kullanılmayan kelimeler olmasıdır. Benzer şekilde İngilizce'de nadir görülen kelimelerin olasılıklarının da düşük olması sağlanılmaktadır. Ayrıca tweet içerisinde başka bir İngilizce kelime varsa bu da eşitliğe dahil edilerek kelimenin Türkçe mi yoksa İngilizce mi olduğuna karar verilmektedir. Burada $P_t(word)$ kelimenin Türkçe olma olasılığı ve $P_e(word)$ kelimenin İngilizce olma olasılığı, $k$ tweet içerisindeki toplam İngilizce kelime sayısı, $corpus_t$ Türkçe derlemdeki toplam kelime sayısı, $corpus_e$ İngilizce derlemdeki toplam kelime sayısı, $count_t(word)$ kelimenin Türkçe derlemde bulunma sayısı, $count_e(word)$ kelimenin İngilizce derlemde bulunma sayısı, $t$ *ve* $e$ sabitler olmak üzere $(t>e)$ kelimelerin Türkçe veya İngilizce olma olasılıkları şöyle hesaplanır.

$$P_t(word) = t\frac{count_t(word)}{corpus_t} \tag{6.1}$$

$$P_e(word) = (e+k)\frac{count_e(word)}{corpus_e} \tag{6.2}$$

Burada olasılık değeri yüksek çıkan duruma göre kelimenin Türkçe veya İngilizce olduğuna karar verilir.

### 6.1.6. Rastgele Harflerle Gülme

Son yıllarda sosyal medyada en sık görülen yazım şekillerinden bir tanesi de Örnek 13'te de çeşitli numuneleri görüldüğü gibi rastgele harfler kullanarak gülmektir. Bu yazım biçiminin algılanması özellikle tweetlerin duygusal analizi için oldukça önemlidir. Öte yandan uzunluk ve kullanılan harfler açısından çok çeşitli formatta olmaları algılanmalarını zorlaştırmaktadır. İlave olarak normalize edilmeye çalışılan diğer yazım hataları nedeniyle de istatiksel olarak tespit edilmesi de güçleşmektedir.

Örnek 13:

- Yarım saattir aynı şeye gülüyorum asdjoapsdopaskfopkasfpa
- Gülmekten ölüyorum mfdjddk

Önerilen modelde rastgele harflerle gülmeleri (RHG) algılamak için Türkçe derlemin karakter seviyesinde 2'den 5'e kadar n-gram modellemesi yapılmaktadır. Bununla birlikte bu durumda özellikle Ünlü Harf Restorasyon problemiyle düzeltilebilecek olan ünlü harf kullanılmadan yazılan tokenların da RHG olarak algılanma problemi oluşmaktadır. Bu nedenle Türkçe derlemimizdeki ünlü harfleri bütünüyle kaldırıp sadece sessiz harfler ile benzer şekilde karakter seviyesinde 2'den 5'e kadar n-gram modeli oluşturulmaktadır. Sonrasında girdi tokenı için ünlü harfleri içeren ve içermeyen şekilde 2'den 5'e kadar n-gram olasılıklarının toplamı hesaplanmaktadır. Toplam olasılık değeri eşik değerin altında ise kelime RHG olarak etiketlenmektedir.

### 6.1.7. Sözcük Ayırma

SMM'lerde karakter kısıtlamaları, hızlı yazımlar, gibi sebeplere bağlı olarak kelimelerin veya cümlelerin arasında ayırt edici bir harf, noktalama işareti gibi herhangi bir sembol veya boşluk bulundurmadan yapılan yazımlar söz konusudur. Örneğin "kaldığıyerden" şeklinde yazılan bir tokenın doğru formu "kaldığı yerden" şeklinde 2 token olmalıdır. Benzer şekilde "kendiobjektifimden" yazımı da "kendi objektifimden" şeklinde 2 token olmalıdır. Bir diğer örnekte 3 kelimeden oluşan doğru formu "Bulutlara esir olduk" cümlesi bütünüyle bitişik olarak "Bulutlaraesirolduk" şeklinde yazılmıştır.

Sosyal medyaya özgü bir diğer yazım biçimi de "BulutlaraEsirOlduk" şeklinde kelimelerin arasında boşluk bırakılmadan fakat ilk harfleri büyük olacak şekilde yazılmasıdır. Bu bölümde öncelikli olarak büyük ve küçük harfleri karışık şekilde içeren tokenlar büyük harflere göre bölünerek geçerli kelime olup olmadıkları kontrol edilir.

Hiç ayırt edici işaret bulunmayan yazımların kelimelere bölünmesi oldukça zordur. Daha önceki bölümlerde de belirtildiği gibi MRL diller için sözlük boyutunun aşırı büyük olacağı için tüm olası kelime formlarının bulunacağı bir sözlük oluşturup bu sözlükte arama yaparak doğru dizilimi bulmak Türkçe için çok olası değildir. Bu nedenle sadece kökler, ekler ve her iki grupta da yer alabilecek yüzey formlarından oluşan toplam 32 600 kelimelik bir sözlük oluşturuldu. Girdi tokenı ile eşleşen en uzun yüzey formları bulanarak DD ile bir arada bulabilirlikleri test edildi. Bulunan yüzey formu kök ise ve bundan sonra gelen kelime de kök ise ilk kök kelime olarak kabul edilir. Eğer sonrasında gelen ek veya her iki gurupta olan yüzey formları ise birleştirilerek test edilir. Başarısız olursa sadece her iki gurupta olan kelimeden itibaren ikiye bölünür ve ayrı ayrı test edilir.

### 6.1.8. Ünlü Harf Restorasyonu

İbranice ve Arapça gibi dillerin aksine Türkçe normal yazım dilinde ünlü harf problemi olan bir dil değildir. Öte yandan sosyal medya yazımlarında zaman zaman ünlü harfleri

hiç kullanmadan veya eksik ünlü harf kullanarak yapılan yazımlar mevcuttur. Arapça gibi dillerde bütünüyle sesli harf kullanılmadan yapılan yazımlar okuyucular tarafından genellikle rahatlıkla seslendirilebilirken Türkçe'nin formal yazım dilinde bu problem olmadığı için uzun ve yaygın olarak kullanılmayan kelimelerde seslendirme problemleri yaşanmaktadır. Bu nedenle birçok durumda ünlü harflerin bütünüyle yazılmaması yerine genellikle eksik yazılması ve çoğu durumda yaygın olarak kullanılan kelimelerde bu problemin bulunduğu görülmektedir. Bu nedenle bu problemin çözümü için sözlük tabanlı bir yaklaşıma odaklanılmaktadır. Yukarıdaki bölümlerde detayı verilen Türkçe derlemde bulunan kelimelerin ünlü harflerini çıkartılarak, kelimelerin ünsüz hallerinin ve olası aday kelimelerin bulunduğu bir sözlük oluşturuldu. DR problemine benzer şekilde Çizelge 6.4'te de görüldüğü aynı ünsüz dizilimine sahip birden fazla kelime bulunabilmektedir. Örneğin "bbn" ünsüz dizilimi için "baban", "babanı"," bobin" ve "beben" gibi olası kelimeler mevcuttur. Normalize edilecek token ünlü harf içermiyorsa doğrudan sözlükte arayarak aday kelimeleri bulunur. Tek bir aday kelime varsa normalizasyon işlemini tamamlanır. Birden fazla aday kelime varsa DR işleminde olduğu gibi Word2vec ile belirsizliği giderip en uygun adayı seçilir. Öte yandan normalize edilecek token ünlü harf içeriyorsa bunları çıkararak sözlükten aday kelimeleri arıyoruz ve tokenda bulunan sesli harflerin konumu ile uygun olmayan aday kelimeler elenir ve birden fazla aday kelime kalacak olursa yine belirsizlik giderme işlemi uygulanır.

Çizelge 6.4. Ünlü harf içermeyen bazı yüzey formları ve aday formları.

| Token | Aday Kelime |
|---|---|
| çks | Uçaksa |
| çks | Açıksa |
| çks | Çoksa |
| bbn | Beben |
| bbn | Bobin |
| bbn | Baban |
| kdnn | Kadının |
| kdnn | Kedinin |
| kdnn | Kodunun |

### 6.1.9. Yazım Hatası Düzeltme

Çizelge 1.3'teki örnekte daha önce de belirtildiği gibi Türkçe yapısı itibariyle birbirine benzer yüzey formlarında anlamsal olarak birbirinden farklı çok fazla kelime üretmeye müsait olan bir dildir. Bunun sonucu olarak hem aday üretme süreci hem de üretilen aday kelimeler arasından doğru kelimeyi seçme süreci oldukça zorlu bir görevdir. Bu nedenle mevcut arama ağacı tabanlı klasik yaklaşımlar MRL'ler için çok uygun değildir.

Bu aşamada problemin çözümü için bir düzenleme mesafesi uzaklığındaki kelimelere odaklanılmaktadır ve aday kelime üretme sürecine karakter silme, ekleme ve yer değiştirme adımlarından oluşan kural tabanlı bir yaklaşım uygulanmaktadır. Sonrasında üretilen aday kelimeler DD ile kontrol edilip morfolojik olarak geçerli aday kelimeler belirlenmektedir. Birden fazla aday kelime olması durumunda Word2vec kullanarak en uygun aday kelimeyi seçilmektedir.

### 6.1.10. Belirsizlik Giderici

Normalizasyon süreci esnasında birçok modül kendi içinde veya diğer modüllerle birlikte birden fazla aday kelime üretmektedir. Bu durumda ilgili metin içerik olarak değerlendirilerek anlamsal olarak en uygun kelime belirlenmelidir.

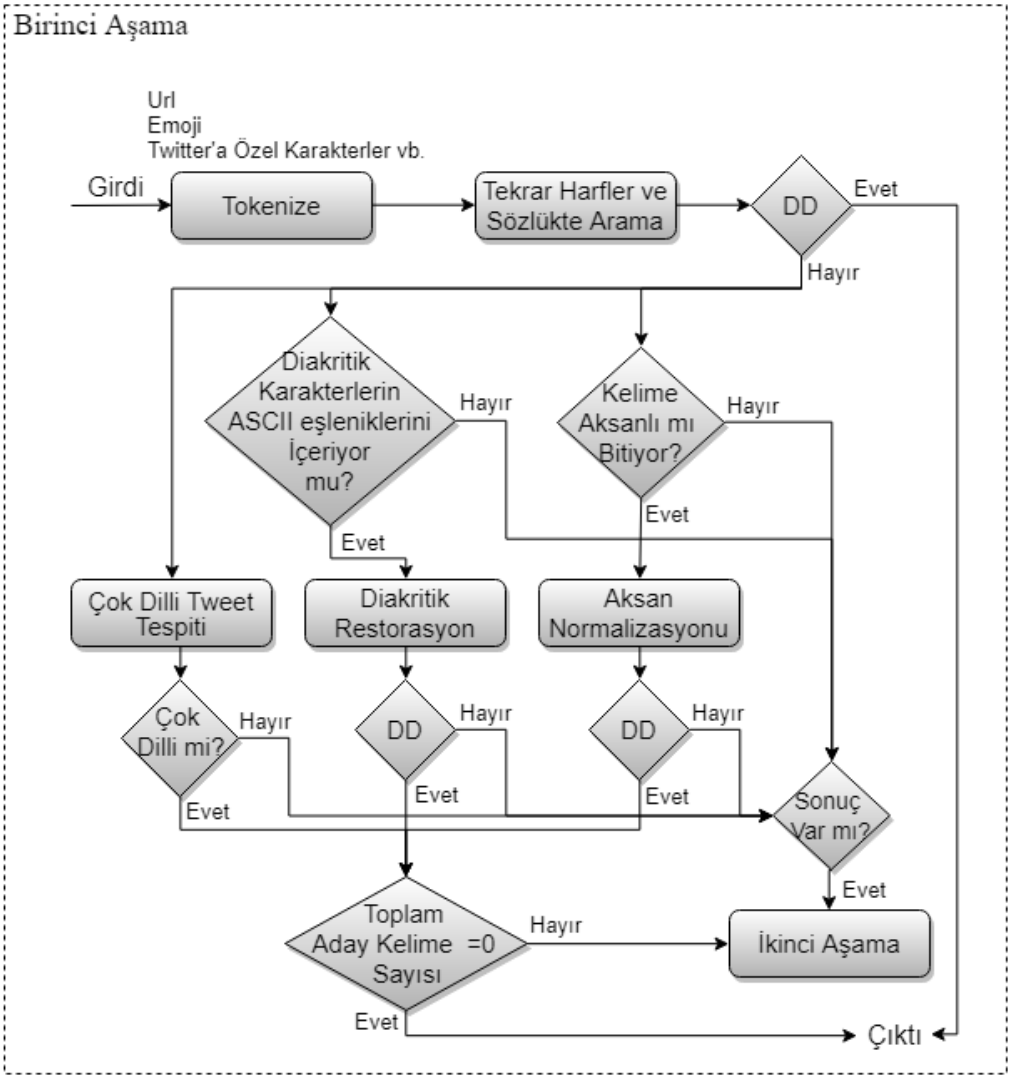
Kelimenin İngilizce mi veya Türkçe mi olduğu konusunda bir belirsizlik yaşanması durumunda Eşitlik (6.1) – (6.2)'deki olasılık değerlerine bakarak karar verilir. Öte yandan birden çok geçerli Türkçe kelime elde edilmiş ise bunların anlamsal bir değerlendirmesini yapmak için Word2vec aracından yararlanılır. Bu amaçla bir derlem kullanılarak, derlem içerisindeki her bir kelimenin özellik vektörü Word2vec araçlığıyla tespit edilir. Bu aşamadan sonra kelimenin diğer kelimelerle olan uyumu kosinüs benzerliği kullanarak hesaplanmaktadır. Burada "a" aday kelime ve "b" ilgili cümle içerisindeki diğer kelimeler, "v" diğer kelimelerin sayısı ve "p" toplam benzerlik oranı olmak üzere kosinüs benzerliği şu şekilde hesaplanır:

$$p = \sum_{t=1}^{v} \cos(a, b_t) = \frac{\sum_1^n (a_i \times b_{ti})}{\sqrt{\sum_1^n a_i^2} \times \sqrt{\sum_1^n b_{ti}^2}} \tag{6.3}$$

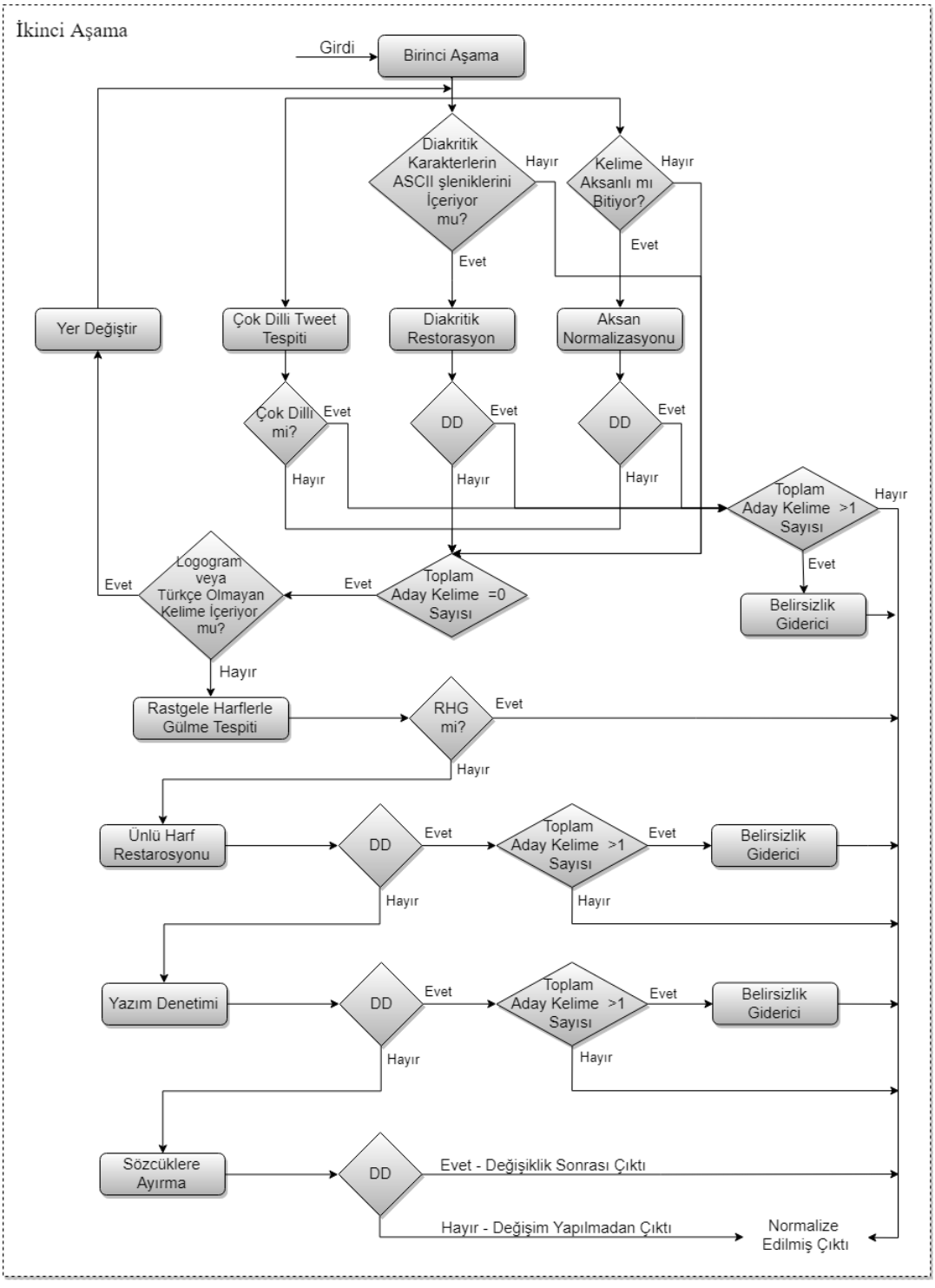
Her bir aday kelime için diğer kelimelerle olan toplam benzerlik değeri hesaplanır ve en yüksek benzerlik değerine sahip kelime normalizasyon işleminin sonucu olarak kabul edilir. Öte yandan herhangi bir kelime derlemde bulunmaz ise morfolojik analiz aracı elde edilen en uzun kökü esas alınarak benzerlik değeri hesaplanır.

## 6.2. HİBRİT MODEL

Her bir modülün davranışı diğer modüllerin performansını etkileyeceği için modüllerin hazırlanması kadar doğru bir mimari ile sunulması da oldukça önemli bir görevdir. Ayrıca aynı anda sonuç üretme ihtimali olan birimlerinde doğru bir şekilde değerlendirilmesi gerekir. Çok Dilli Tweet Tespiti (ÇDTT) bölümünde belirtilen "gun" ve "gün" örneğine benzer şekilde "alcak" şeklindeki bir hatalı girdi için DR ve Aksan Normalizasyon (AN) modülleri sırasıyla "alçak" ve "alacak" şeklinde iki farklı çıktı üreteceklerdir. Her iki kelime de geçerli sözcüklerdir. Bu nedenle model mimarisinde paralel ve kaskad yapıdan oluşan hibrit bir mimari önerilmektedir. Şekil 6.2 ve 6.3'de de detayları görüldüğü gibi aynı anda sonuç üretme ihtimali bulunan çok dilli tweet tespiti, aksan normalizasyonu ve DR birimleri paralel biçimde çalışırken diğer modüller kaskad çalışmaktadır. Buna ilave olarak belirsizlik durumlarında Word2vec kullanarak geçerli Türkçe kelimeler ile diğer kelimeler arasında kosinüs benzerliklerini kontrol ederek karar verilmektedir. Bu nedenle belirsizlik durumlarında daha fazla kelime ile değerlendirme yapabilmek adına çok dilli tweet tespiti, aksan normalizasyonu ve DR işlemleri iki aşamada ele alınmaktadır. İlk olarak bu modüllerin tamamından sadece bir aday kelimenin üretildiği durumu baz alınır. Tek aday kelime üreten tokenlar normalize edildikten sonra birden fazla aday kelimeye sahip olan takip tokenlar normalize edilir.




Şekil 6.2. Önerilen normalizasyon mimarisinin ilk aşamasına ait akış diyagramı.




Şekil 6.3. Önerilen normalizasyon mimarisinin ikinci aşamasına ait akış diyagramı.

## 6.3. DENEYSEL KIYASLAMA

Önerilen normalizasyon yaklaşımında birden fazla aday kelime üretilmesinden dolayı oluşan belirsizlik durumlarına karşılık diyakritik restorasyon, aksan restorasyonu, ünlü harf resterasyonu ve yazım denetimi modülleri Word2vec tabanlı bir belirsizlik giderme aracı ile genişletilmiştir.

Sonrasında önerilen modelin performansı MS Word 2016, Zemberek NLP aracı [54] ve [12]'deki çalışmalar ile kıyaslandı. [12]'deki çalışmaya ait online bir tool [143]'deki çalışmada bildirilmiştir.

Trafik verilerinin sınıflandırılması esnasında tüm harfler küçük harfe dönüştürülüp noktalama işaretleri temizlenmektedir. Bu nedenle kıyaslama işlemi büyük ve küçük harfe duyarsız olacak şekilde yapılmıştır. Ayrıca özel isimlere gelen eklerde kesme işareti bulunup bulunmamasıda göz ardı edilmiştir. Buna ilave olarak yabancı kelimelerin kıyaslanması işlemi gerçekleştirilirken sözcükte herhangi bir bozulma olmamışsa doğru olarak kabul edilmektedir. Normalizasyon işlemi insan denetçiler tarafından hatalı olduğu tespit edilmiş olan kelimeler ile özel isim olarak işaretlenmiş kelimeler üzerinde gerçekleştirilmiştir.

$$\text{Acc} = \frac{\text{correct normalized words}}{\text{OOV tokens}} \tag{6.4}$$

## 6.4. SONUÇLAR

Bu bölümde trafik veri setinin normalizasyon işleminin sonuçları değerlendirilmektedir. Önerilen normalizasyon işleminde belirsizlik durumlarıyla başa çıkabilmek için dört farklı modül Word2vec tabanlı belirsizlik giderici ile genişletilmiştir. Bu modüller sırasıyla diyakritik restorasyon, aksan restorasyonu, ünlü harf restorasyonu ve yazım denetimi modülleridir. Buna ilave olarak bütünüyle kaskad bir mimari yerine hibrit bir mimari kullanılmıştır.

Çizelge 6.5'da da önerilen model ile Zemberek [54], Eryiğit Kaskad [12] ve MS Word 2016'yla olan kıyaslamaları görülmektedir. Zemberek ve MS Word 2016 ile daha adil bir değerlendirme olması adına emojiler ve URL benzeri yazımlar kıyaslama performansına dahil edilmemiştir. Toplam 2 662 kelimenin Eşitlik (6.4) doğrultusunda performans değerlendirilmesi yapılmıştır. Buna göre en düşük başarım %37,38 ile Zemberek tarafından elde edilmiştir. Bununla birlikte Eryiğit Kaskad modeli ve MS Word ile birbirine oldukça yakın değerler elde edilmiştir. Sırasıyla başarım değerleri %59,62 ve %59,88'dir. Bununla birlikte her iki aracında farklı kelime gruplarında birbirlerine üstünlük sağladığı gözlemlenmiştir. Önerilen Word2vec tabanlı belirsizlik giderici ile genişletimiş normalizasyon yaklaşımında ise yaklaşık %10,4'lük bir başarım artışı ile %70,29'luk bir başarım elde edilmiştir.

Çizelge 6.5. Önerilen modelin normalizasyon başarısının literatürdeki güncel teknikler ile kıyaslaması.

| | Doğruluk (%) |
|---|---|
| Eryiğit Kaskad | 59,62 |
| Zemberek | 37,38 |
| MS Word | 59,88 |
| Önerilen Model | **70,29** |

Bu testlere ilave olarak normalize edilen kelime ile buna komşu olan tek bir kelimenin anlamsal olarak değerlendirilmesinin yeterli olup olmayacağını görebilmek için dört farklı durum daha test edildi. Bu testlerde normalize edilen kelimeden sonra gelen ilk kelime ve tweet içerisindeki diğer tüm kelimeler baz alınarak testler gerçekleştirildi. Ayrıca CBOW ve Skip-gram olmak üzere iki farklı Word2vec yönteminin başarı oranları da kıyaslandı. Çizelge 6.6'daki sonuçlarda da görüldüğü gibi en yüksek sonuçlar Skip-gram modeliyle tweetteki tüm kelimelerin değerlendirilmesi durumu için elde edilmiştir. Skip-gram yönteminde tek kelimenin ve tüm tweetin değerlendirilmesinde başarım oranında %3,2'lik artış yaşandığı görülmüştür. Benzer şekilde CBOW yöntemi içinde %1,92'lik bir artış gerçekleştiği görülmektedir. Bu bağlamda tek komşu kelime yerine tweet içerisindeki yazımı doğru olan tüm kelimelerin Word2vec modeli ile anlamsal olarak değerlendirilmesi normalizasyon

performansı açısından daha başarılı sonuçlar sağladığı görülmektedir. Ayrıca Skip-gram yöntemi her durumda CBOW yöntemine göre daha iyi sonuç sağlamıştır.

Çizelge 6.6. Farklı Word2vec modellerinin normalizayon performansı üzerine etkisi.

| | Doğruluk (%) |
|---|---|
| $CBOW_{one}$ | 66,45 |
| $CBOW_{all}$ | 68,37 |
| $Skip\text{-}Gram_{one}$ | 67,09 |
| $Skip\text{-}Gram_{all}$ | **70,29** |

Skip-Gram ve CBOW gibi farklı vektörel temsil elde etme yöntemlerinin yanı sıra vektörel benzerlik hesaplama yöntemlerinin normalizasyon performansına olan etkisi değerlendirilmiştir. Çizelge 6.7'de gösterilen Skip-gram modeli kullanılarak yapılan testlerde en yüksek sonuçlar kosinüs benzerliği ile elde edilmiştir. Öte yandan Öklid, Manhattan ve Minkowski yöntemleri ile aynı değerler elde edilirken bu değer kosinüs benzerliğinden biraz daha düşük olarak gerçekleşmiştir. En düşük sonuç Chebyshev yöntemi ile elde edilmiştir. Kosinüs benzerliği, iki vektör arasındaki mesafenin büyüklüğü değil vektörlerin birbirlerine göre göreceli olarak yönlendirilmesidir. Buna bağlı olarak küçük sentetik veri setimizde en iyi sonuçlar kosinüs benzerliği ile elde edilmiştir.

Çizelge 6.7. Farklı vektörel uzaklık ölçüm metotlarının normalizasyon performansına etkisi.

| Sistem | Doğruluk (%) |
|---|---|
| Kosinüs | **70,29** |
| Öklid | 69,53 |
| Manhattan | 69,53 |
| Minkowski | 69,53 |
| Chebyshev | 68,11 |

Sonuç olarak Türkçe'nin zengin ek ve kök yapısının birbirine yakın yüzey formlarında birçok kelime türetilmesine olanak sağlaması nedeniyle normalizasyon modüllerinin Word2vec tabanlı bir belirsizlik giderici ile genişletilmesi trafik veri setinin normalizasyon performansında güncel tekniklere kıyasla %10,4'lük bir iyileşme sağlamıştır.

# BÖLÜM 7

# TRAFİKLE İLGİLİ TWEETLERİN SINIFLANDIRILMASI

SMM'lerin yüksek miktarda gürültülü veri barındırması DDİ uygulamaları açısından önemli bir zorluk oluşturmaktadır. Normalizasyon işlemi ile metindeki gürültüyü azaltmak literatürde yaygın olarak kullanılan yöntemlerden bir tanesidir. Fakat Türkçe gibi morfolojik açıdan zengin diller için normalizasyon işlemi çok büyük zorluklar içermektedir. Gürültü problemiyle baş etmek için kullanılabilecek bir diğer alternatif yöntem de alana özel büyük etiketsiz derlemler kullanarak kelimelerin vektörel temsillerini elde etmek ve sınıflandırma işleminde bu vekörel temsillerden yararlanmaktadır. Tez çalışmasının bu bölümünde normalizasyon işleminin ve alana özel önceden eğitilmiş kelime temsillerinin kullanılmasının sınıflandırma performansı üzerine olan etkileri değerlendirilmektedir. Böylelikle normalizayon işleminin sağladığı katkının boyutu ile alana özel kelime temsillerinin kullanılmasının sağladığı katkının boyutu ortaya konmaktadır. Ayrıca her iki yaklaşımda LSTM, GRU ve CNN gibi güncel sınıflandırma yöntemleri üzerinde test edilerek her biri için elde edilen başarım ayrı ayrı değerlendirilecektir.

Sınıflandırma işlemi, tez çalışması kapasımda hazırlanan trafikle ilgili Türkçe Twitter mesajlarından oluşan veri seti üzerinde gerçekleştirilmektedir. Sınıflandırma işleminde kullanılan veri setleri Bölüm 5.1'de tanıtılmıştır. Bu kapsamda tweetler öncelikli olarak trafikle ilgili olanlar ve olmayanlar olmak üzere 2 farklı sınıfa ayrılmaktadır. Bir diğer yaklaşımda da trafikle ilgili sürücü hatası, yol çalışması ve kaza gibi daha özel durumları tespit edebilmek için 8 farklı sınıfta değerlendirme yapılmaktadır. Böylelikle trafiği yöneten otoritelerin bu tip özel durumları sosyal medyadaki metin içeriklerinden hızlı bir şekilde tespit edebilmesi hedeflenmektedir. Ayrıca hem normalizasyon işleminin hem de alana özel kelime temsili kullanmanın etkisi iki sınıflı ve çok sınıflı veri seti üzerinde kıyaslanmaktadır.

Son yıllarda kelime temsillerinden elde edilen özellik vektörleri metin sınıflandırma işleminde TF-IDF gibi metotların yerine yaygın olarak kullanılmaya başlanmıştır. Kelime temsillerinin elde edilmesinde kullanılan kaynaklar çeşitlilik gösterebilmektedir. Bu doğrultuda kelime temsilleri doğrudan sınıflandırma işleminde kullanılan etiketli veriden [144] elde edilebileceği gibi önceden eğitilmiş kamuya açık vektörel temsiller de sınıflandırma işleminde kullanılabilmektedir [145]. Bunun yanı sıra alana özel etiketsiz veriler dahil edilmek suretiyle de [146] kelime temsilleri elde edilebilmektedir.


Şekil 7.1. Etiketsiz Trafik derlemiyle elde edilen kelime temsilinde sırasıyla "şöför" ve "şofór" kelimeleri ile en benzer 50 kelime.

Kullanıcılar tarafından üretilen gürültülü metinlerden elde edilen kelime temsillerinde, bir sözcüğün hatalı formunun etrafında toplanan kelimeler ile doğru yazım formunun etrafında toplanan kelimeler büyük ölçüde benzerlik gösterebilmektedir. Şekil 7.1'de etiketsiz trafik derleminden CBOW yöntemi ile elde edilen kelime temsillerinde "şofór" kelimesi ve yaygın olarak kullanılan hatalı yazım formlarından biri olan "şöför" kelimelerine en fazla benzeyen 50'şer kelime görülmektedir. Çizelge 7.1'de de görüldüğü her iki yazım formu için en fazla benzeyen kelimelerin 21 tanesi birebir aynıdır. Yani yanlış yazım formlarının etrafında toplanan kelimeler doğru yazım formundaki kelimeyle oldukça benzeşmektedir. Bu durumda büyük boyutlu etiketsiz

verilerden elde edilen alana özel kelime temsillerinin kullanılmasını sınıflandırma başarımı açısından oldukça avantajlı kılmaktadır.

Çizelge 7.1. "Şoför" ve "şöför" yazımları için ortak benzer kelimeler.

| | | |
|---|---|---|
| acemi | muavin | şoförü |
| aracı | müşteri | şoförün |
| bayan | neyinpeşindeacaba | şöförü |
| direksiyonda | sürücü | taksi |
| kadın | sürücüleri | taksici |
| kullanan | şoförler | taksiciler |
| motorsiklet | şoförleri | yolcu |

Öte yandan Tükçe çok zengin bir kelime hazinesine sahip olan bir dildir. Buna bağlı olarak sosyal medyada karşılaşılan yazım yanlışı türleri de çok fazla çeşitlilik gösterebilmektedir. Normalizasyon bölümünde de belirtildiği gibi SMM'lerde en sık karşılaşılan problemlerden bir tanesi de diyakritik karakter problemidir. Ayrıca diyakritik karakter problemiyle genellikle cümle içerisinde birden fazla kelimede karşılaşılmaktadır. Bu durumda komşu kelimlerin de hatalı formda olmasına neden olmaktadır. Bu durumla ilgili olarak Şekil 7.2'de "araç" kelimesi ve bu kelimenin ASCII formu olan "araç" şeklindeki yazım için, Şekil 7.1'deki ile aynı şekilde elde edilmiş benzer kelimeler görülmektedir. "Şoför" ve "şöför" şeklindeki yazımlarda hatalı ve doğru yazım biçimlerinin etrafında aynı kelimeler toplanırken diyakritik karakter problemi için durum biraz farklılık göstermektedir. Çizelge 7.2'te de görüldüğü gibi "araç" kelimesinin ASCII formundaki yazım biçiminin etrafında kelimelerin ASCII formlarının toplandığı görülmektedir. Örneğin "minibüs" kelimesi "araç" kelimesi ile benzerlik gösterirken bu kelimenin ASCII formu olan "minibüs" yazımı ise "araç" kelimesi ile benzerlik göstermektedir. Yani aynı kelimelerin doğru formları "araç" şeklindeki doğru yazımın etrafında toplanırken, ASCII formları da "araç" şeklindeki yazımın etrafında toplanmaktadır.

Tez çalışmasının bu bölümünde alana özel kelime temsillerinin, benzer kelimeleri sözcüklerin hatalı ve doğru yazım biçimlerinin etrafında toparlayabilme kabiliyetleri ve normalizasyon işleminin kelimeleri doğru yazım biçimlerine çevirebilme kabiliyetleri sınıflandırma görevi üzerinde test edilmektedir. Her iki yöntemin

sağladığı faydalar ve negatif yönleri değerlendirilmektedir. Ayrıca normalizasyon işleminin zorlukları göz önüne alındığında normalizasyona ihtiyaç duymadan yüksek başarı oranları ile bir sınıflandırma işleminin gerçekleştirilip gerçekleştirilemeyeceği değerlendirilmektedir. Bunlara ilave olarak Türkçe SMM'lerden trafik olaylarının yüksek başarım oranları ile sınıflandırılabilme kabiliyeti değerlendirilmektedir.


Şekil 7.2. Etiketsiz Trafik derlemiyle elde edilen kelime temsilinde sırasıyla "arac" ve "araç" kelimeleri ile en benzer 50 kelime.

Çizelge 7.2. "Araç" ve "Arac" yazımları için ortak benzer kelimeler.

| arac ile Benzer Kelimeler | araç ile Benzer Kelimeler | arac ile Benzer Kelimeler | araç ile Benzer Kelimeler |
|---|---|---|---|
| araci | aracı | araclara | araçlara |
| araclar | araçlar | suruculer | sürücüler |
| aracin | aracın | minibus | minibüs |
| araclarin | araçların | sirket | şirket |
| otobus | otobüs | arabalarin | arabaların |
| araclari | araçları | aracta | araçta |
| arabanin | arabanın | | |

## 7.1. ÖNERİLEN YAKLAŞIM

Tweetlerin sınıflandırılması aşamasında CNN ve RNN modelleri kullanılmaktadır. Bölüm 4'te de belirtildiği gibi farklı tipte RNN modelleri mevcuttur. Bu tez çalışmasında sınıflandırma işlemi için temel RNN modelinin yanı sıra GRU, LTM ve BiLSTM modelleride kullanılmıştır. CNN ve RNN modellerine veriler farklı şekillerde uygulanabilmektedir. CNN'ler esasında görüntü işleme uygulamaları için geliştirilmiş olmasından dolayı matris şeklinde bir girdiye ihtiyaç duymaktadırlar. Bu durumda metin verisinin matris formatına dönüştürülerek uygulanması gerekmektedir. Bu doğrultuda *c* tweet içerisindeki kelime sayısı ve *n* her bir kelime kelime vektörünün boyutu olmak üzere *c x n* boyutlu bir matris ağın girişine uygulanır. Buradaki önemli kısıtlamalardan bir tanesi CNN ağlarının sabit boyutlu girdilere ihtiyaç duymasıdır. Bu doğrultuda maksimum bir kelime sayısı belirlenerek tüm tweetler aynı uzunlukta olacak şekilde ağın girşine uygulanmaktadır. Eğer tweet belirlenen maksimum kelime sayısından daha az sayıda kelimeye sahip ise kalan kısımlar 0 vektörleri ile temsil edilmektedir. Öte yandan RNN'ler ise dizilim şeklinde girdiye ihtiyaç duymaktadırlar. Tweet içerisindeki her bir kelime zaten bir dizilim oluşturduğu için kolaylıkla ağa uygulanabilmektedir. Burada CNN'ye benzer şekilde kelimeler önce vektörel temsillere dönüştürülmekte sonrasında ise her biri bir dizilim şeklinde RNN ağına uygulanmaktadır.

Sınıflandırma işleminde veriler üç farklı şekilde kullanılmaktadır. Bunlardan birincisinde veri normalizasyon işlemine tabi tutulmadan doğrudan orijinal haliyle kullanılmaktadır. Bir diğer yöntem de Bölüm 6'de tanıtılan normalizasyon yaklaşımı ile normalize edilen veriler sınıflandırma işleminde kullanılmaktadır. Son olarak ise verilerin 2 insan denetçi tarafından manuel olarak normalize edilmiş şekilleri kullanılarak sınıflandırma işlemi gerçekleştirilmektedir. Bununla birlikte her üç yöntem de bazı ön işlem adımları uygulanmaktadır. Örnek 14'deki tweetlerde de görüldüğü gibi URL'ler genellikle her bir tweet için farklıdır. Sınıflandırma işleminden önce tüm veri modellerinde URL'ler "url" şeklinde bir ifadeye dönüştürülmektedir. Buna ilave olarak her bir karakter küçük harfe dönüştürülmekte ve sonrasında ise noktalama işaretleri temizlenmektedir. En son aşamada ise sayısal karakter içeren ifadeler çıkarılmaktadır. Her üç veri modeline de bu ön işleme adımları

uygulandıktan sonra sınıflandırma işlemine hazır hale gelmektedir. Şekil 7.3'te veri hazırlama basamakları görülmektedir.

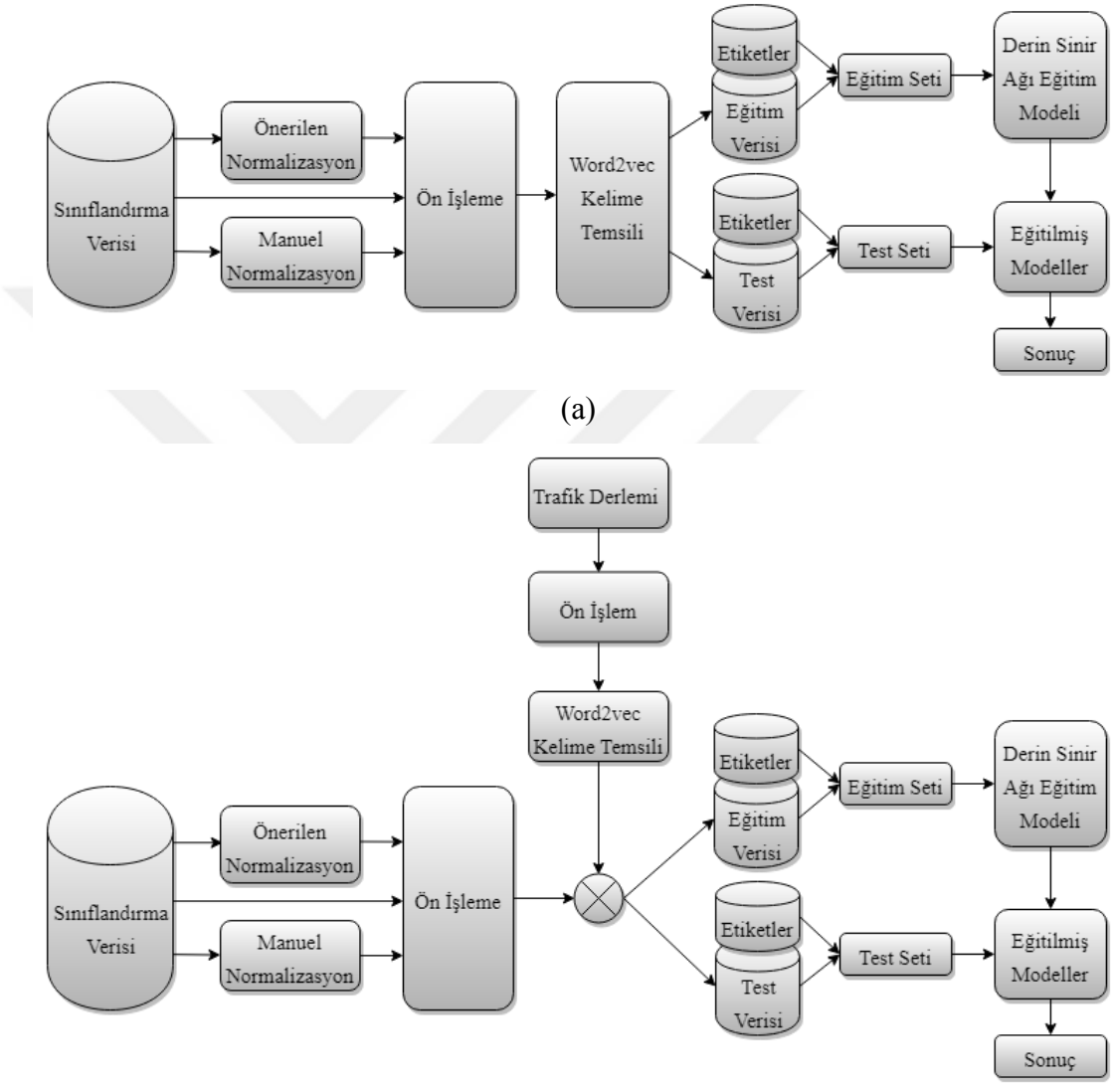
Örnek 14:

- #TrafikteBGN Avcılar-Parseller yönündeki araç arızası bölgede trafik yoğunluğu oluşturmaktadır http://gez.io/byTMQJ
- #TrafikteBGN Haliç-Okmeydanı istikametinde araç arızası nedeniyle trafik yoğunlaşmaktadır http://gez.io/bsvEWt
- @muratkazanasmaz edenler otogardan TEM bağlantı yolunda kamyonun araç arızası yüzenden trafik felç http://yfrog.com/klkqhxlj
- @MuratKazanasmaz D-100 Pendik Kartal yönü araç arızası.. trafik cok yogun..

Veri hazırlama işlemlerinin yanı sıra son derece kritik olan bir diğer konu da kelimelerin vektörel temsillerinin oluşturulmasıdır. Bu tez çalışmasında vektörel temsilleri oluşturmak için iki farklı yaklaşım kullanılmaktadır. İlk olarak kelimelerin vektörel temsilleri sadece sınıflandırma için kullanılan etiketli veriden elde edilmektedir. Ayrıca burada önemli olan konulardan bir tanesi de kelime temsilleri elde edilme aşamasında da veri hazırlama basamaklarındaki işlemler uygulanmaktadır. Örneğin manuel olarak normalize edilmiş veri üzerinde sınıflandırma işlemi gerçekleştirilecekse kelime temsilleri de benzer şekilde manuel olarak nomalize edilmiş veriden elde edilmektedir veya orijinal verinin sınıflandırma işlemi gerçekleştirilecekse sadece ön işleme yöntemleri uygulanmış etiketli veriden kelime temsilleri elde edilmektedir. İkinci yaklaşımda ise kelime temsilleri alana özel etiketsiz verilerek kullanılmak suretiyle elde edilmektedir. Burada hangi tip veri hazırlama yönteminin kullanıldığına bakılmaksızın trafik alanına özel etiketsiz veriden elde edilmiş önceden eğitilmiş bir kelime temsili kullanılmaktadır. Alana özel etiketsiz veriden kelime temsillerinin elde edilmesi aşamasında da ön işleme adımları uygulanmaktadır.

Son aşamada ise normalizasyon ve alana özel kelime temsili kullanmanın sınıflandırma başarımı üzerine olan etkisini değerlendirmek için orjinal veri, önerilen normalizasyon yaklaşımı ile normalize edilen veri ve manuel olarak normalize edilmiş

veri doğrudan etiketli sınıflandırma verilerinden elde edilen kelime temsilleri üzerinde test edilmektedir ve her birinin başarımı kıyaslanmaktadır. Sonrasında ise alana özel kelime temsili kullanımının sağladığı katkıyı değerlendirebilmek için her üç veri tipi alana özel kelime temsili kelime temsili kullanılarak sınıflandırma başarımları test edilmektedir. Şekil 7.3'te önerilen kıyaslama yaklaşımının diyagramı görülmektedir.




Şekil 7.3. a) Etiketli veriden kelime temsillerinin elde edildiği sınıflandırma yaklaşımı. b) Alana özel etiketsiz veriden kelime temsillerinin elde edildiği sınıflandırma yaklaşımı.

## 7.2. DENEYSEL KIYASLAMA

Tez çalışmasının bu bölümünde trafik verilerinin sınıflandırılmasında kullanılan sistemin yapısı hakkında bilgiler verilmektedir. Sınıflandırma işleminde Bölüm 5.1'de tanıtılan trafik veri setleri kullanılmaktadır. Sınıflandırma görevinde hem ikili sınıflandırma işlemi hem de sekiz farklı sınıftan oluşan çoklu sınıflandırma işlemi gerçekleştirilmektedir.

İkili sınıflandırma işleminde kullanılan veri setinde toplam 2 000 adet tweet bulunmaktadır. Bu tweetlerin %80'nine karşılık gelen 1 600 adedi eğitim setinde kullanılırken geri kalan 400 adet tweet test setinde kullanılmıştır. Öte yandan çok sınıflı veri seti toplam 1 200 adet tweetten oluşmaktadır. İki sınıflı veri setine benzer şekilde bu veri setinde de tweetlerin %80'i eğitim için ve geri kalan %20'si test işlemi için kullanılmıştır. Yani toplamda 960 adet tweet eğitim için 240 adet tweet de test için kullanılmıştır. Tüm testlerde eğitim ve test verileri birbiri ile örtüşmeyecek şekilde 5 parça çapraz doğrulama kullanılmıştır. Sınıflandırma başarımının değerlendirilmesinde aşağıdaki ölçütler kullanılmıştır:

$$\text{Başarım}=\frac{DP+DN}{DP+YN+YP+DN} \tag{7.1}$$

$$\text{Kesinlik}=\frac{DP}{DP+YP} \tag{7.2}$$

$$\text{Hassasiyet}=\frac{DP}{DP+YN} \tag{7.3}$$

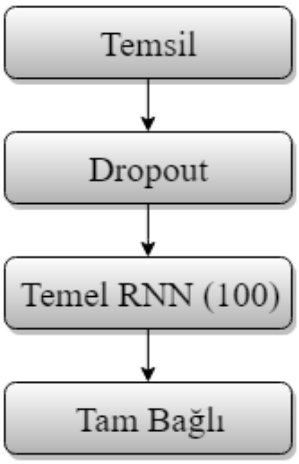
$$F_1=2\cdot\frac{\text{Kesinlik}\cdot\text{Hassasiyet}}{\text{Kesinlik+Hassasiyet}}=\frac{2DP}{2DP+YP+YN} \tag{7.4}$$

## 7.3. MODEL MİMARİLERİ

Trafik verilerinin sınıflandırılması için farklı tipte RNN ve CNN mimarileri oluşturulmuş ve test edilmiştir. Mimarileri oluşturulmasında ve test edilmesinde Python dilinde geliştirilmiş olan Tensorflow ve Keras kütüphaneleri kullanılmıştır. Test ve karşılaştırma işlemleri geçekleştirilen model mimarileri şu şekildedir:

### 7.3.1. Temel RNN Modeli

Dizi verileri üzerinde yüksek başarım sergilemeleri nedeniyle sınıflandırma işleminde çok sayıda RNN modeli kullanılmıştır. Bu kapsamda ilk kullanılan model Temel RNN modelidir. Temel RNN modeli kısa süreli bağımlılıklar aktarılması konusunda oldukça başarılı iken uzun süreli bağımlılıklar konusunda başarılı değildir.



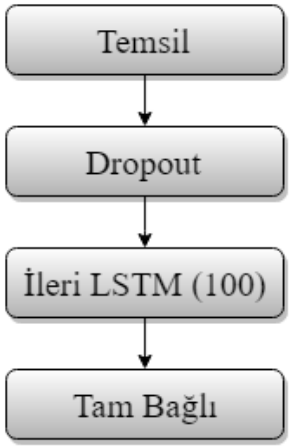

Şekil 7.4. Temel RNN model mimarisi.

Kullanılan Temel RNN modeli Şekil 7.4'te görüldüğü gibi tek katmanlı ileri yönlü ve 100 algılayıcadan oluşan bir RNN katmanı içermektedir. Bu katmanın çıkışı tam bağlı katmana girdi olarak uygulanmaktadır. Öte yandan RNN katmanından önce ise sırasıyla temsil katmanı ve dropout katmanları bulunmaktadır. Son olarak modelin çalıştırılmadan önce derleme işlemenin gerçekleştirilmesi gerekmektedir. Bu işlem için hata fonksiyonu ve optimizer gibi parametrelerin belirlenmesi gerekir. Optimizer için tüm modellerde Adam algortiması kullanılmıştır. Hata fonksiyonu olarak da iki

sınıflı veri setleri için “binary_crossentropy” kullanılırken çok sınıflı veri setleri için “categorical_crossentropy” kullanılmıştır.

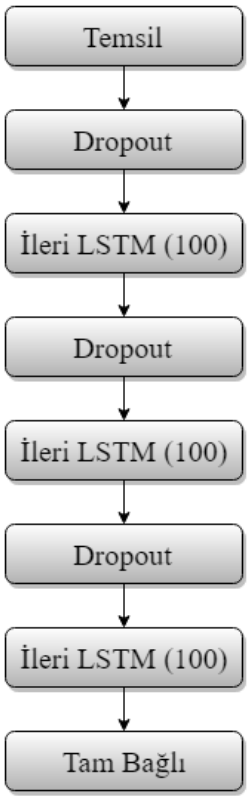
### 7.3.2. Uzun Kısa Dönem Hafıza Modelleri

Son yıllarda literatürde yaygın olarak kullanılan RNN modellerinden bir tanesi de LSTM modelidir. Temel RNN modelinin aksine LSTM modeli uzun süreli bağımlılıkları aktarma konusunda da başarılıdır.




Şekil 7.5. Tek katmanlı ileri yönlü LSTM model mimarisi.

LSTM modeli kullanılırken farklı algılayıcı sayıların ve daha derin mimarilerin kullanımının etkilerini değerlendirebilmek adına yedi farklı mimari gerçekleştirilmiştir. Kullanılan esas LSTM modeli Temel RNN modeline oldukça benzemektedir. Şekil 7.5’te görüldüğü gibi mimaride Temel RNN katmanı yerine 100 algılayıcıdan oluşan tek katmanlı ileri yönlü LSTM katmanı kullanılmıştır. Ayrıca tek katmanlı LSTM modeli algılayıcı sayılarının etkisini değerlendirmek adına sırasıyla 75 ve 125 algılayıcı sayıları ile de test edilmiştir.



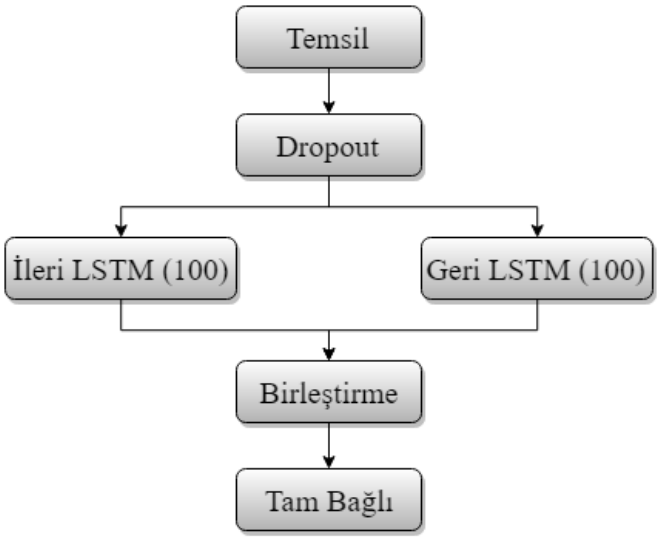

Şekil 7.6. Üç LSTM katmanından oluşan ileri yönlü model mimarisi.

Kullanılan mimariler için kritik konulardan bir tanesi de kaç tane katman kullanılmasının gerekli olduğudur. Özellikle büyük boyutlu verilerde daha derin mimariler kullanmak sınıflandırma performansına olumlu katkı sağlayabilmektedir. Daha derin mimarilerin trafik verisinin sınıflandırılması üzerindeki etkilerini değerlendirebilmek için 2 ve 3 LSTM katmanı içeren modeller test edilmiştir. Kullanılan 3 katmanlı LSTM modeli Şekil 7.6'da görülmektedir. Mimaride her biri 100 algılayıcıdan oluşan toplam 3 adet ileri yönlü LSTM katmanı kullanılmıştır. Her bir LSTM katmanında önce aşırı öğrenme problemini önleyebilmek için dropout katmanı yerleştirilmiştir.

Öte yandan 75 ve 125 algılayıcı sayısından oluşan modellerde tek katmanlı ve 2 katmanlı mimariler kullanılmıştır.

### 7.3.3. Çift Yönlü Uzun Kısa Dönem Hafıza Modeli

LSTM modelleri ileri yönlü olarak çalışmaktadır fakat bazı NLP uygulamalarında gelecekteki temsillerin de öğrenilmesi olumlu katkı sağlayabilmektedir. Bu durumda ileri yönlü çalışan bir LSTM'nin yanı sıra geri yönlü çalışan bir LSTM ilave edilerek gelecekteki temsillerin etkisi de modele dahil edilebilmektedir. Kullanılan BiLSTM modeli LSTM modelindeki esas modele benzer şekilde 100 algılayıcıdan oluşmaktadır. LSTM modelinden farklı olarak bir adet 100 algılayıcıdan oluşan LSTM katmanına ilave olarak bir adette yine 100 algılayıcıdan oluşan geri yönlü LSTM katmanı bulunmaktadır. LSTM katmanlarının öncesinde sırasıyla temsil katmanı ve dropout katmanları bulunmaktadır. LSTM katmanlarının çıkışı birleştirilerek tam bağlı katmana giriş olarak uygulanmaktadır. Şekil 7.7'de kullanılan model görülmektedir.

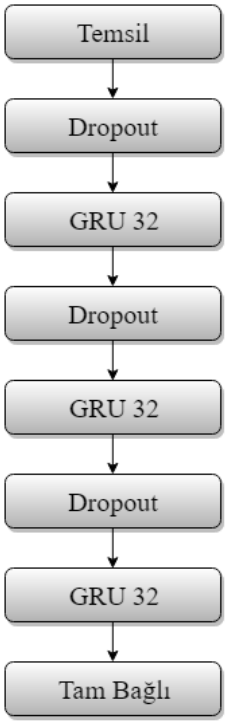



Şekil 7.7. BiLSTM model mimarisi.

### 7.3.4. Kapılı Tekrarlamalı Ünite Modelleri

Trafik verilerinin sınıflandırılmasında kullanılan bir diğer RNN modeli de GRU'lardır. GRU modülleri güncelleme ve sıfırlama kapılarından meydana gelmektedir. GRU modellerinde de LSTM modellerine benzer şekilde farklı algılayıcı sayılarının ve farklı GRU katman sayılarının etkisi değerlendirilmiştir. Bu işlem için toplamda sekiz farklı GRU modeli kullanılmıştır. GRU'nun esas modelinde temel RNN ve LSTM

modelleriyle aynı mimari kullanılmıştır. Buradaki tek farklılık Temel RNN katmanı veya LSTM katmanı yerine 32 algılayıcıdan oluşan GRU katmanı kullanılmasıdır.




Şekil 7.8. Üç GRU katmanından oluşan model mimarisi.

Bununla birlikte derin mimari ve algılayıcı sayılarındaki değişimin etkisini de gözlemlemek amacıyla toplam sekiz farklı GRU mimarisi oluşturulmuştur. Şekil 7.8'de 3 GRU katmanı ve 32 algılyaıcıdan oluşan model görülmektedir. Diğer modellere benzer şekilde modelin girişinde temsil katmanı ve her GRU katmanından öncede dropout katmanı yer almaktadır.

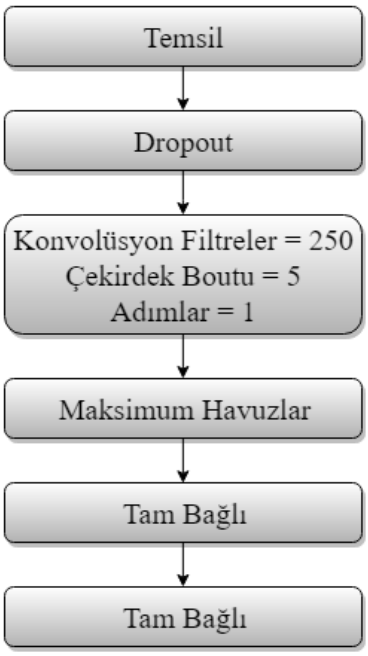
Kullanılan modellerde sırasıyla 16, 32 ve 40 algılayıcı sayıları kullanılmıştır. 16 ve 32 algılayıcı sayılarında 3 katmana kadar olan mimariler test edilirken 40 algılayıcıdan oluşan modellerde tek GRU ve iki GRU katmanından oluşan modeler test edilmiştir.

### 7.3.5. Konvolüsyonal Nöral Network Modeli

CNN'ler esasında bilgisayar görmesi problem için önerilmiş olmaları nedeniyle RNN'lerden farklıdır. RNN'ler dizi şeklinde girdilere ihtiyaç duyarken CNN'ler

matris formunda girdiye ihtiyaç duymaktadırlar. Buna ilave olarak girdilerin sabit boyutlu olması gerekmektedir. Ayrıca NLP uygulamalarında kullanılan CNN'lerin girdilerine karakter seviyesinde veya kelime seviyesinde girişler uygulanabilmektedir. Burada kullanılan modelde kelime seviyesinde girdi kullanılması tercih edilmiştir. Ayrıca her bir kelime sabit uzunluklu bir kelime temsili ile ifade edilmiştir. Öte yandan girdinin diğer boyutunu da tweet içerisindeki bulunan her bir kelime oluşturmaktadır.

CNN'ler dizi verileri üzerinde çalışmak için tasarlanmamış olmakla birlikte CNN'leri avantajlı kılan önemli bir özellikleri örüntünün imge üzerindeki pozisyonuna karşı duyarsız olmalarıdır. Bu durumda CNN'lerin pek çok uygulamada oldukça başarılı olmasını sağlamaktadır.




Şekil 7.9. CNN model mimarisi.

Şekil 7.9'da da görüldüğü gibi konvolüsyon katmanı çekirdek genişliği 5 olan 250 filtreden oluşmaktadır. Ayrıca konvolüsyon katmanında aktivasyon fonksiyonu olarak ReLU kullanılmıştır. Konvolüsyon katmanının çıkışında maksimum havuzlama katmanı bulunmaktadır. Maksimum havuzlama katmanın çıktısı ise 250 algılayıcıdan oluşan bir tam bağlı katmana uygulanmaktadır. Bu katmanı bir dropout katmanı takip etmektedir ve sonrasında bir tane daha tam bağlı katman bulunmaktadır.

### 7.3.6. Konvolüsyonal Nöral Network – LSTM modeli

CNN modellerinin ve RNN modellerinin birbirilerine göre avantajlı ve dezavantajlı olduğu durumlar bulunmaktadır. Bu doğrultuda test işlemleri için CNN ve LSTM modellerinin bir arada bulunduğu bir başka model daha kullanılmıştır.

Bu modelin temelinde Bölüm 7.3.5'te tanıtılan CNN modeli kullanılmıştır. Burada sadece maksimum havuzlama katmanını takip eden tam bağlı katman çıkarılarak onun yerine bir LSTM katmanı kullanılarak model elde edilmiştir.

### 7.3.7. Model Parametreleri

Hiper parametrelerin belirlenmesi önemli bir zorluk kaynağıdır. Veri setine ve problemin türüne bağlı olarak hiper parametrelerde değişiklik gösterebilmektedir. Yani tüm veri setleri için iyi sonuç verecek ortak bir hipermatre bulunmamaktadır.

Öncelikli olarak belirlenmesi gereken hiper parametrelerden bir kısmı kelime temsilleri ile ilişkilidir. Sınıflandırma işleminde kullanılan kelime temsilleri Word2vec CBOW yöntemi ile elde edilmiştir. Burada temel vektör uzunluğu 100 olarak belirlenmiştir. Bununla birlikte sırasıyla 50, 75 ve 125 olmak üzere farklı vektör boyutları ile de kelime temsilleri elde edilmiş ve sınıflandırma performansları değerlendirilmiştir. Kelime temsilleri ile ilgili kullanılan bir diğer parameter de pencere uzunluğudur. Pencere uzunluğu kavramı ilgili kelimenin sağında ve solunda dikkate alınması gereken bağlam sayısını ifade etmektedir. Sınıflandırma işlemi için kullanılan tüm modellerde pencere uzunluğu 5 olarak belirlenmiştir. Kelime temsillerinde kullanılan diğer bir parameter de minimum bulunma sayısıdır. Minimum bulunma sayısı kelimenin derlem içerisinde kaç kez geçtiğini kontrol eder ve eşik değerin altındaki kelimeler için vektörel temsil elde edilmez. Burada minimum bulunma sayısı 1 olarak belirlenmiştir. Yani derlemdeki tüm kelimeler için bir vektörel temsil elde edilmektedir. Sistemin başlanlangıç öğrenme katsayısı (learning rate) 0,025 olarak kullanılmıştır. Öğrenme katsayısı eğitim işlemi esnasında lineer olarak azaltılmaktadır ve en düşük değeri 0,0001 olarak kullanılmaktadır. Kelime temsillerinde kullanılan son hiper parametrede devir sayısıdır (epoch). Devir sayısı

tüm verinin kaç kez eğitildiğini ifade etmektedir. Sistemin devir sayısı 5 olarak belirlenmiştir.

Kelime temsillerinin sınıflandırma işleminde kullanılması esnasında karşılaşılan problemlerden bir tanesi önceden eğitilmiş bir derlem kullanıldığında sınıflandırma verisindeki bir kelimenin bu derlemde bulunmayabileceğidir. Bu durumda kullanılan farklı yaklaşımlar mevcuttur. Bu kelimenin için rastgele bir kelime vektörü üretilebilir veya tamamen 0'lardan oluşan bir vektör ile temsil edilebilir. Bu tez çalışmasında bu tip durumlar için 0 vektörleri kullanılmıştır.

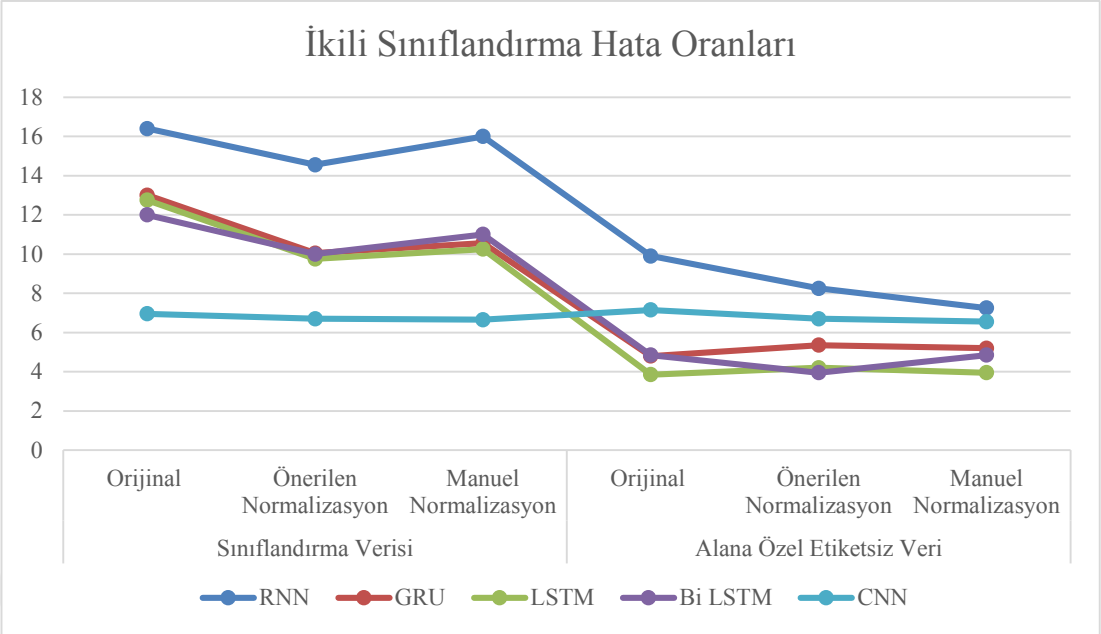
Yapay sinir ağı modelleri ile ilgili belirlenmesi gereken en önemli hiper parametrelerden bazıları gizli katman sayısı ve her bir katmandaki algılayıcı sayılarıdır. Kullanılan algılayıcı ve katman sayıları ile bilgiler yukarıdaki bölümlerde belirtilmiştir. Ayrıca LSTM ve GRU modellerinde farklı algılayıcı ve katman sayıları ile testler gerçekleştirilerek etkileri değerlendirilmiştir. İkili sınıflandırma problemi için tüm modellerde hata fonksiyonu olarak "binary_crossentropy" ve çoklu sınıflandırma problemi için de "categorical_crossentropy" kullanılmıştır. Tüm modellerde eğitim işlemi için Adam optimizer algoritması kullanılmıştır. Ayrıca tüm modellerde maksimum dizlim uzunluğu 30 olarak kullanılmıştır. Ayrıca modellerde kullanılan devir sayıları sınıflandırma türüne ve kullanılan kelime vektörünün türüne göre değişmekle birlikte aynı sınıflandırma türü ve kelime vektörü için aynı devir sayılarında eğitim işlemi gerçekleştirilmiştir. İkili sınıflandırma probleminde sınıflandırma verisinden kelime temsillerinin elde edildiği modeler için devir sayısı 42 olarak belirlenmiştir. Öte yandan yine ikili sınıflandırma probleminde önceden eğitilmiş kelime temsillerinin kullanıldığı modelde öğrenme işleminin daha hızlı gerçekleştiği görülmüş ve devir sayısı 22 olarak belirlenmiştir. Çoklu sınıflandırma probleminde kelime temsillerinin sınıflandırma verisinden elde edildiği durum için devir sayısı 150 olarak kullanılırken önceden eğitilmiş kelime temsillerinin kullanılması durumu için devir sayısı 32 olarak belirlenmiştir.

## 7.4. SONUÇ VE DEĞERLENDİRME

Sosyal medya platformları son yıllarda hemen hemen her türlü konunun tartışıldığı ve her alanda bilgi verildiği bir platform haline geldi. Bu anlamda içerisinde çok miktarda keşfedilmeyi bekleyen veri bulunmaktadır. Fakat son derece yüksek miktarda gürültülü veri barındırması nedeniyle metin madenciliği uygulamaları için oldukça zorlu bir görevdir. Bu gürültünün azaltılması için İngilizce gibi dillerde önerilen birçok normalizasyon yaklaşımı bulunmaktadır. Fakat morfolojik açıdan zengin dillerin bu konuda kendine has problemleri vardır ve bu problemler normalizasyon işlemini oldukça zorlu bir hale getirmektedir. Öte yandan son yıllarda etiketsiz verilerden elde edilen alana özel kelime temsillerinin kullanılması sınıflandırma başarımlarının yukarı taşınmasında önemli bir katkı sağlamaktadır. Bu tez çalışmasında öncelikli olarak Türkçe SMM'lerin normalizasyonunu belirsizlik durumları için kelime temsilleri ile genişleten bir yaklaşım sunulmuştur. Tez çalışmasının bu bölümünde ise normalizayon işlemi ve alana özel kelime temsilleri kullanmanın etkileri değerlendirilmektedir. Sınıflandırma işlemi hem iki sınıflı trafik veri seti üzerinde hem de çok sınıflı trafik veri seti üzerinde gerçekleştirlmektedir. Ayrıca sınıflandırma işlemi için RNN, GRU, LSTM, BiLSTM ve CNN olmak üzere farklı yapay sinir ağı yöntemleri kullanılmıştır.

### 7.4.1. İki Sınıflı Trafik Verisi Üzerinde Alana Özel Kelime Temsili ve Normalizasyon İşlemlerinin Sınıflandırma Başarımına Etkisi

Bu bölümde ilk olarak iki sınıflı trafik veri seti üzerinde normalizasyon işlemlerinin ve alana özel kelime temsili kullanımının sağladığı katkılar değerlendirilmektedir.




Şekil 7.10. İki sınıflı trafik verisi hata oranları.

Şekil 7.10'ün sol tarafındaki bölümde de görüldüğü gibi normalizasyon işlemi özellikle kelime temsillerinin sınıflandırma verisindeki düşük boyutlu veriden elde edildiği durumda tüm sınıflandırma yöntemleri için orijinal verinin kullanılmasına kıyasla bir iyileşme sağlamaktadır. Ayrıca Çizelge 7.3'te de görüldüğü gibi sınıflandırma başarımındaki artış oranı en iyi durumda LSTM ile %3 oranındaki gerçekleşirken en kötü durumda CNN ile yalnızca %0,25 olarak gerçekleşmiştir. Bununla birlikte alana özel bir derlemden kelime temsillerinin elde edilmesi iki sınıflı veri seti üzerinde CNN haricindeki tüm modeller için normalizasyon işlemine kıyasla çok daha iyi bir performans artışı sağlamıştır. LSTM modelinde sınıflandırma verisi kullanılarak elde edilen kelime temsilleri üzerinde normalizasyon işlemi %3'lük iyileşme sağlarken alana özel veriden kelime temsillerinin elde edilmesi, sınıflandırma başarımında %8,9'luk bir iyileşme sağlamaktadır. Buna ilave olarak alana özel bir kelime temsili kullanırken sınıflandırma verisi üzerinde normalizasyon işlemi yapmak LSTM ve GRU modelleri için pozitif bir katkı sağlamazken RNN modeli için önerilen normalizasyon ve manuel normalizasyon yaklaşımları sırasıyla %1,65 ve %2,65'lik bir iyileşme sağlamaktadır. Ayrıca alana özel kelime temsilleri kullanılırken normalizasyon işleminin BiLSTM ve CNN modellerindeki sağladığı katkı son derece kısıtlı düzeydedir. Bunun yanı sıra Çizelge 7.4 ve Çizelge 7.5'teki sonuçlar kıyaslandığında en yüksek başarım oranının orijinal sınıflandırma verisi ile alana özel

kelime temsillerinin kullanılarak LSTM modeli ile elde edildiği görülmektedir. Burada LSTM modelinin başarımı %96,15 olarak gerçekleşmiştir. Bu model üzerinde önerilen normalizasyon işlemi ve manuel normalizasyon işlemi ile elde edilen sınıflandırma başarımları ise en yüksek skorun çok az altında sırasıyla %95,8 ve 96,05'tir. Dolayısıyla iki sınıflı veri seti için etiketsiz verilerden elde edilen alana özel kelime temsillerinin kullanıldığı durumda normalizasyon işlemi yapmak herhangi bir pozitif katkı sağlamamaktadır. Bu ilave olarak büyük boyutlu alana özel verileri kullanarak kelime temsillerini elde etmek küçük boyutlu sınıflandırma verisinden kelime temsillerini elde etmeye kıyasla çok daha iyi bir sınıflandırma başarımı sağlamaktadır. Bu veriler ve Türkçe gibi dillerde ki normalizasyon işleminin zorluğu dikkate alındığında alana özel kelime temsili kullanımı SMM'leri normalize etmeye kıyasla daha akılcı bir çözümdür.

Çizelge 7.3. İki sınıflı trafik verisi hata oranları tablosu.

| | Sınıflandırma Verisi | | | Alana Özel Etiketsiz Veri | | |
|---|---|---|---|---|---|---|
| | Orijinal | Önerilen Normalizasyon | Manuel Normalizasyon | Orijinal | Önerilen Normalizasyon | Manuel Normalizasyon |
| RNN | 16,4 | 14,55 | 16 | 9,9 | 8,25 | 7,25 |
| GRU | 13 | 10,05 | 10,55 | 4,8 | 5,35 | 5,2 |
| LSTM | 12,75 | 9,75 | 10,25 | 3,85 | 4,2 | 3,95 |
| Bi LSTM | 12 | 10 | 11 | 4,85 | 3,95 | 4,85 |
| CNN | 6,95 | 6,7 | 6,65 | 7,15 | 6,7 | 6,55 |

Ayrıca CNN modelinin sınıflandırma başarımı değerlendirildiğinde diğer modellerden çok daha farklı bir davranış sergilediği görülmektedir. Alana özel kelime temsili kullanılması diğer modeller için belirgin bir şekilde pozitif bir katkı sağlarken CNN modeli bu noktada diğer modellerden ayrışmış ve alana özel kelime temsili kullanımı pozitif bir katkı sağlamamıştır. Öte yandan Şekil 7.4'te görüldüğü gibi CNN modelinin hata oranları son derece doğrusal bir şekilde gerçekleşmiştir. Bunun yanı sıra sınıflandırma verisinden kelime temsillerinin elde edildiği durumda orijinal veriden elde edilen sınıflandırma başarımı ile en yakın sonucun %5,05 üzerinde %93,05'lik bir skor elde edilmiştir. Bunlara ilave olarak modelin gürültüye karşı olan direncinin

de diğer modellerden daha iyi olduğu görülmektedir. Normalizasyon işlemleri, sınıflandırma verisinden elde edilen kelime temsilleri kullanılarak yapılan testlerde yalnızca %0,25 ve %0,3'lük bir pozitif katkı sağlamıştır. Bu durumda alana özel büyük boyutlu kelime temsillerinin elde edilemediği durumlar için CNN modeli daha iyi sonuç sağlayabilir. Buna ilave olarak normalizasyon işleminin etkisinin son derece limitli olması nedeni ile küçük boyutlu kelime temsilleri üzerinde normalizasyon işlemi yapmadan doğrudan CNN modeli kullanılarak normalizasyonun getirdiği ilave hesaplama maliyeti ortadan kaldırılabilir.

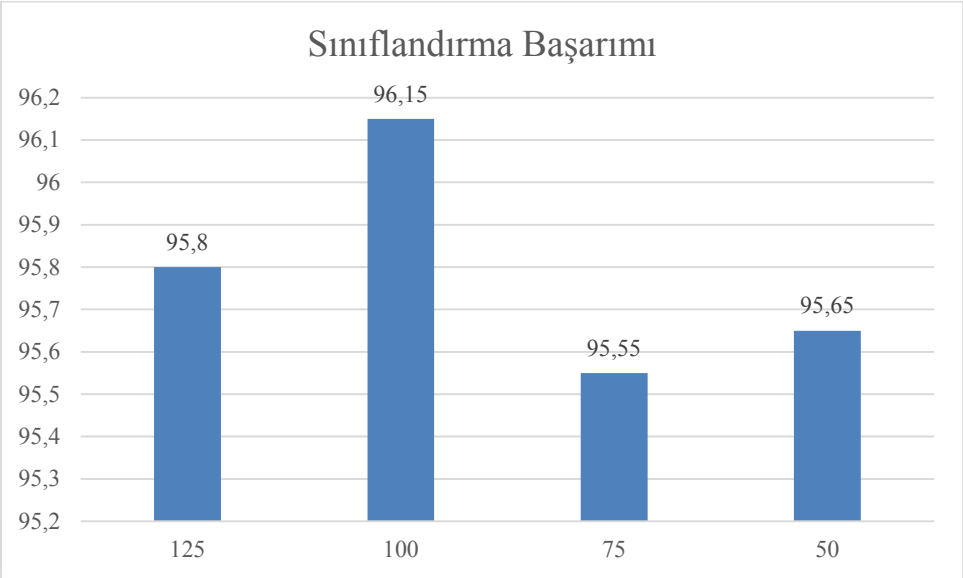
Çizelge 7.4. İki sınıflı veri setinde sınıflandırma verisi ile elde edilen kelime temsillerinin sınıflandırma başarımı.

| | Orijinal | Önerilen Normalizasyon | Manuel Normalizasyon |
|---|---|---|---|
| RNN | 83,6 | 85,45 | 84 |
| GRU | 87 | 89,95 | 89,45 |
| LSTM | 87,25 | 90,25 | 89,75 |
| Bi LSTM | 88 | 90 | 89 |
| CNN | 93,05 | 93,3 | **93,35** |

Çizelge 7.5. İki sınıflı veri setinde alana özel kelime temsillerinin sınıflandırma başarımı.

| | Orijinal | Önerilen Normalizasyon | Manuel Normalizasyon |
|---|---|---|---|
| RNN | 90,1 | 91,75 | 92,75 |
| GRU | 95,2 | 94,65 | 94,8 |
| LSTM | **96,15** | 95,8 | 96,05 |
| Bi LSTM | 95,15 | 96,05 | 95,15 |
| CNN | 92,85 | 93,3 | 93,45 |

### 7.4.2. Farklı Temsil Uzunluklarının Sınıflandırma Başarımına Etkisi




Şekil 7.11. Alana özel kelime temsillerinde kullanılan vektör uzunluğunun 2 sınıflı trafik verisi sınıflandırma başarımına etkisi.

Hiper parametrelerin optimizasyonu sınıflandırma performansını etkileyen önemli noktalardan bir tanesidir. Kullanılan kelime temsillerinin vektör boyutunun belirlenmesi önemli parametrelerden biridir. Vektör boyutundaki değişimin etkisini değerlendirmek için ikili sınıflandırma veri setindeki en yüksek sınıflandırma başarımını sağlayan alana özel kelime temsili ve LSTM sınıflandırma yöntemi farklı vektör boyutları ile test edilmiştir. Şekil 7.11'de farklı vektör boyutları ile eğitilmiş alana özel kelime temsilleri ile elde edilmiş sınıflandırma başarımları görülmektedir. 50, 75, 100 ve 125 vektör boyutları için sınıflandırma skorları birbirine oldukça yakındır. En kötü durumda vektör boyutu 75 için sınıflandırma başarımı %95,55 olarak gerçekleşirken en iyi durumda vektör boyutu 100 için %96,15 olarak gerçekleşmiştir. Geri kalan testlerde vekör boyutu 100 ile devam edilmiştir.

### 7.4.3. Algılayıcı Sayılarının ve Derin Mimari Kullanımının İki Sınıflı Veri Setinin Sınıflandırma Başarımına Etkisi

Kullanılan mimarideki katman sayısı ve her bir katmanda bulunan algılaycı sayısı da hem sınıflandırma başarımını hem de eğitim sürelerini etkilemektedirler. Bu

doğrultuda LSTM modeli üzerinde farklı algılayıcı sayısı ve farklı katman sayıları kullanılarak testler gerçekleştirildi. Özellikle daha derin bir mimari kullanmanın sınıflandırma başarımı üzerine olan etkisi değerlendirildi. Çizelge 7.6'da görüldüğü gibi sınıflandırma veri setindeki veri boyutunun çok büyük olmamasına bağlı olarak derin bir mimari kullanmak sınıflandırma başarımına pozitif bir katkı sağlamadı. 75, 100 ve 125 algılayıcı sayısı ile yapılan testlerde en yüksek başarım %96,15 ile 100 algılayıcı sayısına sahip olan tek LSTM katmanlı model ile elde edildi. 100 algılayıcı sayısına sahip 2 ve 3 LSTM katmanlı modellerde ise sınıflandırma başarımı sırasıyla %95,35 ve %95,6 olarak gerçekleşti. Tüm algılayıcı sayıları için ilave LSTM katmanının kullanılması sınıflandırma başarımını negatif yönde etkiledi.

Çizelge 7.6. İki sınıflı veri setinde farklı katman ve algılayıcı sayılarının LSTM başarımına etkisi.

| Katman Sayısı | Algılayıcı Sayısı | Sınıflandırma Başarımı |
|---|---|---|
| 1 | 75 | 95,6 |
| 2 | 75 | 95,45 |
| 1 | 100 | **96,15** |
| 2 | 100 | 95,35 |
| 3 | 100 | 95,6 |
| 1 | 125 | 95,6 |
| 2 | 125 | 95,2 |

### 7.4.4. CNN-LSTM Modelinin İki Sınıflı Trafik Verisi Üzerindeki Sınıflandırma Başarımı

CNN modeli sınıflandırma verisinden kelime temsillerinin oluşturulduğu yaklaşımda diğer modellere göre belirgin bir iyileşme sağlamaktadır. Öte yandan alana özel kelime temsillerinin kullanılması durumunda da LSTM modeli çok başarılı bir sınıflandırma performansı sergilemektedir. Bu iki modelin birleşiminden oluşacak bir modelin sınıflandırma başarımı üzerine etkisini test etmek için CNN-LSTM modeli oluşturulmuştur. Çizelge 7.7'deki sonuçlarda da görüldüğü gibi CNN-LSTM

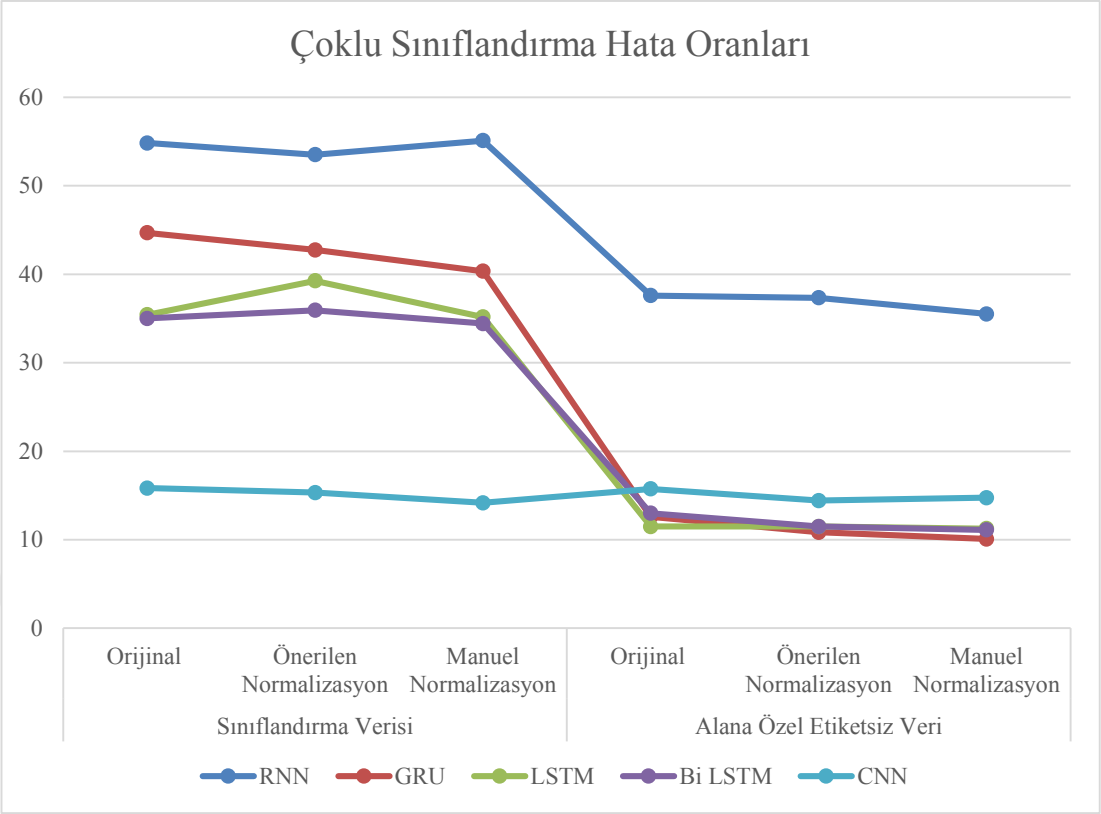
modelinde kelime temsillerinin sınıflandırma verisinden veya alana özel etiketsiz veriden elde edilmesi bir fark oluşturmamıştır. Her iki durumda da neredeyse aynı sonuçlar elde edilmiştir. Ayrıca CNN ve LSTM modellernin tek başlarına kullanıldıkları modellere görede daha düşük bir sınıflandırma başarımı elde edilmiştir.

Çizelge 7.7. İki sınıflı trafik verisi üzerinde CNN-LSTM modelinin sınıflandırma başarımı.

| | Sınıflandırma Verisi | Alana Özel Etiketsiz Veri |
|---|---|---|
| Orijinal | 89,90 | 89,90 |
| Önerilen Normalizasyon | 89,40 | 89,45 |
| Manuel Normalizasyon | 90,60 | 90,60 |

### 7.4.5. Çok Sınıflı Trafik Verisi Üzerinde Alana Özel Kelime Temsili ve Normalizasyon İşlemlerinin Sınıflandırma Başarımına Etkisi

İki sınıflı trafik veri seti üzerinde yapılan çalışmalar çok sınıflı trafik veri seti üzerinde tekrar edildi. Şekil 7.12'de de görüldüğü gibi CNN haricindeki tüm modellerin sınıflandırma verisinden kelime temsillerinin elde edildiği yaklaşımda oldukça zayıf bir sınıflandırma başarımı sağladığı görülmektedir. Çizelge 7.8'de görüldüğü gibi RNN modelinde hata oranları %55'lere ulaşmaktadır. Buna ilave olarak CNN ve GRU modeli haricindeki modeller için normalizasyon işlemleri tutarlı bir davranış sergilememektedir. CNN ile orijinal veri, önerilen normalizasyon ve manuel normalizasyon ile sırasıyla %15,83, %15,33 ve %14,17'lik hata oranları elde edilirken normalizasyon işleminin katkısı yine sırasıyla %0,5 ve %1,6 olarak gerçekleşmiştir. GRU modeli için ise hata oranları yine aynı sıra ile %44,67, %42,75 ve %40,33 olarak gerçekleşmiştir. Normalizasyon işleminin katkısı ise %1,92 ve %4,34 olarak gerçekleşmiştir.




Şekil 7.12. Çok sınıflı trafik verisi hata oranları.

Öte yandan alana özel kelime temsilleri kullanılarak gerçekleştirilen sınıflandırma işleminde tüm modeller için sınıflandırma başarımında bir artış sağlanmıştır. Bu noktada CNN modeli için elde edilen skor sınıflandırma verisinden kelime temsillerinin elde edildiği yaklaşıma oldukça yakındır. Bununla birlikte diğer tüm modellerde sınıflandırma başarımı oldukça keskin bir artış kaydetmiştir. Normalize edilmemiş orijinal veri ile yapılan sınıflandırma işlemlerinde başarım oranları RNN, GRU, LSTM ve Bi LSTM modelleri için sırasıyla %17,24, %32,08, %23,92 ve %22 oranlarında artış göstermiştir. İki sınıflı trafik veri setinde LSTM modeli ile elde edilen %8,9'luk artış oranı ile kıyaslandığında çok sınıflı veri seti üzerinde alana özel kelime temsillerinin kullanılmasının çok daha büyük bir katkı sağladığı açıkça görülmektedir. Benzer şekilde GRU modeli üzerinde sınıflandırma verileri ile normalizasyon yapılarak sağlanan %1,92 ve %4,34'lük başarımında çok çok ötesinde bir katkı sunmaktadır.

Ayrıca alana özel kelime temsillerine ilave olarak normalizasyon işleminde yapılması da tüm modellerde pozitif bir katkı sağlamaktadır. Fakat bu katkının oranı alana özel

kelime temsillerinin kullanılmasına kıyasla çok düşüktür. En iyi katkı oranının elde edildiği GRU modelinde bile önerilen normalizasyon işleminin ve manuel normalizasyon işleminin katkıları sırasıyla %1,75 ve %2,51 olarak gerçekleşmiştir. Ayrıca orjinal veri seti modeli ile önerilen ve manuel normalizasyon modelinin arasındaki sonuçların istatiksel olarak anlamlı olup olmadığını değerlendirmek için k-kat çapraz doğrulanmış eşli t-testi uygulanmıştır. Her iki testten elde edilen sonuçlarda (p-değeri < $10^{-4}$) istatiksel olarak anlamlıdır.

Bunlara ilave olarak CNN modeli iki sınıflı trafik veri setine oldukça benzeyen bir sınıflandırma performansı sergilemiştir. Sınıflandırma verileri kullanılarak elde edilen kelime temsillerinde orijinal veri ile yapılan sınıflandırma işleminde BiLSTM modeli ile elde edilen en yakın sonuçtan %19,17 daha iyi bir performans sağlamıştır. Yine iki sınıflı veri setinde olduğu gibi normalizasyon işlemi başarımı bir miktar artırmış olmakla birlikte bu artış oranı önerilen normalizasyon yaklaşımı için yalnızca %0,5 ve manuel normalizasyon işlemi %1,66 olarak gerçekleşmiştir.

Çizelge 7.8. Çok sınıflı veri setinde kelime temsillerinin ve normalizasyon işlemlerinin hata oranlarına etkisi.

| | Sınıflandırma Verisi | | | Alana Özel Etiketsiz Veri | | |
|---|---|---|---|---|---|---|
| | Orijinal | Önerilen Normalizasyon | Manuel Normalizasyon | Orijinal | Önerilen Normalizasyon | Manuel Normalizasyon |
| RNN | 54,83 | 53,5 | 55,09 | 37,59 | 37,34 | 35,5 |
| GRU | 44,67 | 42,75 | 40,33 | 12,59 | 10,84 | 10,08 |
| LSTM | 35,42 | 39,25 | 35,17 | 11,5 | 11,5 | 11,25 |
| Bi LSTM | 35 | 35,92 | 34,42 | 13 | 11,5 | 11,12 |
| CNN | 15,83 | 15,33 | 14,17 | 15,75 | 14,42 | 14,75 |

Çizelge 7.9 ve Çizelge 7.10'da da görüldüğü gibi sınıflandırma verisinden elde edilen kelime temsillerinde en iyi skor manuel normalizasyon yapılmış CNN modeli %85,83 olarak elde edilirken alana özel kelime temsillerinin kullanıldığı yaklaşımda en iyi skor yine manuel normalizasyon ve GRU modeli ile elde edilmiştir. Yukarıda da belirtildiği gibi normalizasyon işlemi sınıflandırma işlemine olumlu yönde bir katkı sağlamakla birlikte bu katkının etkisi alana özel kelime temsilinin kullanılmasına kıyasla oldukça düşük olduğu açıkça görülebilmektedir.

Çizelge 7.9. Çok sınıflı veri setinde sınıflandırma verisi ile elde edilen kelime temsillerinin sınıflandırma başarımı.

| | Orijinal | Önerilen Normalizasyon | Manuel |
|---|---|---|---|
| RNN | 45,17 | 46,5 | 44,91 |
| GRU | 55,33 | 57,25 | 59,67 |
| LSTM | 64,58 | 60,75 | 64,83 |
| Bi LSTM | 65 | 64,08 | 65,58 |
| CNN | 84,17 | 84,67 | **85,83** |

Çizelge 7.10. Çok sınıflı veri setinde alana özel kelime temsillerinin sınıflandırma başarımı.

| | Orijinal | Önerilen Normalizasyon | Manuel Normalizasyon |
|---|---|---|---|
| RNN | 62,41 | 62,66 | 64,5 |
| GRU | 87,41 | 89,16 | **89,92** |
| LSTM | 88,5 | 88,5 | 88,75 |
| Bi LSTM | 87 | 88,5 | 88,88 |
| CNN | 84,25 | 85,58 | 85,25 |

### 7.4.6. GRU Modeli Alt Sınıflar Bazında Başarım Değerlendirmesi

Çizelge 7.11'de GRU modeli ile elde edilen sınıflandırma işlmemi sonuçlarının alt sınıflar bazında dağılımları görülmektedir. Sınıflandırma verisinden kelime temsillerinin elde edildiği testlerde yol çalışması sınıfının F1 skorunun 0 olduğu görülmektedir. Yani hiçbir durumda yol çalışması durumu başarılı bir şekilde tespit edilememiştir. Öte yandan alana özel kelime temsillerinin kullanılmasıyla birlikte yol çalışması durumunun sınıflandırma işleminden orjinal veri, önerilen normalizasyon ve manuel normalizasyon verileri için sırasıyla %84,42, %87.68 ve %88,12'lik F1 skorları elde edilmiştir. Benzer şekilde kaza sınıfı da sınıflandırma verisinden son derece düşük bir oranda tespit edilebilirken alana özel kelime temsillerinin

kullanılmasıyla başarım oranları çok ciddi bir atış kaydetmiştir. Bununla birlikte her durumda en yüksek F1 skoruna sahip sınıf hava durumu olarak gerçekleşmektedir.

Çizelge 7.11. GRU modeli alt sınıf bazında sınıflandırma değerleri.

| | | Sınıflandırma Verisi | | | Etiketsiz Veri | | |
|---|---|---|---|---|---|---|---|
| | | Orijinal | Önerilen Normalizasyon | Manuel Normalizasyon | Orijinal | Önerilen Normalizasyon | Manuel Normalizasyon |
| Kesinlik (%) | Hava Durumu | 78,10 | 74,14 | 95,09 | 96,38 | 97,49 | 97,14 |
| | Harici Olay | 39,20 | 41,05 | 38,56 | 74,49 | 77,84 | 81,44 |
| | Yol Çalışması | 0,00 | 0,00 | 0,00 | 88,42 | 89,90 | 90,82 |
| | Genel | 40,58 | 50,48 | 58,54 | 92,47 | 93,48 | 88,00 |
| | Kaza | 0,00 | 40,00 | 18,18 | 91,67 | 93,00 | 92,86 |
| | Hatalı Sürücü | 66,07 | 57,14 | 64,17 | 90,37 | 90,23 | 92,42 |
| | Araç Arızası | 38,30 | 17,24 | 44,44 | 80,41 | 81,19 | 81,13 |
| | Trafik Olmayan | 57,56 | 69,92 | 60,07 | 84,43 | 88,61 | 93,23 |
| Hassasiyet (%) | Hava Durumu | 89,13 | 93,48 | 91,30 | 96,38 | 98,55 | 98,55 |
| | Harici Olay | 85,71 | 85,71 | 85,16 | 80,22 | 82,97 | 86,81 |
| | Yol Çalışması | 0,00 | 0,00 | 0,00 | 80,77 | 85,58 | 85,58 |
| | Genel | 28,00 | 53,00 | 24,00 | 86,00 | 86,00 | 88,00 |
| | Kaza | 0,00 | 6,19 | 2,06 | 90,72 | 95,88 | 93,81 |
| | Hatalı Sürücü | 28,03 | 27,27 | 58,33 | 92,42 | 90,91 | 92,42 |
| | Araç Arızası | 17,82 | 4,95 | 27,72 | 77,23 | 81,19 | 85,15 |
| | Trafik Olmayan | 86,06 | 83,17 | 85,58 | 86,06 | 85,10 | 83,17 |
| F1-Skoru (%) | Hava Durumu | 83,25 | 82,69 | 93,16 | 96,38 | 98,02 | 97,84 |
| | Harici Olay | 53,79 | 55,52 | 53,08 | 77,25 | 80,32 | 84,04 |
| | Yol Çalışması | 0,00 | 0,00 | 0,00 | 84,42 | 87,68 | 88,12 |
| | Genel | 33,14 | 51,71 | 34,04 | 89,12 | 89,58 | 88,00 |
| | Kaza | 0,00 | 10,71 | 3,70 | 91,19 | 94,42 | 93,33 |
| | Hatalı Sürücü | 39,36 | 36,92 | 61,11 | 91,39 | 90,57 | 92,42 |
| | Araç Arızası | 24,32 | 7,69 | 34,15 | 78,79 | 81,19 | 83,09 |
| | Trafik Olmayan | 68,98 | 75,97 | 70,59 | 85,24 | 86,82 | 87,91 |

### 7.4.7. Algılayıcı Sayılarının ve Derin Mimari Kullanımının Çok Sınıflı Veri Setinin Sınıflandırma Başarımına Etkisi

İki sınıflı veri setinde olduğu gibi farklı algılayıcı sayıları ve daha derin bir mimari kullanmanın sınıflandırma performansını nasıl etkilediği test edildi. Sonuçlar en yüksek başarımı sağlayan GRU modeli üzerinde değerlendirildi. Çizelge 7.12'de görüldüğü gibi her biri 16 algılayıcıdan oluşan 3 katmanlı mimaride 2 katmanlı mimariye kıyasla daha iyi sonuç elde edildi. Ayrıca 2 katman ve 32 algılayıcıdan oluşan modelde tek katmana kıyasla çok düşük miktara bir artış sağlanmış olmakla birlikte eğitim sürelerinin göreceli olarak fazla maliyet getirmesi nedeni ile tek katmanlı mimari tercih edildi.

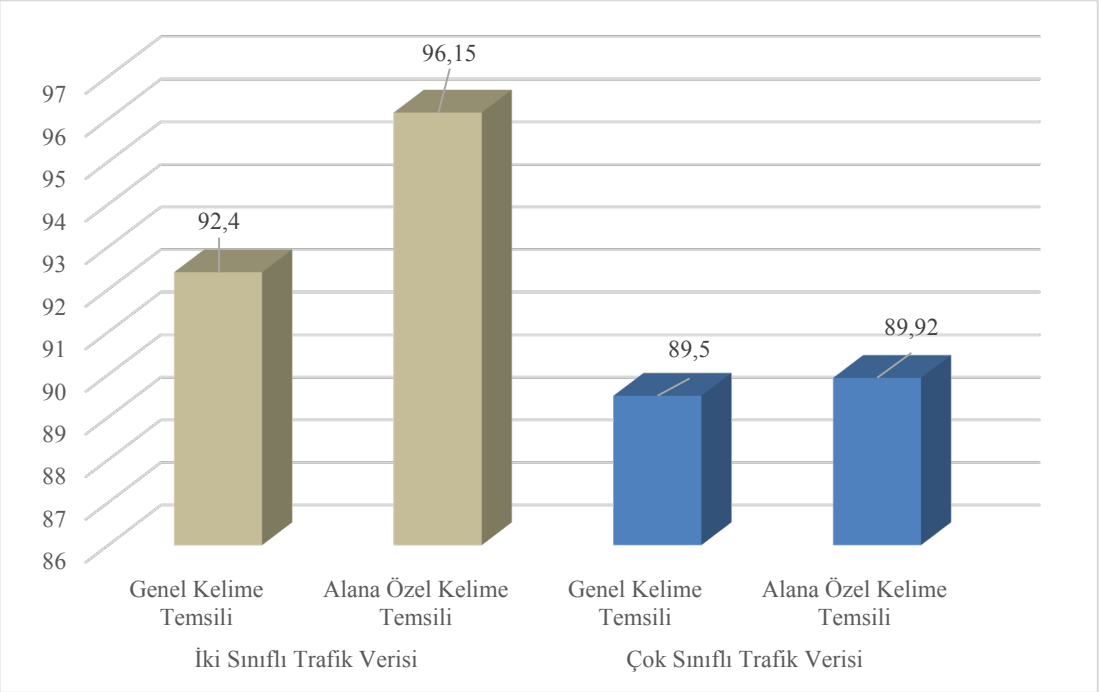
Çizelge 7.12. Çok sınıflı veri setinde farklı katman ve algılayıcı sayılarının GRU başarımına etkisi.

| Katman Sayısı | Algılayıcı Sayısı | Sınıflandırma Başarımı |
|---|---|---|
| 1 | 16 | 87,5 |
| 2 | 16 | 72,33 |
| 3 | 16 | 78,25 |
| 1 | 32 | 89,92 |
| 2 | 32 | **89,97** |
| 3 | 32 | 88 |
| 1 | 40 | 89,5 |
| 2 | 40 | 88,83 |

### 7.4.8. Genel veya Alana Özel Kelime Temsili Kullanılmasının Sınıflandırma Başarımlarına Etkisi

Bu testlere ilave olarak alana özel kelime temsili kullanılması veya genel bir kelime temsili kullanılmasının sınıflandırma performansını nasıl etkilediğini değerlendirmek adına Wikipedia ve Hürriyet gazetesi arşivlerinden elde edilmiş yine 100 vektör boyutlu bir kelime temsili kullanarak sonuçları her iki sınıflandırma probleminde en

iyi sonuçları sağlayan modeller üzerinde kıyaslandı. Her iki veri setinde de alana özel kelime temsili kullanılması daha iyi sonuç sağlamaktadır. Bununla birlikte alana özel kelime temsili kullanılmasının katkısı iki sınıflı veri setinde çok sınıflı veri setine kıyasla çok daha fazladır. Şekil 7.13'de görüldüğü gibi alana özel kelime temsilinin pozitif katkısı iki sınıflı veri seti için %3,75 iken çok sınıflı veri seti için %0,42 olarak gerçekleşmiştir.




Şekil 7.13. Genel kelime temsili ve alana özel kelime temsili kullanımının sınıflandırma başarımına etkisi.

# BÖLÜM 8

# SONUÇ VE ÖNERİLER

Trafikle ilgili olayları makul bir doğruluk oranı içerisinde, maliyet etkin bir çözümle, geniş bir kapsama alanı içerisinde tespit edebilecek bir yöntem özellikle büyük şehirlerdeki trafik sorunun yönetilmesi açısından oldukça önemlidir. Bu tez çalışmasında SMM'lerden elde edilen metin verilerinin sınıflandırılmasına ve trafikle ilgili olanların tespit edilerek araç arızası, kaza, hatalı sürücü, hava durumu veya yol çalışması gibi olayın alt türünün belirlenmesine dayanan bir yöntem sunulmaktadır. Öte yandan SMM'ler yüksek miktardaki yüksek gürültü içerikleri nedeniyle metin madenciliği ve DDİ uygulamaları için oldukça zorlu görevlerdir. Bu tez çalışmasında SMM'lerdeki gürültü problemini azaltmak adına öncelikle Türkçe normalizasyon mimarilerinde kullanılan diyakritik, aksan ve ünlü harf resterasyon modülleri ile yazım denetimi modülü Word2vec tabanlı belirsizlik giderme modülü ile genişletilmiştir. Ayrıca bütünüyle kaskad bir mimari yerine hibrit bir yapı tercih edilmiştir. Önerilen normalizasyon yaklaşımı ile trafik veri seti normalize edilmiş ve güncel tekniklere kıyasla %25,95'lik bağıl hata azaltım oranına denk %10,41'lik bir iyileşme sağlanmıştır. Trafik veri seti için normalizasyon başarımı %70,29 olarak gerçekleşmiştir.

Literatürdeki birçok çalışma normalizasyon işleminin sınıflandırma performansına olumlu katkı sağladığını ortaya koymaktadır. Bununla birlikte normalizasyon işlemi Türkçe gibi diller için birçok zorluğa sahiptir. Bunun yanı sıra hataya meyilli olmaları, sosyal medyadaki yazım dilinin son derece canlı olması ve sürekli yeni yazım stillerinin ortaya çıkması, normalizasyon başarımının alana ve veri sertine göre farklılık göstermesi ve getirdikleri ilave hesaplama maliyetleri gibi nedenlerden dolayı birçok dezavantaja da sahiptir. Bununla birlikte son yıllarda alana özel kelime temsillerinin kullanılması kullanıcılar tarafından türetilen gürültülü metinlerin sınıflandırılmasında oldukça başarılı sonuçlar sağlamaktadır. Bu doğrultuda trafik

alanına özel yaklaşık 1,5 M etiketsiz tweetten oluşan bir derlem hazırılanmıştır. Trafik derlemini oluşturmak için Python dilinde geliştirilmiş olan Scrapy aracı kullanılmıştır. Bu derlem kullanılarak Word2vec yöntemi ile kelime temsilleri elde edilerek trafik verilerinin sınıflandırılması için kullanılmıştır.

Trafikle ilgili olayları tespit edebilmek için ise Twitter REST API kullanılarak anahtar kelime temeline dayalı olarak tweetler toplanmıştır. Toplanan tweetlerden 2 veri seti oluşturulmuştur. Birinci veri seti trafikle ilgili olan ve olmayan tweetler olmak üzere iki sınıftan oluşmaktadır ve her bir sınıftan 1 K olmak üzere toplam 2 K tweet mevcuttur. İkinci veri seti de toplam 1,2 K tweetten oluşmakta ve trafikle ilgili kaza, yol çalışması ve hatalı sürücü gibi alt durumları da içeren sekiz farklı sınıftan oluşmaktadır. Hazırlanan veri setleri üzerinde normalizasyon işleminin ve alana özel kelime temsili kullanılmasının sağladığı katkılar değerlendirilmiştir. Sınıflandırma işleminde tweet içerisindeki her bir kelime bir vektörel temsil ile ifade edilmiştir. Kelime temsilleri sırasıyla etiketli veriden ve etiketsiz alana özel derlemden elde edilmiştir. 2 sınıflı veri seti üzerinde gerçekleştirilen testlerde sınıflandırma verisinden kelime temsillerinin elde edildiği durum için normalizasyon yapmak tüm modellerde sınıflandırma performansını artırmıştır. Bu koşullarda önerilen normalizasyon yaklaşımı ile sınıflandırma başarımındaki en yüksek artış %3'lük bir değerle LSTM modelinde gerçekleşmiştir. Öte yandan etiketli kelime temsillerinin kullanıldığı durumda en yüksek sınıflandırma başarımı CNN modeli ile elde edilmiştir. Normalize edilmemiş orjinal veri ile yapılan testlerde CNN modeli %93,05'lik sınıflandırma başarımı sağlamıştır. Bu başarım değeri en yakın modelin %5,05 üzerindedir. Bununla birlikte CNN modeli üzerinde önerilen normalizasyon yaklaşımının sağladığı katkı yalnızca %0,25'tir. Alana özel kelime temsillerinin kullanılması durumunda ise CNN haricindeki tüm modellerde sınıflandırma başarımı belirgin bir şekilde artış göstermiştir. 2 sınıflı veri seti üzerinde en iyi sınıflandırma skoru %96,15'lik değerle alana özel kelime temsili ve orjinal veri kullanılarak LSTM modeli ile elde edilmiştir. LSTM modelinde etiketli veriden elde edilen kelime temsilleri kullanılırken normalizasyon işlemi yapmak %3'lük katkı sağlarken, LSTM modelinde alana özel kelime temsili kullanmak %8,9'luk çok daha yüksek bir katkı sağlamıştır. Ayrıca yine LSTM modelinde alana özel kelime temsili kullanırken normalizasyon işlemi yapmak sınıflandırma performansına olumlu bir katkı sağlamamıştır. Çok sınıflı veri setleri

üzerinde yapılan testler incelendiğinde CNN haricindeki tüm modellerde etiketli veriden elde edilen kelime temsillerinin kullanımı sınıflandırma başarımını çok ciddi bir şekilde düşürmektedir. Ayrıca etiketli veriden elde kelime temsilleri kullanılırken normalizasyon işlemi CNN ve GRU modelleri için başarımı artırıırken diğer modellerde tutarlı bir davranış görülememiştir. Bununla birlikte alana özel kelime temsili kullanmak tüm modellerin performansını yukarı taşımıştır. Alana özel kelime temsili kullanımıyla elde edilen sınıflandırma performansındaki artış oranı 2 sınıflı veri setindeki %8,9'luk başarımın çok daha üzerinde %32,08 olarak elde edilmiştir. Alana özel veriye ilave olarak normalizasyon yapmak tüm modellerde sınıflandırma başarımını bir miktar yukarı taşımakla birlikte bu oran alana özel kelime temsili kullanımıyla elde edilen katkının çok çok altındadır. Orjinal veri ve alana özel kelime temsili kullanılırken en yüksek başarım %88,5 ile 2 sınıflı veri setine benzer şekilde LSTM modeli elde edilmiştir. Öte yandan alana özel kelime temsili kullanılırken önerilen normalizasyon yaklaşımı ve manuel normalizasyon ile en iyi skorlar GRU modelinde elde edilmiştir. GRU için başarım oranları sırasıyla %89,16 ve %89,92 olarak gerçekleşmiştir. Bu durumda LSTM modelinin orjinal veriden elde edilen skorlarıyla kıyaslandığında önerilen normalizasyon işleminin katkısı %0,66 olurken, manuel normalizasyon işleminin sağladığı katkı %1,42 olarak gerçekleşmiştir.

Bir diğer önemli noktada küçük boyutlu etiketli veriden kelime temsillerinin elde edildiği durumlarda hem iki sınıflı hem çok sınıflı veri seti için CNN modeli diğer modellere kıyasla belirgin şekilde çok daha iyi performans sağlamaktadır. Ayrıca her durumda normalizasyon işlemi CNN modelinin sınıflandırma başarımını artırmakla birlikte bu başarım artışı oldukça küçüktür. Orjinal veri ile yapılan testlerde iki sınıflı veri setinde en yakın modele kıyasla %5,05 daha iyi bir sınıflandırma başarımı sağlarken, çok sınıflı veri seti üzerinde en yakın modele göre %19,17'lik daha iyi bir skor sağlamıştır.

Sonuç olarak normalizasyon işlemi birçok modelde sınıflandırma başarımını yukarı taşımakla birlikte bu etki alana özel kelime temsili kullanımına kıyasla çok daha düşüktür. Bu doğrultuda Türkçe gibi diller için normalizasyon işleminin zorlukları ve getirdiği ilave maliyetler dikkate alındığında normalizasyon işlemi yerine alana özel kelime temsilleri kullanmak çok daha akılcı bir çözümdür. Hem iki sınıflı hem çok

sınıflı veri setinde alana özel kelime temsilleri ve orjinal veri kullanılırken en yüksek skorlar LSTM modeli ile elde edilmektedir. Bunun yanı sıra alana özel kelime temsili kullanmak CNN performansında belirgin bir değişikliğe neden olmamaktadır. Öte yandan küçük boyutlu etiketli veriden kelime temsilleri elde edilirken CNN modeli diğer modellere kıyasla belirgin bir şekilde çok daha iyi sonuçlar sağlamaktadır. Normalizasyon işlemi CNN modelinin sınıflandırma başarımını arttırsa da bu artış göz ardı edilebilecek kadar küçüktür. Nihai olarak alana özel kelime temsillerinin mevcudiyeti durumunda normalizasyon yapmadan LSTM modeli kullanılabileceği gibi bu tip bir verinin olmaması durumunda CNN modeli yine normalizasyon yapmadan sınıflandırma başarımından oldukça küçük bir tavizle başarılı sonuçlar elde edilebilmektedir.

## KAYNAKLAR


1. Gutierrez, C., Figuerias, P., Oliveira, P., Costa, R. and Jardim-Goncalves, R. "Twitter mining for traffic events detection", ***In Science and Information Conference (SAI)***, IEEE, 371-378 (2015).

2. Gu, Y., Qian, Z. S. and Chen, F., "From Twitter to detector: Real-time traffic incident detection using social media data", ***Transportation research part C: emerging technologies***, 67: 321-342 (2016).

3. Fu, K., Lu, C. T., Nune, R. and Tao, J. X., "Steds: Social Media Based Transportation Event Detection with Text Summarization", ***In ITSC***, 1952-1957 (2015).

4. Leduc, G., "Road traffic data: Collection methods and applications", ***Working Papers on Energy, Transport and Climate Change***, 1(55) (2008).

5. Xu, S., Li, S. and Wen, R., "Sensing and detecting traffic events using geosocial media data: A review", ***Computers, Environment and Urban Systems***, (2018).

6. Doan, A., Ramakrishnan, R. and Halevy, A. Y., "Crowdsourcing systems on the world-wide web", ***Communications of the ACM***, 54(4): 86-96 (2011).

7. Hasan, M., Orgun, M. A. and Schwitter, R., "Real-time event detection from the Twitter data stream using the TwitterNews+ Framework", ***Information Processing & Management*** (2018).

8. Vilares, D., Alonso, M. A. and Gómez-Rodríguez, C., "Supervised sentiment analysis in multilingual environments", ***Information Processing & Management***, 53(3): 595-607 (2017).

9. Wang, X., Gerber, M. S. and Brown, D. E., "Automatic crime prediction using events extracted from twitter posts", ***In International conference on social computing, behavioral-cultural modeling, and prediction,*** Springer, Berlin, Heidelberg, 231-238 (2012).

10. Giridhar, P., Amin, M. T., Abdelzaher, T., Wang, D., Kaplan, L., George, J. and Ganti, R., "Clarisense+: An enhanced traffic anomaly explanation service using social network feeds", ***Pervasive and Mobile Computing***, 33: 140-155 (2016).

11. Saloot, M. A., Idris, N. and Mahmud, R., "An architecture for Malay Tweet normalization", ***Information Processing & Management***, 50(5): 621-633 (2014).

12. Eryiğit, G. and Torunoğlu-Selamet, D. İ. L. A. R. A., "Social media text normalization for Turkish", ***Natural Language Engineering***, 23(6): 835-875 (2017).

13. Adali, K. and Eryiğit, G., "Vowel and diacritic restoration for social media texts", ***In Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)***, 53-61 (2014).

14. Systematics, C., "Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation", ***Final Report, Texas Transportation Institute***, http://ops. fhwa. dot. gov/congestion_report_04/index. htm (2005).

15. Zheng, X., Chen, W., Wang, P., Shen, D., Chen, S., Wang, X. and Yang, L., "Big data for social transportation", ***IEEE Transactions on Intelligent Transportation Systems***, 17(3): 620-630 (2016).

16. Metaxas, P. T. and Mustafaraj, E., "Social media and the elections", ***Science***, 338(6106): 472-473 (2012).

17. Sakaki, T., Okazaki, M. and Matsuo, Y., "Earthquake shakes Twitter users: real-time event detection by social sensors", ***In Proceedings of the 19th international conference on World wide web***, ACM, 851-860 (2010).

18. Kryvasheyeu, Y., Chen, H., Obradovich, N., Moro, E., Van Hentenryck, P., Fowler, J. and Cebrian, M., "Rapid assessment of disaster damage using social media activity", ***Science advances***, 2(3), e1500779 (2016).

19. Watanabe, K., Ochi, M., Okabe, M. and Onai, R. Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs", ***In Proceedings of the 20th ACM international conference on Information and knowledge management***, ACM, 2541-2544 (2011).

20. D'Andrea, E., Ducange, P., Lazzerini, B. and Marcelloni, F., "Real-time detection of traffic from twitter stream analysis", ***IEEE transactions on intelligent transportation systems***, 16(4): 2269-2283 (2015).

21. Atefeh, F. and Khreich, W., "A survey of techniques for event detection in twitter", ***Computational Intelligence***, 31(1): 132-164 (2015).

22. Parikh, R. and Karlapalem, K. "Et: events from tweets", ***In Proceedings of the 22nd international conference on world wide web***, ACM, 613-620 (2013).

23. Albuquerque, F. C., Casanova, M. A., Lopes, H., Redlich, L. R., de Macedo, J. A. F., Lemos, M. and Renso, C., "A methodology for traffic-related Twitter messages interpretation", ***Computers in Industry***, 78: 57-69 (2016).

24. Zhang, Z., He, Q., Gao, J. and Ni, M. "A deep learning approach for detecting traffic accidents from social media data", ***Transportation research part C: emerging technologies***, 86: 580-596 (2018).

25. Wongcharoen, S. and Senivongse, T., "Twitter analysis of road traffic congestion severity estimation", ***In Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference on***, IEEE, 1-6 (2016).

26. Fu, K., Lu, C. T., Nune, R. and Tao, J. X., "Steds: Social Media Based Transportation Event Detection with Text Summarization", ***In ITSC***, 1952-1957 (2015).

27. Tang, D., Qin, B. and Liu, T., "Document modeling with gated recurrent neural network for sentiment classification", ***In Proceedings of the 2015 conference on empirical methods in natural language processing***,1422-1432 (2015).

28. Severyn, A., and Moschitti, A., "Twitter sentiment analysis with deep convolutional neural networks", ***In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval***, 959-962 (2015).

29. Zhou, C., Sun, C., Liu, Z. and Lau, F., "A C-LSTM neural network for text classification", arXiv preprint arXiv:1511.08630 (2015).

30. Wang, W. E. N. T. I. N. G., "Mining adverse drug reaction mentions in twitter with word embeddings", ***In Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing***, (2016).

31. Sarma, P. K., Liang, Y. and Sethares, W. A., "Domain Adapted Word Embeddings for Improved Sentiment Classification", arXiv preprint arXiv:1805.04576 (2018).

32. Jiang, Z., Li, L., Huang, D. and Jin, L., "Training word embeddings for deep learning in biomedical text mining tasks", ***In Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on***, IEEE, 625-628 (2015).

33. Major, V., Surkis, A. and Aphinyanaphongs, Y., "Utility of general and specific word embeddings for classifying translational stages of research", arXiv preprint arXiv:1705.06262 (2017).

34. Nguyen, D. T., Al-Mannai, K., Joty, S. R., Sajjad, H., Imran, M. and Mitra, P., "Robust Classification of Crisis-Related Data on Social Networks Using Convolutional Neural Networks", ***In ICWSM,*** 632-635 (2017).

35. Datta, D., Brashers, V., Owen, J., White, C. and Barnes, L. E., "A Deep Learning Methodology for Semantic Utterance Classification in Virtual Human Dialogue Systems", ***In International Conference on Intelligent Virtual Agents***, Springer, Cham, 451-455 (2016).

36. Hankamer, J., "Morphological parsing and the lexicon", ***In Lexical representation and process***, MIT Press, 392-408 (1989).

37. Aw, A., Zhang, M., Xiao, J. and Su, J., “A phrase-based statistical model for SMS text normalization”, ***In Proceedings of the COLING/ACL on main conference poster sessions***, Stroudsburg, PA, USA: Association for Computational Linguistics, 33–40 (2006).

38. Cook, P. and Stevenson, S., "An unsupervised model for text message normalization", ***Proceedings of the workshop on computational approaches to linguistic creativity***, Association for Computational Linguistics (2009).

39. Choudhury, M., Saraf, R., Jain, V., Mukherjee, A., Sarkar, S. and Basu, A., “Investigation and modeling of the structure of texting language”, ***International Journal of Document Analysis and Recognition (IJDAR)***, 10(3-4): 157-174 (2007).

40. Kaufmann, M. and Kalita, J., “Syntactic normalization of twitter messages”, ***In International conference on natural language processing***, Kharagpur, India (2010).

41. Clark, E. and Araki, K., “Text normalization in social media: progress, problems and applications for a pre-processing system of casual English”, ***Procedia-Social and Behavioral Sciences***, 27: 2-11 (2011).

42. Yang, Y. and Eisenstein, J., “A log-linear model for unsupervised text normalization”, ***In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing***, 61-72 (2013).

43. Sridhar, V. K. R., “Unsupervised text normalization using distributed representations of words and phrases”, ***In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing***, 8-16 (2015).

44. Akhtar, M. S., Sikdar, U. K. and Ekbal, A., “IITP: Hybrid approach for text normalization in Twitter”, ***In Proceedings of the Workshop on Noisy User-generated Text,*** 106-110 (2015).

45. Xu, K., Xia, Y. and Lee, C. H., “Tweet normalization with syllables”, ***In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)***, 920-928 (2015).

46. Mezhar, A., Ramdani, M. and El Mzabi, A., “A novel weakly supervised approach for casual English normalization”, ***In Intelligent Systems: Theories and Applications (SITA), 2016 11th International Conference on***, IEEE, 1-6 (2016).

47. Ansari, S. A., Zafar, U. and Karim, A., “Improving Text Normalization by Optimizing Nearest Neighbor Matching”, *arXiv preprint arXiv:1712.09518*, (2017).

48. Rehan, P., Kumar, M. and Singh, S., "A Modular Approach for Social Media Text Normalization", ***In Information and Decision Sciences***, Springer, Singapore,187-195 (2018).

49. De Clercq, O., Schulz, S., Desmet, B., Lefever, E. and Hoste, V., "Normalization of Dutch user-generated content", ***In Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013***, 179-188 (2013).

50. Cotelo, J. M., Cruz, F. L., Troyano, J. A. and Ortega, F. J., "A modular approach for lexical normalization applied to Spanish tweets", ***Expert Systems with Applications***, 42(10): 4743-4754 (2015).

51. Ruiz, P., Cuadros, M. and Etchegoyhen, T., "Lexical Normalization of Spanish Tweets with Rule-Based Components and Language Models", ***Procesamiento del Lenguaje Natural***, 52 (2014).

52. Nguyen, V. H., Nguyen, H. T. and Snasel, V., "Normalization of vietnamese tweets on twitter", ***In Intelligent Data Analysis and Applications***, 179-189 (2015).

53. Qian, T., Zhang, Y., Zhang, M., Ren, Y. and Ji, D., "A transition-based model for joint segmentation, pos-tagging and normalization", ***In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing***, 1837-1846 (2015).

54. Akın, A. A. and Akın, M. D., "Zemberek, an open source nlp framework for turkic languages", ***Structure***, 10: 1-5 (2007).

55. Tür, G., "A statistical information extraction system for Turkish", Ph.D. Thesis, ***Bilkent University Graduate School of Engineering and Science***, Ankara (2000).

56. Yuret, D. and De La Maza, M., "The greedy prepend algorithm for decision list induction", ***In International Symposium on Computer and Information Sciences***, Springer, Berlin, Heidelberg, 37-46 (2006).

57. Alpkoçak, A. and Ceylan, M., "Effects of diacritics on Turkish information retrieval", ***Turkish Journal of Electrical Engineering & Computer Sciences***, 20(5): 787-804 (2012).

58. Arslan, A., "DeASCIIfication approach to handle diacritics in Turkish information retrieval", ***Information Processing & Management***, 52(2): 326-339 (2016).

59. Al-Smadi, M., Al-Ayyoub, M., Jararweh, Y. and Qawasmeh, O., "Enhancing Aspect-Based Sentiment Analysis of Arabic Hotels' reviews using morphological, syntactic and semantic features", ***Information Processing & Management*** (2018).

60. Azmi, A. M. and Almajed, R. S., “A survey of automatic Arabic diacritization techniques”, ***Natural Language Engineering***, 21(3): 477-495 (2015).

61. Al-Anzi, F. S. and AbuZeina, D., “Beyond vector space model for hierarchical Arabic text classification: A Markov chain approach”, ***Information Processing & Management***, 54(1): 105-115 (2018).

62. Šantić, N., Šnajder, J., and Bašić, B. D., “Automatic diacritics restoration in croatian texts”, ***INFuture2009: Digital Resources and Knowledge Sharing***, 309-318 (2009).

63. Do, T. N. D., Nguyen, D. B., Mac, D. K. and Tran, D. D., “Machine translation approach for vietnamese diacritic restoration”, ***In Asian Language Processing (IALP), 2013 International Conference on*** IEEE, 103-106 (2013).

64. Grozea, C., “Experiments and results with diacritics restoration in Romanian”, ***In International Conference on Text, Speech and Dialogue,*** Springer, Berlin, Heidelberg, 199-206 (2012).

65. Öztürk, M. B., “Türkçede morfolojik analiz yapan bir sistemin morfolojik türetme için kullanılması”, Yüksek Lisans Tezi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü***, Ankara (2016).

66. Şahin, H., “Türkçe’de ön ek”, ***Uludağ Üniversitesi Fen-Edebiyat Fakültesi, Sosyal Bilimler Dergisi***, 65–77 (2006).

67. Ford, M. A., Davis, M. H. and Marslen-Wilson, W. D., “Derivational morphology and base morpheme frequency”, ***Journal of Memory and Language***, 63(1): 117-130 (2010).

68. Chakrabarti, S., Ester, M., Fayyad, U., Gehrke, J., Han, J., Morishita, S. and Wang, W., “Data mining curriculum: A proposal (Version 1.0)”, ***Intensive Working Group of ACM SIGKDD Curriculum Committee***, 140 (2006).

69. Radev, P. D., “Introduction Natural Language Processing”.

70. Karcıoğlu, A. A., “İngilizce ve Türkçe twitter mesajlarının Word2vec modeli ile sınıflandırılması”, Yüksek Lisans Tezi, ***Atatürk Üniversitesi Fen Bilimleri Enstitüsü,*** Erzurum (2018).

71. Turing, A. M., “Computing machinery and intelligence‖”, ***Mind***, 59: 433-460 (1950).

72. Hutchins, J., "The history of machine translation in a nutshell", ***Retrieved December,*** 20: 2005 (2009).

73. Adalı, E., "Doğal Dil İşleme", ***Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi,*** 5.2 (2012).

74. Thomas, M., Ryan, C., Alexander, F. and Hinrich, S., "Joint Lemmatization and Morphological Tagging with LEMMING", ***In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing***, 2268-2274 (2015).

75. Bergmanis, T. and Goldwater, S., "Context Sensitive Neural Lemmatization with Lematus", ***In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies Volume 1***, Vol. 1: 1391-1400 (2018).

76. Pundge, A. M., Khillare, S. A. and Mahender, C. N., "Question answering system, approaches and techniques: A review", ***International Journal of Computer Applications***, 141(3) (2016).

77. Nadeau, D. and Sekine, S., "A survey of named entity recognition and classification", ***Lingvisticae Investigationes***, 30(1): 3-26 (2007).

78. Küçük, D. and Arıcı, N., "Türkçe için Wikipedia Tabanlı Varlık İsmi Tanıma Sistemi", ***Politeknik Dergisi***, 19(3): 325-332 (2016).

79. Narin, B., "Uzman Görüşleri Bağlamında Haber Üretiminde Otomatikleşme: Robot Gazetecilik", ***Ileti-s-im***, 27 (2017).

80. Kutlugün, M. A. and Şirin, Y., "Turkish meaningful text generation with class based n-gram model", ***In 2018 26th Signal Processing and Communications Applications Conference (SIU)***, IEEE, 1-4 (2018).

81. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D. and Kappas, A., "Sentiment strength detection in short informal text", ***Journal of the American Society for Information Science and Technology***, 61(12): 2544-2558 (2010).

82. Lee, J. H., Park, S., Ahn, C. M. and Kim, D., "Automatic generic document summarization based on non-negative matrix factorization", ***Information Processing & Management,*** 45(1), 20-34 (2009).

83. Kaynar, O., Işik, Y. E. and Görmez, Y., "Graph based automatic document summarization with different similarity methods", ***In Signal Processing and Communications Applications Conference (SIU),*** IEEE, 1-4 (2017).

84. Erenler Y., "BLG 505 Doğal dil işleme ders notları", ***İ.T.Ü. Fen Bilimleri Enstitüsü***, İstanbul (2004).

85. Mocan, Z., "Metin işleme: Soru soran bir sistem tasarımı", Yüksek Lisans Tezi, ***İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul (2005).

86. Sarma, B. D. and Prasanna, S. M., "Acoustic–phonetic analysis for speech recognition: A review", ***IETE Technical Review***, 35(3), 305-327 (2018).

87. Narayanan, R., "Mining text for relationship extraction and sentiment analysis", PhD Thesis, ***Northwestern University*** (2010).

88. İlhan, S., Duru, N., Karagöz, Ş. and Sağır, M., "Metin Madenciliği ile Soru Cevaplama Sistemi", ***ELECO Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu***, 356-359 (2008).

89. Özçakır, F. C., "Mikroblog hizmetlerindeki örtük bilginin veri madenciliği teknikleri ile keşfi", Doktora Tezi, ***İstanbul Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul (2016).

90. Dolgun, M. Ö., Özdemir, T. G. and Oğuz, D., "Veri madenciliği'nde yapısal olmayan verinin analizi: Metin ve web madenciliği", ***İstatistikçiler Dergisi: İstatistik ve Aktüerya***, 2(2) (2009).

91. Nahm, U. Y. and Mooney, R. J., "Text mining with information extraction", ***In Proceedings of the AAAI 2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases***, Stanford CA, 60-67 (2002).

92. Gaizauskas, R., "An information extraction perspective on text mining: Tasks, technologies and prototype applications", ***In Euromap Text Mining Seminar***, Sheffield (2002).

93. Gupta, V. and Lehal, G. S., "A survey of text mining techniques and applications", ***Journal of emerging technologies in web intelligence***, 1(1): 60-76 (2009).

94. Hamde, M. A., "Kurumsal belgelere (metin verilerine) metin madenciliği tekniği ile erişimin değerlendirilmesi: Türk özel sektörüne yönelik bir inceleme", Doktora Tezi, ***İstanbul Üniversitesi Sosyal Bilimler Enstitüsü***, İstanbul (2018).

95. Zanasi, A., "Virtual weapons for real wars: Text mining for national security", ***In Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS'08***, Springer, Berlin, Heidelberg, 53-60 (2009).

96. Liem, D. A., Murali, S., Sigdel, D., Shi, Y., Wang, X., Shen, J. and Han, J. "Phrase Mining of Textual Data to Analyze Extracellular Matrix Protein Patterns Across Cardiovascular Disease", ***American Journal of Physiology-Heart and Circulatory Physiology*** (2018).

97. Szklarczyk, D., Morris, J. H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M. and Jensen, L. J., "The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible", ***Nucleic acids research***, gkw937 (2016).

98. Papanikolaou, N., Pavlopoulos, G. A., Theodosiou, T. and Iliopoulos, I., "Protein–protein interaction predictions using text mining methods", ***Methods***, 74: 47-53 (2015).

99. Pang, B., Lee, L. and Vaithyanathan, S., “Thumbs up?: sentiment classification using machine learning techniques”, ***In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10***, Association for Computational Linguistics, 79-86 (2002).

100. Sparck Jones, K., “A statistical interpretation of term specificity and its application in retrieval”, ***Journal of documentation***, 28(1): 11-21 (1972).

101. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J., “Distributed representations of words and phrases and their compositionality”, ***In Advances in neural information processing system***, 3111-3119 (2013).

102. Lebret, R. and Collobert, R., “Word emdeddings through hellinger PCA”, arXiv preprint arXiv:1312.5542 (2013).

103. Levy, O. and Goldberg, Y., “Neural word embedding as implicit matrix factorization”, ***In Advances in neural information processing systems,*** 2177-2185 (2014).

104. Li, Y., Xu, L., Tian, F., Jiang, L., Zhong, X. and Chen, E., “Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective”, ***In IJCAI***, 3650-3656 (2015).

105. Globerson, A., Chechik, G., Pereira, F. and Tishby, N., “Euclidean embedding of co-occurrence data”, ***Journal of Machine Learning Research***, 2265-2295 (2007).

106. Qureshi, M. A. and Greene, D., “EVE: explainable vector based embedding technique using Wikipedia”, ***Journal of Intelligent Information Systems***, 1-29 (2017).

107. Levy, O. and Goldberg, Y., “Linguistic regularities in sparse and explicit word representations”, ***In Proceedings of the eighteenth conference on computational natural language learning***, 171-180 (2014).

108. Socher, R., Bauer, J. and Manning, C. D., “Parsing with compositional vector grammars”, ***In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)***, 455-465 (2013).

109. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. and Potts, C., “Recursive deep models for semantic compositionality over a sentiment treebank”, ***In Proceedings of the 2013 conference on empirical methods in natural language processing***, 1631-1642 (2013).

110. Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E. and Smith, N. A., “Retrofitting word vectors to semantic lexicons”, arXiv preprint arXiv:1411.4166, (2014).

111. Kenter, T. and De Rijke, M., "Short text similarity with word embeddings", ***In Proceedings of the 24th ACM international on conference on information and knowledge management***, 1411-1420 (2015, October).

112. Tsvetkov, Y., Faruqui, M., Ling, W., MacWhinney, B. and Dyer, C., "Learning the curriculum with bayesian optimization for task-specific word representation learning", arXiv preprint arXiv:1605.03852 (2016).

113. Mikolov, T., Le, Q. V., and Sutskever, I., "Exploiting similarities among languages for machine translation", arXiv preprint arXiv:1309.4168, (2013).

114. Mikolov, T., Chen, K., Corrado, G. and Dean, J., "Efficient estimation of word representations in vector space", arXiv preprint arXiv:1301.3781 (2013).

115. Wang, Y. H., Lee, H. Y. and Lee, L. S., "Segmental audio word2vec: Representing utterances as sequences of vectors with applications in spoken term detection", ***In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),*** IEEE, 6269-6273 (2018).

116. Le, Q. and Mikolov, T., "Distributed representations of sentences and documents", ***In International Conference on Machine Learning***, 1188-1196 (2014).

117. Pennington, J., Socher, R. and Manning, C., "Glove: Global vectors for word representation", ***In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)***, 1532-1543 (2014).

118. Çimen B., "Text classification via word embeddings: An application for Turkish music mood detection", Yüksek Lisans Tezi, ***Boğaziçi Üniversitesi Sosyal Bilimler Enstitüsü***, İstanbul, (2017).

119. Ozer, I., Ozer, Z. and Findik, O., "Noise robust sound event classification with convolutional neural network", ***Neurocomputing***, 272: 505-512 (2018).

120. Turkson, R. F., Yan, F., Ali, M. K. A. and Hu, J., "Artificial neural network applications in the calibration of spark-ignition engines: an overview", ***Engineering science and technology, an international journal***, 19(3): 1346-1359 (2016).

121. Eichie, J. O., Oyedum, O. D., Ajewole, M. O. and Aibinu, A. M., "Artificial Neural Network model for the determination of GSM Rxlevel from atmospheric parameters", ***Engineering Science and Technology, an International Journal***, 20(2): 795-804 (2017).

122. Ozer, I., Ozer, Z. and Findik, O., "Lanczos kernel based spectrogram image features for sound classification", ***Procedia computer science***, 111: 137-144 (2017).

123. Cavnar, W. B. and Trenkle, J. M., "N-gram-based text categorization", ***Ann arbor mi***, 48113(2), 161-175 (1994).

124. Lin, C. Y. and Hovy, E., "Automatic evaluation of summaries using n-gram co-occurrence statistics", ***In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1***, Association for Computational Linguistics, 71-78 (2003).

125. Tüske, Z., Schlüter, R. and Ney, H., "Investigation on LSTM Recurrent N-gram Language Models for Speech Recognition", ***Proc. Interspeech 2018***, 3358-3362 (2018).

126. Raff, E., Zak, R., Cox, R., Sylvester, J., Yacci, P., Ward, R. and Nicholas, C., "An investigation of byte n-gram features for malware classification", ***Journal of Computer Virology and Hacking Techniques***, 14(1): 1-20 (2018).

127. Mountrakis, G., Im, J. and Ogole, C. "Support vector machines in remote sensing: A review", ***ISPRS Journal of Photogrammetry and Remote Sensing***, 66(3): 247-259 (2011).

128. Qi, Z., Tian, Y. and Shi, Y., "Robust twin support vector machine for pattern classification", ***Pattern Recognition***, 46(1): 305-316 (2013).

129. Guo, G., Li, S. Z. and Chan, K., "Face recognition by support vector machines", ***In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on***, IEEE, 196-201 (2000).

130. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., "Learning phrase representations using RNN encoder-decoder for statistical machine translation", arXiv preprint arXiv:1406.1078, (2014).

131. Singh, D., Khan, M. A., Bansal, A. and Bansal, N., "An application of SVM in character recognition with chain code", ***In Communication, Control and Intelligent Systems (CCIS)***, IEEE, 167-171 (2015).

132. Lauer, F. and Bloch, G., "Incorporating prior knowledge in support vector machines for classification: A review", ***Neurocomputing***, 71(7-9): 1578-1594 (2008).

133. Boser, B. E., Guyon, I. M. and Vapnik, V. N., "A training algorithm for optimal margin classifiers", ***In Proceedings of the fifth annual workshop on Computational learning theory***, ACM, 144-152 (1992).

134. Kaya, D., "Biyomedikal işaretlerin sınıflandırılması için akıllı tekniklerin Labview ortamında gerçeklenmesi", Doktora Tezi, ***Fırat Üniversitesi Fen Bilimleri Enstitüsü***, Elazığ (2018).

135. Eren, Ö., "Alerjen Proteinlerin Otomatik Sınıflandırılması", Yüksek Lisans Tezi, ***Başkent Üniversitesi Fen Bilimleri Enstitüsü***, Ankara (2008).

136. Quinlan, J. R., "C4. 5: Programming for machine learning", ***Morgan Kauffmann***, 38, 48 (1993).

137. McCallum, A. and Nigam, K., "A comparison of event models for naive bayes text classification", ***In AAAI-98 workshop on learning for text categorization*** (752, No. 1, 41-48 (1998).

138. Ozer, I., Ozer, Z. and Findik, O., "Noise robust sound event classification with convolutional neural network", ***Neurocomputing***, 272, 505-512 (2018).

139. İnternet: Science Technology Arts Magazine (Bilim Teknoloji Sanat), "DerlemTR Projesi", **http://gurmezin.com/derlemtr-projesi/** (2018).

140. Tsarfaty, R., Seddah, D., Goldberg, Y., Kübler, S., Candito, M., Foster, J. and Tounsi, L., "Statistical parsing of morphologically rich languages (SPMRL): what, how and whither", ***In Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages***, Association for Computational Linguistics, 1-12 (2010).

141. Sarikaya, R., Kirchhoff, K., Schultz, T. and Hakkani-Tur, D., "Introduction to the special issue on processing morphologically rich languages", ***IEEE Transactions on Audio, Speech, and Language Processing,*** 17(5): 861-862 (2009).

142. Hassan, H. and Menezes, A., "Social text normalization using contextual graph random walks", In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, 1: 1577-1586 (2013).

143. Eryiğit, G., "ITU Turkish NLP web service", ***In Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics***, 1-4 (2014).

144. Showkatramani, G. J., Khatri, N., Landicho, A. and Layog, D., "Trademark Design Code Identification Using Deep Neural Networks", ***In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA),*** IEEE, 61-65 (2018).

145. Zhou, C., Sun, C., Liu, Z. and Lau, F., "A C-LSTM neural network for text classification", arXiv preprint arXiv:1511.08630, (2015).

146. Zharmagambetov, A. S. and Pak, A. A., "Sentiment analysis of a document using deep learning approach and decision trees", ***In Electronics Computer and Computation (ICECCO), 2015 Twelve International Conference on***, IEEE, 1-4 (2015, September).

# EK AÇIKLAMALAR A.
# ÖRNEK FİİL ÇEKİM TABLOSU

| No | Çekimler | Basit/Birinci Zaman/Kip | Birleşik/İkinci Zaman/Kip | Şahıs Eki |
|---|---|---|---|---|
| 1 | anladım | Dili Geçmiş | Hiçbiri (Basit Çekim) | 1. Tekil Şahıs |
| 2 | anladın | Dili Geçmiş | Hiçbiri (Basit Çekim) | 2. Tekil Şahıs |
| 3 | anladı | Dili Geçmiş | Hiçbiri (Basit Çekim) | 3. Tekil Şahıs |
| 4 | anladık | Dili Geçmiş | Hiçbiri (Basit Çekim) | 1. Çoğul Şahıs |
| 5 | anladınız | Dili Geçmiş | Hiçbiri (Basit Çekim) | 2. Çoğul Şahıs |
| 6 | anladılar | Dili Geçmiş | Hiçbiri (Basit Çekim) | 3. Çoğul Şahıs |
| 7 | anlamışım | Mişli Geçmi | Hiçbiri (Basit Çekim) | 1. Tekil Şahıs |
| 8 | anlamışsın | Mişli Geçmi | Hiçbiri (Basit Çekim) | 2. Tekil Şahıs |
| 9 | anlamış | Mişli Geçmi | Hiçbiri (Basit Çekim) | 3. Tekil Şahıs |
| 10 | anlamışız | Mişli Geçmi | Hiçbiri (Basit Çekim) | 1. Çoğul Şahıs |
| 11 | anlamışsınız | Mişli Geçmi | Hiçbiri (Basit Çekim) | 2. Çoğul Şahıs |
| 12 | anlamışlar | Mişli Geçmi | Hiçbiri (Basit Çekim) | 3. Çoğul Şahıs |
| 13 | anlıyorum | Şimdiki | Hiçbiri (Basit Çekim) | 1. Tekil Şahıs |
| 14 | anlıyorsun | Şimdiki | Hiçbiri (Basit Çekim) | 2. Tekil Şahıs |
| 15 | anlıyor | Şimdiki | Hiçbiri (Basit Çekim) | 3. Tekil Şahıs |
| 16 | anlıyoruz | Şimdiki | Hiçbiri (Basit Çekim) | 1. Çoğul Şahıs |
| 17 | anlıyorsunuz | Şimdiki | Hiçbiri (Basit Çekim) | 2. Çoğul Şahıs |
| 18 | anlıyorlar | Şimdiki | Hiçbiri (Basit Çekim) | 3. Çoğul Şahıs |
| 19 | anlayacağım | Gelecek | Hiçbiri (Basit Çekim) | 1. Tekil Şahıs |
| 20 | anlayacaksın | Gelecek | Hiçbiri (Basit Çekim) | 2. Tekil Şahıs |
| 21 | anlayacak | Gelecek | Hiçbiri (Basit Çekim) | 3. Tekil Şahıs |
| 22 | anlayacağız | Gelecek | Hiçbiri (Basit Çekim) | 1. Çoğul Şahıs |
| 23 | anlayacaksınız | Gelecek | Hiçbiri (Basit Çekim) | 2. Çoğul Şahıs |
| 24 | anlayacaklar | Gelecek | Hiçbiri (Basit Çekim) | 3. Çoğul Şahıs |
| 25 | anlarım | Geniş Zaman | Hiçbiri (Basit Çekim) | 1. Tekil Şahıs |
| 26 | anlarsın | Geniş Zaman | Hiçbiri (Basit Çekim) | 2. Tekil Şahıs |
| 27 | anlar | Geniş Zaman | Hiçbiri (Basit Çekim) | 3. Tekil Şahıs |
| 28 | anlarız | Geniş Zaman | Hiçbiri (Basit Çekim) | 1. Çoğul Şahıs |
| 29 | anlarsınız | Geniş Zaman | Hiçbiri (Basit Çekim) | 2. Çoğul Şahıs |
| 30 | anlarlar | Geniş Zaman | Hiçbiri (Basit Çekim) | 3. Çoğul Şahıs |
| 31 | anlamalıyım | Gereklilik | Hiçbiri (Basit Çekim) | 1. Tekil Şahıs |
| 32 | anlamalısın | Gereklilik | Hiçbiri (Basit Çekim) | 2. Tekil Şahıs |

| | | | | |
|---|---|---|---|---|
| 33 | anlamalı | Gereklilik | Hiçbiri (Basit Çekim) | 3. Tekil Şahıs |
| 34 | anlamalıyız | Gereklilik | Hiçbiri (Basit Çekim) | 1. Çoğul Şahıs |
| 35 | anlamalısınız | Gereklilik | Hiçbiri (Basit Çekim) | 2. Çoğul Şahıs |
| 36 | anlamalılar | Gereklilik | Hiçbiri (Basit Çekim) | 3. Çoğul Şahıs |
| 37 | anlasam | Dilek Kipi | Hiçbiri (Basit Çekim) | 1. Tekil Şahıs |
| 38 | anlasan | Dilek Kipi | Hiçbiri (Basit Çekim) | 2. Tekil Şahıs |
| 39 | anlasa | Dilek Kipi | Hiçbiri (Basit Çekim) | 3. Tekil Şahıs |
| 40 | anlasak | Dilek Kipi | Hiçbiri (Basit Çekim) | 1. Çoğul Şahıs |
| 41 | anlasanız | Dilek Kipi | Hiçbiri (Basit Çekim) | 2. Çoğul Şahıs |
| 42 | anlasalar | Dilek Kipi | Hiçbiri (Basit Çekim) | 3. Çoğul Şahıs |
| 43 | anlayayım | İstek | Hiçbiri (Basit Çekim) | 1. Tekil Şahıs |
| 44 | anlayasın | İstek | Hiçbiri (Basit Çekim) | 2. Tekil Şahıs |
| 45 | anlaya | İstek | Hiçbiri (Basit Çekim) | 3. Tekil Şahıs |
| 46 | anlayalım | İstek | Hiçbiri (Basit Çekim) | 1. Çoğul Şahıs |
| 47 | anlayasınız | İstek | Hiçbiri (Basit Çekim) | 2. Çoğul Şahıs |
| 48 | anlayalar | İstek | Hiçbiri (Basit Çekim) | 3. Çoğul Şahıs |
| 49 | anlasana | İstek | Hiçbiri (Basit Çekim) | 2. Tekil Şahıs |
| 50 | anlasanıza | İstek | Hiçbiri (Basit Çekim) | 3. Tekil Şahıs |
| 51 | anla | Emir | Hiçbiri (Basit Çekim) | 2. Tekil Şahıs |
| 52 | anlasın | Emir | Hiçbiri (Basit Çekim) | 3. Tekil Şahıs |
| 53 | anlayın | Emir | Hiçbiri (Basit Çekim) | 2. Çoğul Şahıs |
| 54 | anlayınız | Emir | Hiçbiri (Basit Çekim) | 2. Çoğul Şahıs |
| 55 | anlasınlar | Emir | Hiçbiri (Basit Çekim) | 3. Çoğul Şahıs |
| 56 | anladıydım | Dili Geçmiş | Hikaye | 1. Tekil Şahıs |
| 57 | anladıydın | Dili Geçmiş | Hikaye | 2. Tekil Şahıs |
| 58 | anladıydı | Dili Geçmiş | Hikaye | 3. Tekil Şahıs |
| 59 | anladıydık | Dili Geçmiş | Hikaye | 1. Çoğul Şahıs |
| 60 | anladıydınız | Dili Geçmiş | Hikaye | 2. Çoğul Şahıs |
| 61 | anladıydılar | Dili Geçmiş | Hikaye | 3. Çoğul Şahıs |
| 62 | anlamıştım | Mişli Geçmi | Hikaye | 1. Tekil Şahıs |
| 63 | anlamıştın | Mişli Geçmi | Hikaye | 2. Tekil Şahıs |
| 64 | anlamıştı | Mişli Geçmi | Hikaye | 3. Tekil Şahıs |
| 65 | anlamıştık | Mişli Geçmi | Hikaye | 1. Çoğul Şahıs |

| | | | | |
|---|---|---|---|---|
| 66 | anlamıştınız | Mişli Geçmi | Hikaye | 2. Çoğul Şahıs |
| 67 | anlamıştılar | Mişli Geçmi | Hikaye | 3. Çoğul Şahıs |
| 68 | anlamışlardı | Mişli Geçmi | Hikaye | 3. Çoğul Şahıs |
| 69 | anlıyordum | Şimdiki | Hikaye | 1. Tekil Şahıs |
| 70 | anlıyordun | Şimdiki | Hikaye | 2. Tekil Şahıs |
| 71 | anlıyordu | Şimdiki | Hikaye | 3. Tekil Şahıs |
| 72 | anlıyorduk | Şimdiki | Hikaye | 1. Çoğul Şahıs |
| 73 | anlıyordunuz | Şimdiki | Hikaye | 2. Çoğul Şahıs |
| 74 | anlıyordular | Şimdiki | Hikaye | 3. Çoğul Şahıs |
| 75 | anlıyorlardı | Şimdiki | Hikaye | 3. Çoğul Şahıs |
| 76 | anlayacaktım | Gelecek | Hikaye | 1. Tekil Şahıs |
| 77 | anlayacaktın | Gelecek | Hikaye | 2. Tekil Şahıs |
| 78 | anlayacaktı | Gelecek | Hikaye | 3. Tekil Şahıs |
| 79 | anlayacaktık | Gelecek | Hikaye | 1. Çoğul Şahıs |
| 80 | anlayacaktınız | Gelecek | Hikaye | 2. Çoğul Şahıs |
| 81 | anlayacaktılar | Gelecek | Hikaye | 3. Çoğul Şahıs |
| 82 | anlayacaklardı | Gelecek | Hikaye | 3. Çoğul Şahıs |
| 83 | anlardım | Geniş Zaman | Hikaye | 1. Tekil Şahıs |
| 84 | anlardın | Geniş Zaman | Hikaye | 2. Tekil Şahıs |
| 85 | anlardı | Geniş Zaman | Hikaye | 3. Tekil Şahıs |
| 86 | anlardık | Geniş Zaman | Hikaye | 1. Çoğul Şahıs |
| 87 | anlardınız | Geniş Zaman | Hikaye | 2. Çoğul Şahıs |
| 88 | anlardılar | Geniş Zaman | Hikaye | 3. Çoğul Şahıs |
| 89 | anlarlardı | Geniş Zaman | Hikaye | 3. Çoğul Şahıs |
| 90 | anlamalıydım | Gereklilik | Hikaye | 1. Tekil Şahıs |
| 91 | anlamalıydın | Gereklilik | Hikaye | 2. Tekil Şahıs |
| 92 | anlamalıydı | Gereklilik | Hikaye | 3. Tekil Şahıs |
| 93 | anlamalıydık | Gereklilik | Hikaye | 1. Çoğul Şahıs |
| 94 | anlamalıydınız | Gereklilik | Hikaye | 2. Çoğul Şahıs |
| 95 | anlamalıydılar | Gereklilik | Hikaye | 3. Çoğul Şahıs |
| 96 | anlasaydım | Dilek Kipi | Hikaye | 1. Tekil Şahıs |
| 97 | anlasaydın | Dilek Kipi | Hikaye | 2. Tekil Şahıs |
| 98 | anlasaydı | Dilek Kipi | Hikaye | 3. Tekil Şahıs |

| | | | | |
|---|---|---|---|---|
| 99 | anlasaydık | Dilek Kipi | Hikaye | 1. Çoğul Şahıs |
| 100 | anlasaydınız | Dilek Kipi | Hikaye | 2. Çoğul Şahıs |
| 101 | anlasaydılar | Dilek Kipi | Hikaye | 3. Çoğul Şahıs |
| 102 | anlasalardı | Dilek Kipi | Hikaye | 3. Çoğul Şahıs |
| 103 | anlayaydım | İstek | Hikaye | 1. Tekil Şahıs |
| 104 | anlayaydın | İstek | Hikaye | 2. Tekil Şahıs |
| 105 | anlayaydı | İstek | Hikaye | 3. Tekil Şahıs |
| 106 | anlayaydık | İstek | Hikaye | 1. Çoğul Şahıs |
| 107 | anlayaydınız | İstek | Hikaye | 2. Çoğul Şahıs |
| 108 | anlayaydılar | İstek | Hikaye | 3. Çoğul Şahıs |
| 109 | anlamışmışım | Mişli Geçmi | Rivayet | 1. Tekil Şahıs |
| 110 | anlamışmışsın | Mişli Geçmi | Rivayet | 2. Tekil Şahıs |
| 111 | anlamışmış | Mişli Geçmi | Rivayet | 3. Tekil Şahıs |
| 112 | anlamışmışız | Mişli Geçmi | Rivayet | 1. Çoğul Şahıs |
| 113 | anlamışmışsınız | Mişli Geçmi | Rivayet | 2. Çoğul Şahıs |
| 114 | anlamışmışlar | Mişli Geçmi | Rivayet | 3. Çoğul Şahıs |
| 115 | anlamışlarmış | Mişli Geçmi | Rivayet | 3. Çoğul Şahıs |
| 116 | anlıyormuşum | Şimdiki | Rivayet | 1. Tekil Şahıs |
| 117 | anlıyormuşsun | Şimdiki | Rivayet | 2. Tekil Şahıs |
| 118 | anlıyormuş | Şimdiki | Rivayet | 3. Tekil Şahıs |
| 119 | anlıyormuşuz | Şimdiki | Rivayet | 1. Çoğul Şahıs |
| 120 | anlıyormuşsunuz | Şimdiki | Rivayet | 2. Çoğul Şahıs |
| 121 | anlıyormuşlar | Şimdiki | Rivayet | 3. Çoğul Şahıs |
| 122 | anlıyorlarmış | Şimdiki | Rivayet | 3. Çoğul Şahıs |
| 123 | anlayacakmışım | Gelecek | Rivayet | 1. Tekil Şahıs |
| 124 | anlayacakmışsın | Gelecek | Rivayet | 2. Tekil Şahıs |
| 125 | anlayacakmış | Gelecek | Rivayet | 3. Tekil Şahıs |
| 126 | anlayacakmışız | Gelecek | Rivayet | 1. Çoğul Şahıs |
| 127 | anlayacakmışsınız | Gelecek | Rivayet | 2. Çoğul Şahıs |
| 128 | anlayacakmışlar | Gelecek | Rivayet | 3. Çoğul Şahıs |
| 129 | anlayacaklarmış | Gelecek | Rivayet | 3. Çoğul Şahıs |
| 130 | anlarmışım | Geniş Zaman | Rivayet | 1. Tekil Şahıs |
| 131 | anlarmışsın | Geniş Zaman | Rivayet | 2. Tekil Şahıs |

| | | | | |
|---|---|---|---|---|
| 132 | anlarmış | Geniş Zaman | Rivayet | 3. Tekil Şahıs |
| 133 | anlarmışız | Geniş Zaman | Rivayet | 1. Çoğul Şahıs |
| 134 | anlarmışsınız | Geniş Zaman | Rivayet | 2. Çoğul Şahıs |
| 135 | anlarmışlar | Geniş Zaman | Rivayet | 3. Çoğul Şahıs |
| 136 | anlarlarmış | Geniş Zaman | Rivayet | 3. Çoğul Şahıs |
| 137 | anlamalıymışım | Gereklilik | Rivayet | 1. Tekil Şahıs |
| 138 | anlamalıymışsın | Gereklilik | Rivayet | 2. Tekil Şahıs |
| 139 | anlamalıymış | Gereklilik | Rivayet | 3. Tekil Şahıs |
| 140 | anlamalıymışız | Gereklilik | Rivayet | 1. Çoğul Şahıs |
| 141 | anlamalıymışsınız | Gereklilik | Rivayet | 2. Çoğul Şahıs |
| 142 | anlamalıymışlar | Gereklilik | Rivayet | 3. Çoğul Şahıs |
| 143 | anlamalılarmış | Gereklilik | Rivayet | 3. Çoğul Şahıs |
| 144 | anlasaymışım | Dilek Kipi | Rivayet | 1. Tekil Şahıs |
| 145 | anlasaymışsın | Dilek Kipi | Rivayet | 2. Tekil Şahıs |
| 146 | anlasaymış | Dilek Kipi | Rivayet | 3. Tekil Şahıs |
| 147 | anlasaymışız | Dilek Kipi | Rivayet | 1. Çoğul Şahıs |
| 148 | anlasaymışsınız | Dilek Kipi | Rivayet | 2. Çoğul Şahıs |
| 149 | anlasaymışlar | Dilek Kipi | Rivayet | 3. Çoğul Şahıs |
| 150 | anlasalarmış | Dilek Kipi | Rivayet | 3. Çoğul Şahıs |
| 151 | anlayaymışım | İstek | Rivayet | 1. Tekil Şahıs |
| 152 | anlayaymışsın | İstek | Rivayet | 2. Tekil Şahıs |
| 153 | anlayaymış | İstek | Rivayet | 3. Tekil Şahıs |
| 154 | anlayaymışız | İstek | Rivayet | 1. Çoğul Şahıs |
| 155 | anlayaymışsınız | İstek | Rivayet | 2. Çoğul Şahıs |
| 156 | anlayaymışlar | İstek | Rivayet | 3. Çoğul Şahıs |
| 157 | anladıysam | Dili Geçmiş | Şart Kipi | 1. Tekil Şahıs |
| 158 | anladıysan | Dili Geçmiş | Şart Kipi | 2. Tekil Şahıs |
| 159 | anladıysa | Dili Geçmiş | Şart Kipi | 3. Tekil Şahıs |
| 160 | anladıysak | Dili Geçmiş | Şart Kipi | 1. Çoğul Şahıs |
| 161 | anladıysanız | Dili Geçmiş | Şart Kipi | 2. Çoğul Şahıs |
| 162 | anladıysalar | Dili Geçmiş | Şart Kipi | 3. Çoğul Şahıs |
| 163 | anladılarsa | Dili Geçmiş | Şart Kipi | 3. Çoğul Şahıs |
| 164 | anlamışsam | Mişli Geçmi | Şart Kipi | 1. Tekil Şahıs |

| | | | | |
|---|---|---|---|---|
| 165 | anlamışsan | Mişli Geçmi | Şart Kipi | 2. Tekil Şahıs |
| 166 | anlamışsa | Mişli Geçmi | Şart Kipi | 3. Tekil Şahıs |
| 167 | anlamışsak | Mişli Geçmi | Şart Kipi | 1. Çoğul Şahıs |
| 168 | anlamışsanız | Mişli Geçmi | Şart Kipi | 2. Çoğul Şahıs |
| 169 | anlamışsalar | Mişli Geçmi | Şart Kipi | 3. Çoğul Şahıs |
| 170 | anlamışlarsa | Mişli Geçmi | Şart Kipi | 3. Çoğul Şahıs |
| 171 | anlıyorsam | Şimdiki | Şart Kipi | 1. Tekil Şahıs |
| 172 | anlıyorsan | Şimdiki | Şart Kipi | 2. Tekil Şahıs |
| 173 | anlıyorsa | Şimdiki | Şart Kipi | 3. Tekil Şahıs |
| 174 | anlıyorsak | Şimdiki | Şart Kipi | 1. Çoğul Şahıs |
| 175 | anlıyorsanız | Şimdiki | Şart Kipi | 2. Çoğul Şahıs |
| 176 | anlıyorsalar | Şimdiki | Şart Kipi | 3. Çoğul Şahıs |
| 177 | anlıyorlarsa | Şimdiki | Şart Kipi | 3. Çoğul Şahıs |
| 178 | anlayacaksam | Gelecek | Şart Kipi | 1. Tekil Şahıs |
| 179 | anlayacaksan | Gelecek | Şart Kipi | 2. Tekil Şahıs |
| 180 | anlayacaksa | Gelecek | Şart Kipi | 3. Tekil Şahıs |
| 181 | anlayacaksanız | Gelecek | Şart Kipi | 1. Çoğul Şahıs |
| 182 | anlayacaksak | Gelecek | Şart Kipi | 2. Çoğul Şahıs |
| 183 | anlayacaksalar | Gelecek | Şart Kipi | 3. Çoğul Şahıs |
| 184 | anlayacaklarsa | Gelecek | Şart Kipi | 3. Çoğul Şahıs |
| 185 | anlarsam | Geniş Zaman | Şart Kipi | 1. Tekil Şahıs |
| 186 | anlarsan | Geniş Zaman | Şart Kipi | 2. Tekil Şahıs |
| 187 | anlarsa | Geniş Zaman | Şart Kipi | 3. Tekil Şahıs |
| 188 | anlarsak | Geniş Zaman | Şart Kipi | 1. Çoğul Şahıs |
| 189 | anlarsanız | Geniş Zaman | Şart Kipi | 2. Çoğul Şahıs |
| 190 | anlarsalar | Geniş Zaman | Şart Kipi | 3. Çoğul Şahıs |
| 191 | anlarlarsa | Geniş Zaman | Şart Kipi | 3. Çoğul Şahıs |
| 192 | anlamalıysam | Gereklilik | Şart Kipi | 1. Tekil Şahıs |
| 193 | anlamalıysan | Gereklilik | Şart Kipi | 2. Tekil Şahıs |
| 194 | anlamalıysa | Gereklilik | Şart Kipi | 3. Tekil Şahıs |
| 195 | anlamalıysak | Gereklilik | Şart Kipi | 1. Çoğul Şahıs |
| 196 | anlamalıysanız | Gereklilik | Şart Kipi | 2. Çoğul Şahıs |
| 197 | anlamalıysalar | Gereklilik | Şart Kipi | 3. Çoğul Şahıs |

# EK AÇIKLAMALAR B.
# PROGRAM KODLARI

```
using System;
using System.Collections.Generic;
using System.Collections.Concurrent;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace NGramV2._0.NGrams
{
    public class NGramModel
    {
        private const string validAlphabet =
"abcçdefgğhıijklmnoöpqrsştuüvwxyz ";
        private const string beginSymbol = "<s>";
        private const string endSymbol = "</s>";
        private int tokenCount;
        public int maxNGram;
        public int minNGram;
        public readonly ConcurrentDictionary<NGram, int> nGrams;


        public NGramModel(int nGramSize, string modelFilePath)
        {
            string filteredLine;
            int i = 0;
            maxNGram = nGramSize;
            nGrams = new ConcurrentDictionary<NGram, int>();

            IEnumerable<string> lines =
fileReadLines(modelFilePath);

            foreach (string line in lines)
            {
                i++;
                filteredLine = normalize(line);
                AddLine(filteredLine.Split(null).ToList());
                if (i % 100000 == 0)
                {
                    Console.WriteLine("Read " + i + " lines");
                }

            }

        }

        private IEnumerable<string> fileReadLines(string
modelFilePath)
        {
            IEnumerable<string> lines =
File.ReadLines(modelFilePath);

            return lines;
        }

        private string normalize(string line)
        {
            StringBuilder normalizedLine = new StringBuilder();
            string newLine;
            foreach (char ch in line.ToLower().ToCharArray())
```

```
            {
                string newch = ch.ToString();
                if (!validAlphabet.Contains(ch))
                {
                    newch = "";
                }
                normalizedLine.Append(newch);
            }

            RegexOptions options = RegexOptions.None;
            Regex regex = new Regex("[ ]{2,}", options);
            newLine = regex.Replace(normalizedLine.ToString(), " ");

            return newLine;
        }

        public void AddLine(IEnumerable<string> tokens)
        {
            List<string> tokenList = tokens.ToList();

            tokenCount += tokenList.Count;
            AddBeginEndSymbols(tokenList);
            AddSeries(tokenList);
        }


        public void AddBeginEndSymbols(IList<string> tokens)
        {
            if (maxNGram <= 1) return;

            for (int i = 0; i < maxNGram - 1; i++)
            {
                string index = i.ToString();
                tokens.Insert(0, beginSymbol.Insert(2, index));
            }

            tokens.Add(endSymbol);
        }

        public void AddSeries(IEnumerable<string> tokens)
        {
            ConcurrentDictionary<NGram, int> newNGrams =
createDictionary(tokens);
            nGrams.Unify(newNGrams);
        }


        public ConcurrentDictionary<NGram, int>
createDictionary(IEnumerable<string> tokens)
        {
            var dictionary = new ConcurrentDictionary<NGram, int>();
            IList<NGram> nGrams = createList(tokens);

            foreach (NGram nGram in nGrams)
            {
                if (!dictionary.ContainsKey(nGram))
                {
                    dictionary.TryAdd(nGram, 1);
                }
                else
```

```
                {
                    dictionary[nGram]++;
                }
            }

            return dictionary;
        }

        public IList<NGram> createList(IEnumerable<string> tokens)
        {
            var nGrams = new List<NGram>();
            List<string> tokenList = tokens.ToList();
            for (int i = 0; i <= tokens.Count() - minNGram; i++)
            {
                nGrams.AddRange(GetNGrams(tokenList, i));
            }
            return nGrams;
        }

        private IEnumerable<NGram> GetNGrams(IList<string> tokens,
int index)
        {
            var nGrams = new List<NGram>();
            int maxWindowSize = Math.Min(tokens.Count() - index,
maxNGram);
            for (int windowSize = minNGram; windowSize <=
maxWindowSize; windowSize++)
            {
                NGram nGram = GetNGram(tokens, index, windowSize);
                nGrams.Add(nGram);
            }
            return nGrams;
        }

        private NGram GetNGram(IList<string> tokens, int index, int
windowSize)
        {
            var nGram = new List<string>();
            for (int i = 0; i < windowSize; i++)
            {
                nGram.Add(tokens[index + i]);
            }
            return new NGram(nGram);
        }

        public void  ReadNgramFromFile(string fileName)
        {
            ConcurrentDictionary<NGram, int> nGrams = new
ConcurrentDictionary<NGram, int>();
            var fileStream = new FileStream(fileName, FileMode.Open,
FileAccess.Read);

            using (var streamReader = new StreamReader(fileStream,
Encoding.UTF8))
            {

                string line;
                while ((line = streamReader.ReadLine()) != null)
                {
                    string[] row = line.Split('\t');
```

```
                    var nGram = new NGram(row[0].Split(null));
                    int freq = Int32.Parse(row[1]);

                    if (!nGrams.ContainsKey(nGram))
                    {
                        nGrams.TryAdd(nGram, freq);
                    }
                }
            }

        }

    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Oz.Lang;
using Oz.Morphologic.Structure;
using Oz.Tokenizers;

namespace ConsoleApplication5.deAsciification
{
    public class SimpleDeasciifier
    {
        public void generateCandidateList(List<string> candidates,
string word, int index)
        {

            string s = "ıougcsIOUGCS";
            if (index < word.Length)
            {
                if (s.IndexOf(word[index]) != -1)
                {
                    int size = candidates.Count;
                    for (int i = 0; i < size; i++)
                    {
                        char[] modified =
candidates[i].ToCharArray();
                        switch (word[index])
                        {
                            case 'ı':
                                modified[index] = 'i';
                                break;
                            case 'o':
                                modified[index] = 'ö';
                                break;
                            case 'u':
                                modified[index] = 'ü';
                                break;
                            case 'g':
                                modified[index] = 'ğ';
                                break;
```

```
                                case 'c':
                                    modified[index] = 'ç';
                                    break;
                                case 's':
                                    modified[index] = 'ş';
                                    break;
                                case 'I':
                                    modified[index] = 'İ';
                                    break;
                                case 'O':
                                    modified[index] = 'Ö';
                                    break;
                                case 'U':
                                    modified[index] = 'Ü';
                                    break;
                                case 'G':
                                    modified[index] = 'Ğ';
                                    break;
                                case 'C':
                                    modified[index] = 'Ç';
                                    break;
                                case 'S':
                                    modified[index] = 'Ş';
                                    break;
                            }
                            candidates.Add(new string(modified));

                        }
                    }
                    generateCandidateList(candidates, word, index + 1);
                }
            }

            public  List<string> candidateList(string word)
            {

                List<string> candidates;
                candidates = new List<string>();
                Language tr =
LanguageFactory.Create(LanguageType.Turkish);

                candidates.Add(word);
                generateCandidateList(candidates, word, 0);
                for (int i = 0; i < candidates.Count; i++)
                {
                    IList<Word> solutions = tr.Analyze(candidates[i]);
                    if (solutions.Count == 0)
                    {

                        solutions =
tr.Analyze(FirstCharToUpper(candidates[i]));

                    }
                    if(solutions.Count ==0)
                    {
                        candidates.RemoveAt(i);
                        i--;
                    }
                }
                return candidates;
```

```
        }

        public static string FirstCharToUpper(string input)
        {
            if (String.IsNullOrEmpty(input))
                throw new ArgumentException("ARGH!");
            return input.First().ToString().ToUpper() +
input.Substring(1);
        }
    }
}



using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication5.deAsciification
{
    class SimpleAsciifier
    {
        public String asciify(string word)
        {
            char[] modified = word.ToCharArray();
            for (int i = 0; i < modified.Length; i++)
            {
                switch (modified[i])
                {
                    case 'ç':
                        modified[i] = 'c';
                        break;
                    case 'ö':
                        modified[i] = 'o';
                        break;
                    case 'ğ':
                        modified[i] = 'g';
                        break;
                    case 'ü':
                        modified[i] = 'u';
                        break;
                    case 'ş':
                        modified[i] = 's';
                        break;
                    case 'ı':
                        modified[i] = 'i';
                        break;
                    case 'Ç':
                        modified[i] = 'C';
                        break;
                    case 'Ö':
                        modified[i] = 'O';
                        break;
                    case 'Ğ':
                        modified[i] = 'G';
                        break;
                    case 'Ü':
                        modified[i] = 'U';
```

```
                        break;
                    case 'Ş':
                        modified[i] = 'S';
                        break;
                    case 'İ':
                        modified[i] = 'I';
                        break;
                }
            }
            return new String(modified);
        }
    }
}

using Oz.Lang;
using Oz.Morphologic.Structure;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication5.spellCheck
{
    class spellCheck
    {
        public List<string> candidate(string word)
        {
            List<string> candidate = new List<string>();
            List<string> candidate_list =
generateCandidateList(word);
            candidate= finalList(candidate_list);
            return candidate;
        }

        public List<String> generateCandidateList(string word)
        {
             string s = "abcçdefgğhıijklmnoöprsştuüvyz ";;
            List<String> candidates = new List<String>();
            for (int i = 0; i < word.Length; i++)
            {
                String deleted = word.Substring(0, i) +
word.Substring(i + 1);
                candidates.Add(deleted);
                for (int j = 0; j < s.Length; j++)
                {
                    String replaced = word.Substring(0, i) + s[j] +
word.Substring(i + 1);
                    candidates.Add(replaced);
                }

            }
            for (int i = 0; i <= word.Length; i++)
            {
                for (int j = 0; j < s.Length; j++)
                {
                    String added = word.Substring(0, i) + s[j] +
word.Substring(i);
                   // Console.WriteLine(added);
                    candidates.Add(added);
```

```
                }

            }
                return candidates;
        }

        protected List<string> finalList(List<string> candidate)
        {
            Language tr =
LanguageFactory.Create(LanguageType.Turkish);

            for(int i=0; i< candidate.Count;i++)
            {
                IList<Word> solutions = tr.Analyze(candidate[i]);
                if (solutions.Count == 0)
                {
                    candidate.RemoveAt(i);
                    i--;

                }

            }

            return candidate;
        }
    }
}



using System;
using System.Linq;
using System.Data.SqlClient;

namespace ConsoleApplication2
{
    class Program
    {
        static void Main(string[] args)
        {

            var tokens = new Twitterizer.OAuthTokens
            {
                ConsumerKey = "xxxxxxx",
                ConsumerSecret = "xxxxxxx",
                AccessToken = "xxxxxxxx",
                AccessTokenSecret = "xxxxxxx"
            };

            decimal maksimumID =1;

            SqlConnection baglanti = new SqlConnection("Data
Source=ZEYNEP-BILGI;Initial Catalog=twitter1;Integrated
Security=True");
            baglanti.Open();

            while (true)
            {
                Console.WriteLine("döngü");
                maksimumID = maksimumID + 1;
```

```
                try
                {
                  var response =
Twitterizer.TwitterSearch.Search(tokens, " ",
                  new Twitterizer.SearchOptions
                  {
                      GeoCode = "39.913543,32.816591,27km",
                      Count = 100,
                      Language = "tr",
                      SinceId = maksimumID,

                  });

                    System.Threading.Thread.Sleep(10000);
                    if (response.Result !=
Twitterizer.RequestResult.Success || response.ResponseObject.Count
== 0)
                    {
                        System.Threading.Thread.Sleep(900);
                    }
                    else
                    {
                        foreach (var status in
response.ResponseObject)
                        {
                            Console.WriteLine(status.Id);
                            Console.Write("tarih  ");
                            Console.WriteLine(status.CreatedDate);
                            Console.Write("ID si ");
                            Console.WriteLine(status.Id);
                            Console.Write("Kullanıcı ADI ");
                            Console.WriteLine(status.User.Name);

Console.WriteLine(status.User.ScreenName);
                            Console.WriteLine(status.User.Location);
                            Console.WriteLine(status.Text);

                            if (status.Id > maksimumID)
                                maksimumID = status.Id;

                            SqlCommand ekle = new SqlCommand("Insert
Into tweet_data
(tweetID,kullaniciScreenName,kullaniciADI,tarih,tweet) Values ('" +
status.Id + "','" + status.User.ScreenName.Replace("'", "''") +
"','" + status.User.Name.Replace("'", "''") + "','" +
status.CreatedDate.ToString("yyyy - MM - dd HH: mm:ss") + "','" +
status.Text.Replace("'", "''") + "')", baglanti);
                            ekle.ExecuteNonQuery();

                            if (status.Geo != null)
                            {
                                Console.Write("geotagging");

Console.WriteLine(status.Geo.Coordinates.ElementAt(0).Latitude);

Console.WriteLine(status.Geo.Coordinates.ElementAt(0).Longitude);

                                SqlCommand guncelle = new
SqlCommand();
                                guncelle.Connection = baglanti;
```

```
                                    guncelle.CommandText = "Update
tweet_data set enlem ='" +
status.Geo.Coordinates.ElementAt(0).Latitude + "', boylam = '" +
status.Geo.Coordinates.ElementAt(0).Longitude + "' where tweetID ='"
+ status.Id + "'";
                                    guncelle.ExecuteNonQuery();

                                }
                                if (status.Place != null)
                                {
                                    Console.WriteLine("mekan");

Console.WriteLine(status.Place.FullName);

                                    SqlCommand guncelle1 = new
SqlCommand();
                                    guncelle1.Connection = baglanti;
                                    guncelle1.CommandText = "Update
tweet_data set mekan='" + status.Place.FullName.Replace("'", "''") +
"' where tweetID ='" + status.Id + "'";
                                    guncelle1.ExecuteNonQuery();

                                }

                            }
                        }
                    }
                    catch { Console.WriteLine("döngüde hata oluştu"); }
                }

            }
        }
    }
```

## ÖZGEÇMİŞ

Zeynep ÖZER 1986 yılında Ankara'da doğdu; ilk öğrenimini Kırıkkale'de, orta öğrenimini ise Ankara'da tamamladı. Pursaklar Anadolu Lisesinden mezun oldu. 2005 yılında Dumlupınar Üniversitesi Fen-Edebiyat Fakültesi Matematik Bölümü'nde öğrenime başlayıp 2009 yılında mezun oldu. 2010 yılında Ankara Açı Eğitim Kurumları'nda Geometri Öğretmeni olarak göreve başladı. 2011 yılında Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar ve Bilişim Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimini tamamlayarak 2012 yılında Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda doktora eğitimine başladı. 2011 yılında Karabük Üniversitesi'nde göreve başladı ve halen aynı yerde çalışmaya devam etmektedir.

## ADRES BİLGİLERİ

Adres : Kartaltepe Mah. Cevizkent Toki Konutları
C4-1 Blok, 14. Kat, No:62
Merkez / KARABÜK
Tel : (507) 437 2656
E-posta : zeynep.ozer@outlook.com