T.C.
GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

İmla Hatalı Cümleleri Düzelten Sistem (Machine Translation)

Yunus ÇEVİK

Danışman
Dr. Öğr. Üyesi Burcu YILMAZ

Ocak, 2019
Gebze, KOCAELİ

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

İmla Hatalı Cümleleri Düzelten Sistem (Machine Translation)

Yunus ÇEVİK

Danışman
Dr. Öğr. Üyesi Burcu YILMAZ

Ocak, 2019
Gebze, KOCAELİ

Bu çalışma ..../..../2019 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümü'nde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı	Dr. Öğr. Üyesi Burcu YILMAZ	
Üniversite	GEBZE TEKNİK ÜNİVERSİTESİ	
Fakülte	MÜHENDİSLİK FAKÜLTESİ	

Jüri Adı	Dr. Murat ŞEKER	
Üniversite	GEBZE TEKNİK ÜNİVERSİTESİ	
Fakülte	MÜHENDİSLİK FAKÜLTESİ	

ÖNSÖZ

Tez çalışmamın planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın danışman hocam Dr. Burcu YILMAZ' a ve bu çalışmayı destekleyen Gebze Teknik Üniversitesi'ne sonsuz ve içten teşekkürlerimi sunarım.

Ayrıca eğitimim süresince bana her konuda tam destek veren aileme ve bana hayatlarıyla örnek olan tüm hocalarıma saygı ve sevgilerimi sunarım.

Ocak, 2019 **Yunus ÇEVİK**

İÇİNDEKİLER

ŞEKİL LİSTESİ

TABLO LİSTESİ

KISALTMA LİSTESİ

NLP : Natural Language Processing (Doğal Dil İşleme)

LSTM : Long-Short Term Memory (uzun-kısa süreli bellek)

ML : Machine Learning (Makine Öğrenmesi)

MT : Machine Translation (Makine Çevirisi)

RNN : Recurrent Neural Network

ÖZET

Projenin amacı imla hatalı cümleleri düzelten bir sistem kurmaktır. Günlük hayatta birçok kelime kısaltılarak ve imla kurallarından saptırılarak kullanılır. Bu duruma mesajlaşma dilinde "merhaba" yerine "mrhb" veya "selam" yerine "slm" kullanılması gibi veya bilgisayarda on parmak yazı yazılırken yanlış bir harfe basmak örnek olarak verilebilir. Bu durum resmi yazılarda imla bozukluğundan dolayı anlam bozukluğuna ve anlam karmaşasına yol açar.

İmla Hatalı Cümleleri Düzelten Sistem ise bu tür problemleri Türkçe imla kurallarına bağlı olarak tekrar düzenler ve bu şekilde oluşabilecek sorunların giderilmesine yardım eder. 2015 yılına ait makale, dergi, gazete gibi düzgün anlatıma sahip kaynaklardan Python' da crawler yardımı ile bilgiler çekilerek doğru imla kurallı bir dataset oluşturulmuştur. Oluşturulan dataset (Doğru imla kurallı) de bulunan kelimelerin veya cümlelerin yapısının bozulması ile yeni bir dataset (Bozuk imla kurallı) oluşturulmuştur. Makine öğrenmesi ile imla kurallarına uymayan kelime veya cümlelerin orijinal halleri ile sistem eğitilmiştir.

One-hot encode yapısı kullanılarak karakter tabanlı vektörler oluşturulmuştur. Hazırlanan datasetlerin eğitimi için Google Colab kullanılmıştır. İmla Hatalı Cümleleri Düzelten sistem bir web sitesi aracılığı ile kullanıcıya sunulmuştur. Projenin derin öğrenme bölümü Python programlama dili ile yazılmıştır. Python dili ile yazabilmek için Anaconda' nın Spyder ve Jupyter Notebook programları kullanılmıştır. Makine öğrenmesi bölümü, Theano veya Tensorflow'u backend olarak kullanan Keras kütüphanesi ile yapılmıştır.

SUMMARY

The aim of the project is to establish a system that corrects incorrect spelling sentences. In daily life, many words are shortened and disrupted from spelling rules. An example of this is the use of "mrhb" instead of "merhaba" or "slm" instead of "selam" in the messaging language, or an incorrect letter while writing ten fingers on the computer. This situation leads to meaningfulness and meaning confusion due to spelling in the official writings.

The Incorrect Spelling Sentences Corrector System restores such problems depending on the Turkish spelling rules and help to solve problems that may occur in this way. In Python, data set that has correct spelling rules has been created with the help of crawler. The datas has been taken from the sources such as articles, journals and newspapers in 2015. A new dataset (with corrupted spelling) was created with the disruption of the structure of the words or sentences found in the created correct dataset. The system is trained by machine learning with and original and disrupted conditions of words or sentences .

Character-based vectors are constructed using the one-hot encode structure. Google Colab is used for training of prepared datasets. Incorrect Spelling Sentences The Corrector System is presented to the user via a website. The deep learning section of the project is written in the Python programming language. Anaconda's Spyder and Jupyter Notebook programs were used to write in Python. The machine learning section was made with the Keras library, which uses Theano or Tensorflow as a backend.

1. GİRİŞ

Hayat hızlanıyor ve bu hız her geçen gün artıyor. Artan bu hız karşısında insanlar artık aynı zamanda daha fazla iş yapmak zorunda kalıyor. Bu gereksinim doğal olarak insanlar arasındaki iletişimi de etkilemektedir. İletişimin de artık daha pratik ve hızlı yapılması, karşı tarafa aynı kelimeyi daha çabuk anlatabilmeyi, daha fazla sayıda ve farklı ülkelere, kültürlere, eğitime, bilgiye, mesleğe sahip insanları anlayabilmeyi gerektiriyor. İşyerlerinde yapılan yazışmalarda, tanıdıklara gönderilen mesajlarda, bir adresi ya da yeri anlatırken, her zaman daha hızlı ve çabuk olanı kullanmak hızlanan dünyada şart hale gelmiştir. Bu yüzden kelimeler kısaltılmaktadır **([1] Altuğ TATLI, 25 Kasım 2014)**.

Dil, insanlar arasında anlaşmaya, iletişim kurmaya yarayan bir araçtır. Bu yüzden dili doğru kullanmak çok önemlidir. Dillere özgü kurallar farklılık göstermektedir, bu durum okuyucuların metinleri okurken farklı bağlamlar oluşturmasına yol açabilir. Özellikle Türkçe dili sahip olduğu özellikler bakımından oldukça zorlayıcı yapılara ve karmaşık kurallara sahiptir. Bu karmaşıklığı önlemek için Derin Öğrenme tekniklerini kullanmak uygun bir çözüm oluşturur.

Derin Öğrenme (Deep Learning), giriş ile çıkış arasında çeşitli yapay nöronlar kullanılarak katmanlı yapıda oluşturulan sinir ağı anlamına gelmektedir. Derin öğrenme, resim, konuşma ve doğal dil işleme de diğer metotlara göre daha iyi sonuçlar vermektedir. Bunun sebebi ise tıpkı beynimiz gibi girdi ve çıktı arasında bulunan katmanlarda probleme özel özelliklerin tanımlanmasını ve işlenmesini gerçekleştirmesidir [2].

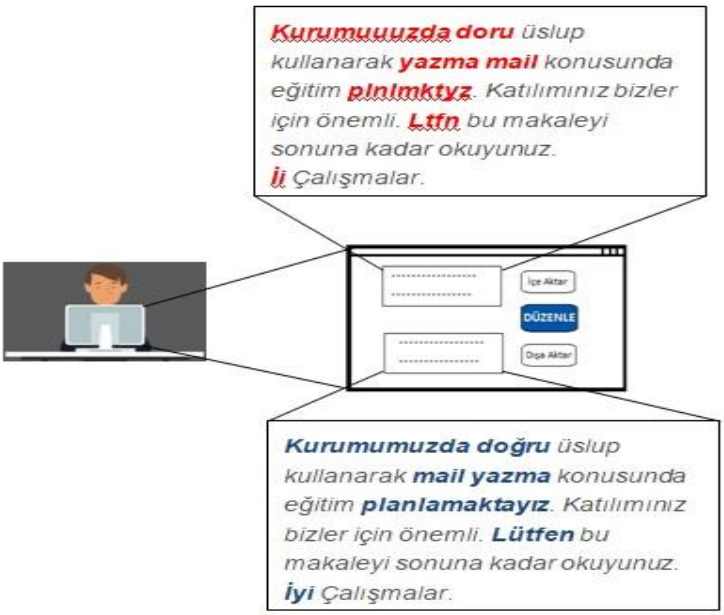
Doğal dil işleme (NLP) yapay zeka ve dilbilimin alt kategorisinde bulunmaktadır. NLP, doğal dildeki metinlerin ve/veya seslerin bilgisayar ortamında işlenmesi üzerine çalışmalar yürüten, bilgisayar bilimi ve dil bilimciliğinin bir alt bilim dalıdır. Bu kapsamda sesin dalgalarının tanınması ve bilgisayar ortamında metne aktarılması, metin seslendirme, biçim bilimsel çözümleme/üretme, sözdizim çözümlemesi, anlamsal çözümleme gibi yöntemler kullanılmakta ve

geliştirilmektedir. Yazım hatalarının denetlenmesi / düzeltilmesi, bilgisayarlı çeviri, bilgi çıkarımı, bilgi getirimi, soru cevap sistemlerinin geliştirilmesi, özet çıkartma gibi uygulamalar doğal dil işleme tanımı altında toplanmıştır [3]. Doğal dil işlemenin özellikleri bu projede kullanılmak için oldukça uygun bulunmuştur.

Projenin amacı; akademik çalışmaların dökümantasyonunda, resmi yazışmalarda ve önemli yayınlarda(dergi, makale vb.) meydana gelebilelecek on parmak hızlı yazmaktan kaynaklanan, yazılacak yazıyı tahmin edebilen telefon uygulamalarından kaynaklanan veya kullanıcının kelimeyi yanlış bilmesinden kaynaklanan yazım ve imla hatalarının tespit edilip, yüksek verimlilikle düzeltilmesidir.

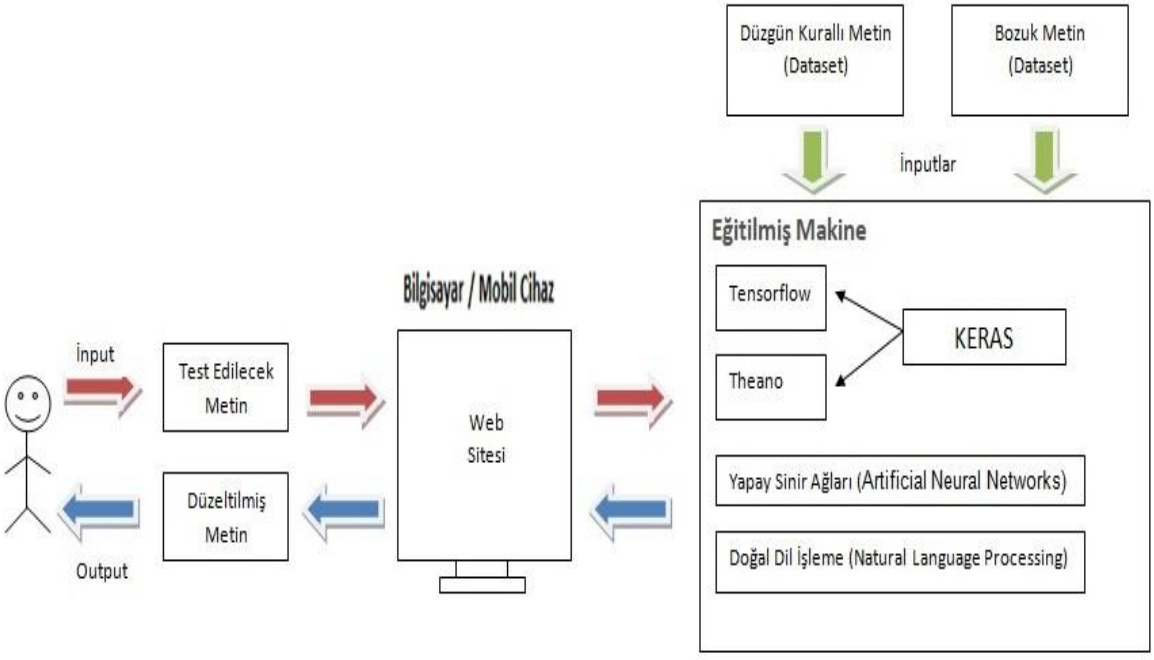
1.1. PROJENİN TANIMI

İş hayatında veya resmi yazışmalarda kullanılan dil, konunun resmiyetine uygun olmalıdır. Ayrıca günlük hayatta kullanılan dilde ise değişmeler olmuş ve bu değişmeler insanları kelimeleri kısaltarak kullanmaya yöneltmiştir. Örnek verecek olursak Şekil 1’ deki gibi bir kişi kısaltmalar sonucu anlam karmaşası yaşamıştır. Ancak bu anlam karmaşasını önlemek için Derin Öğrenme ile eğitilmiş olan bir sisteme yanlış imla kurallı yapıyı vererek, doğru imla kurallı yapıyı elde edebilmiştir.



Şekil 1: Sistemin Genel Yapısı

Belirtilen örnekte kullanıcının kullandığı sistemin arka planı Şekil 2' deki gibi özetlenebilir.



Şekil 2: Sistemin Çalışma Prensibi

1.2. PROJENİN NEDENİ VE AMAÇLARI

- ❖ Resmi yazışmalarda kullanılan kısaltmalar sonucu oluşabilecek anlam karmaşalarına engel olmak,
- ❖ Akademik yazılar yayınlarken, literatüre uymayan ifadeleri engellemek,
- ❖ Kullanıcın eksik bildiği yazım kurallarını tamamlayabilmektir.

Ayrıca tüm dünya çapında Derin Öğrenmeye karşı duyulan ilgi ve merak artmaktadır. Bununla birlikte benzer konularda yapılan çalışmaların başarılı olması sebebi ile Derin Öğrenme teknikleri kullanılmak istenmiştir.

2. MALZEME VE YÖNTEM

Bu bölümde giriş bölümünde bahsedilen proje amaçları doğrultusunda projenin gereksinimleri ele alınacaktır. Projenin alt yapısını oluşturan makine öğrenmesi, Theano veya Tensorflow'u backend olarak kullanan Keras kütüphanesi, derin öğrenme, doğal dil işleme, makine çevirisi, LSTM, Google Colab gibi gereksinimlerden bu bölümde bahsedilecektir.

2.1. PROJE GEREKSİNİMLERİ

2.1.1. MAKİNE ÖĞRENMESİ

Makine öğrenmesi (ML), yazılım uygulamalarının açıkça programlanmadan sonuçları tahmin etmede daha doğru olmalarını sağlayan bir algoritma kategorisidir. Makine öğreniminin temel öncülü girdi verilerini alabilen algoritmalar oluşturmak ve çıktıları güncellerken yeni verileri mevcut hale getirirken çıktıyı tahmin etmek için istatistiksel analizi kullanmaktır. Makine öğreniminde yer alan süreçler, veri madenciliği ile tahmine dayalı modellemeye benzerdir. Her ikisi de, kalıp aramak ve program eylemlerini buna göre ayarlamak için verilerde arama yapılmasını gerektirir.

Makine öğreniminin neredeyse sınırsız kullanımı olduğu gibi, makine öğrenme algoritmalarında da bir sıkıntı yoktur. Oldukça basit olandan oldukça karmaşık olana kadar çeşitlilik gösterirler. İşte en sık kullanılan modellerden birkaçı:

Karar ağaçları: Bu modeller belirli eylemler hakkında gözlemler kullanır ve istenen bir sonuca ulaşmak için en uygun yolu belirler.

Yapay sinir ağları: Bu derin öğrenme modelleri, gelecekte gelen verileri işlemeyi öğrenmek için birçok değişken arasındaki korelasyonu tanımlamak için büyük miktarda eğitim verisi kullanır. Projenin makine öğrenmesi bölümünde yapay sinir ağları kullanılmıştır**(Margaret Rouse, Mayıs 2018).**

.

2.1.2. THEANO VEYA TENSORFLOW BACKEND İLE KERAS KÜTÜPHANESİ

Keras, derin öğrenme modelleri geliştirmek için üst düzey yapı taşları sağlayan, model düzeyinde bir kütüphanedir. Tensör ürünleri, konvolüsyonlar ve benzeri gibi düşük seviye işlemlerini yapmaz. Bunun yerine, Keras'ın "backend" olarak hizmet veren özel, iyi optimize edilmiş bir tensör manipülasyon kütüphanesine dayanır. Tek bir tensör kütüphanesi seçmek ve Keras'ın o kütüphaneye bağlı olmasını sağlamak yerine, Keras sorunu modüler bir şekilde ele alıyor ve birkaç farklı backend, Keras ile sorunsuz bir şekilde birleştirilebilir. Şu anda, Keras'ın iki backend uygulaması mevcuttur: BunlarTensorFlow ve Theano' dır.

TensorFlow, Google, Inc. tarafından geliştirilen açık kaynaklı bir sembolik tensör manipülasyon framework' dür.

Theano, Université de Montréal'deki LISA / MILA Lab tarafından geliştirilen açık kaynaklı bir sembolik tensör manipülasyon framework' dür[5].

2.1.3. DERİN ÖĞRENME

Derin öğrenme (derin yapılandırılmış öğrenme veya hiyerarşik öğrenme olarak da bilinir), göreve özgü algoritmaların aksine, veri sunumlarını temel alan daha geniş bir makine öğrenme yöntemleri ailesinin bir parçasıdır. Öğrenme denetlenebilir, yarı denetlenebilir veya denetlenemez[14].

Bilgisayarda görme, konuşma tanıma, doğal dil işleme, ses tanıma, sosyal ağ filtreleme, makine çevirisi, biyoinformatik, ilaç tasarımı, tıbbi görüntü analizi gibi alanlara derin sinir ağları ve tekrarlayan sinir ağları gibi derin öğrenme mimarileri

uygulanmıştır. Bu bölümde encode ve decode yapıları ile cümleler vektörlere dönüştürülerek eğitilmiştir.

Bir encoder, girdiyi alan ve bir özellik haritası / vektörü / tensörü çıkaran bir ağdır (FC, CNN, RNN, vb.). Bu özellik vektörü girdiyi temsil eden bilgiyi tutar. Decoder, yine özellik vektörünü encoder' dan alan ve gerçek girdiye veya amaçlanan çıktıya en yakın eşleşmeyi veren bir ağdır (genellikle encoder ile aynı ağ yapısına karşılık gelir). Encoder' lar, decoder' lar ile birlikte eğitilir. Meydana gelen kayıp ise, gerçek ve yeniden yapılandırılmış girdi arasındaki delta hesaplamaya dayanır. Bir kez eğitilen encoder, yeniden yapılandırılmış girdiyi gerçek girdi olarak tanınabilir kılar ve decoder için kullanılabilecek vektör haline dönüştürür. Bu şekilde eğitilen veriler yeni girdi olur [6].

2.1.4. DOĞAL DİL İŞLEME

NLP, bilgisayarların insan dilinden akıllıca ve kullanışlı bir şekilde anlam çıkarması, anlaması ve türetmesi için bir yoldur. Geliştiriciler, NLP kullanarak, otomatik özetleme, çeviri, ilişki çıkarma, duyarlılık analizi, konuşma tanıma ve konu bölümlendirme gibi görevleri yerine getirmek için bilgileri düzenleyebilir ve yapılandırabilir[15].

NLP, bilgisayar bilimlerinde zor bir problem olarak tanımlanmaktadır. İnsan dili nadiren kesindir veya açıkça konuşulur. İnsan dilini anlamak, sadece kelimeleri değil, kavramları ve anlam yaratmak için nasıl bir araya geldiklerini anlamaktır. Dil, insanların öğrenmesi en kolay şeylerden biri olmasına rağmen, dilin belirsizliği, doğal dili işlemeyi bilgisayarların ustalaşması için zor bir sorun yapar[7].

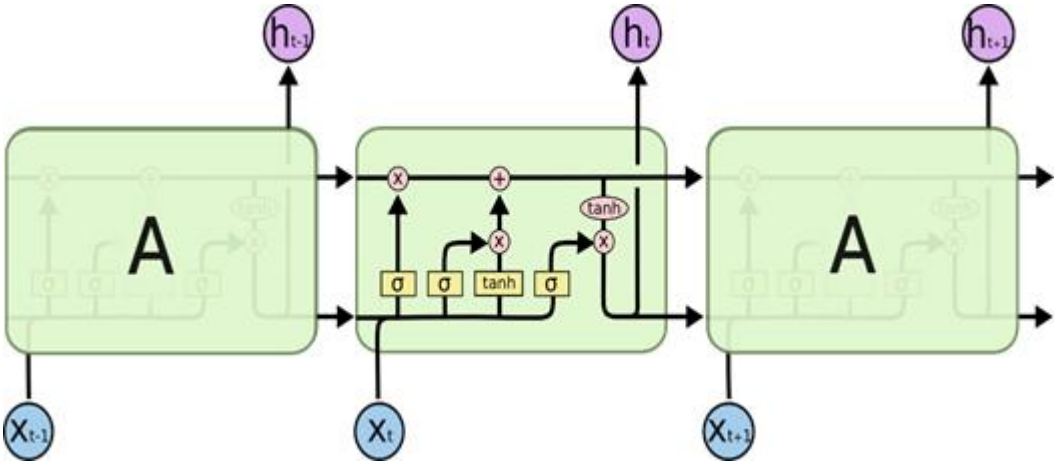
2.1.5. MAKİNE ÇEVİRİSİ

Makine çevirisi, metin veya konuşmayı bir dilden diğerine çevirmek için yazılım kullanımını araştıran, hesaplamalı dilbilimin alt alanıdır. Temel düzeyde MT, bir dildeki kelimelerin başka bir dilden basit bir şekilde değiştirilmesini sağlar, ancak tek başına genellikle bir metnin iyi bir çevirisini yapamaz, çünkü tüm ifadelerin ve hedef dilindeki en yakın benzerlerinin tanınması gerekir[13]. Bu sorunu corpus istatistik ve sinir ağı teknikleriyle çözmek, daha iyi çevirilere, dilbilimsel tipolojideki farklılıkları ele almak, deyimlerin çevirisi ve anomalilerin izole edilmesine yol açar[8].

2.1.6. LSTM YAPISI

Uzun kısa süreli bellek (LSTM) birimleri, tekrarlayan bir sinir ağının (RNN) birimleridir.(Şekil 4) LSTM birimlerinden oluşan bir RNN'ye genellikle LSTM ağı denir. Ortak bir LSTM birimi bir hücreden, bir giriş geçidinden, bir çıkış geçidinden ve bir unutma geçidinden oluşur. Hücre, keyfi zaman aralıkları boyunca değerleri hatırlar ve üç bu geçit, hücrenin içine ve dışına bilgi akışını düzenler (Şekil 3) .

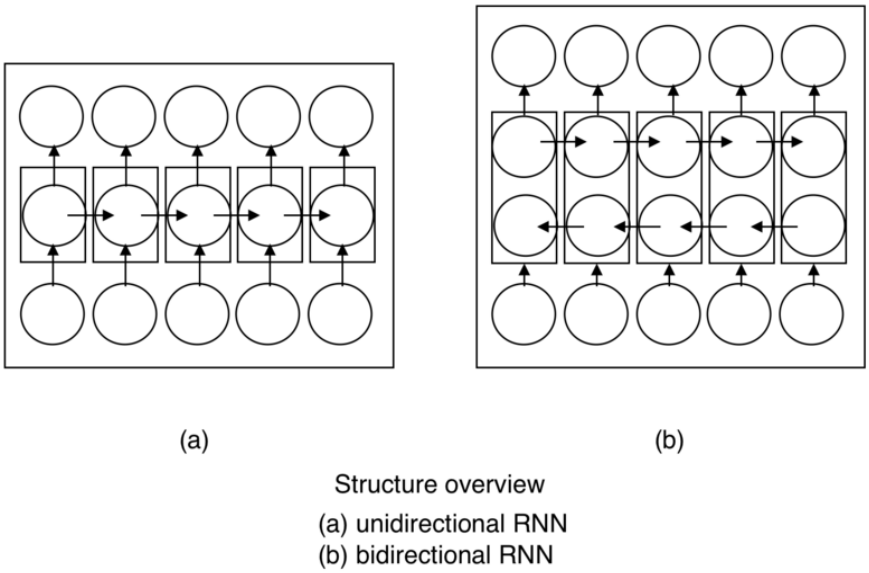
LSTM ağları, zaman serisi verilerine dayanarak sınıflandırmalar tahminler yapmak için çok uygundur, çünkü zaman serisi içindeki önemli olaylar arasında bilinmeyen süre gecikmeleri olabilir. LSTM'ler, geleneksel RNN'lerin eğitimi sırasında karşılaşılabilecek olan kaybolma problemleriyle başa çıkmak için geliştirilmiştir. Boşluk uzunluğuna bağıl duyarsızlık LSTM' in bir avantajıdır[9]. Uzun vadeli bağımlılıkları öğrenebilen özel bir RNN türüdür. Çok çeşitli problemlerde çok iyi çalıştıkları için günümüzde çok yaygın şekilde kullanılıyorlar.



Şekil 3: LSTM Yapısı (elitcenkalp.blogspot.com' dan alınmıştır.)

2.1.7. BIDIRECTIONAL LSTM YAPISI

İki Yönlü Tekrarlayan Sinir Ağları (BRNN), aynı çıkışa iki ters yöndeki gizli katmanı birbirine bağlar. BRNN'nin prensibi, normal bir RNN'nin nöronlarını, biri pozitif zaman yönü (ileri durumlar) ve diğeri negatif zaman yönü (geri durumlar) için iki yöne bölmektir. Bu iki durumun çıktısı, ters yön durumlarının girdilerine bağlı değildir. RNN ve BRNN'nin genel yapısı Şekil 4' de gösterilebilir[10].



Şekil 4: LSTM ve Bidirectional LSTM (RNN YAPILARI) [10]

2.1.8. GOOGLE COLAB' IN KULLANILMASI

Colab' ta Python programlama dilinde uygulama geliştirilebilir. Keras, TensorFlow, PyTorch ve OpenCV gibi kütüphaneleri kullanarak derin öğrenme (deep learning) uygulamaları yapılabilir. Colab'ı diğer ücretsiz bulut servislerinden ayıran en önemli özellik; Colab'ın ücretsiz GPU sağlamasıdır.[11] Hazırlanan datasetlerin eğitimi için Google Colab kullanılmıştır.

Googe Colab' ın özellikleri;

- CPU: Intel(R) Xenon(R) CPU @ 2.30 GHz
- GPU: NVIDIA Tesla K80
- Memory: 12.71 GB

Şekil 5: Google Colab [11]

3. BULGULAR

Bu bölümde oluşturulmuş veri kümesi ile yapılan çalışmalara ve bu çalışmaların verilerine vurgu yapılacaktır. Öncelikle veri kümesinin nasıl elde edildiğinden bahsetmek gerekir. Crawler yardımı ile makale, dergi ve gazete gibi düzgün yazım kurallarına sahip kaynaklardan farklı kategorilerde (tarih, siyaset, ekonomi, sağlık, spor, teknoloji, kültür sanat vb.) paragraflar bulunup, her paragraf ayrı dosyalara kaydedilmiştir. Python da yazılan basit bir fonksiyon ile bütün dosyalar tek bir dosyada birleştirilmiştir. Paragraflardan oluşan bu dosya cümlelere ayrılarak ".csv" uzantılı bir dosyaya kaydedilmiştir. Sonuç olarak 3610 paragraf ve 56.100 cümleden oluşan bir veri kümesi toplanmıştır. Toplanan bu veri kümesi sadece doğru imla kurallı cümlelerden oluşmaktadır.

Bu veri kümesi üzerinde yedi işlem gerçekleştirilmiştir.

1) Doğru imla kurallı veri kümesinin cümleleri belli bir yapı çerçevesinde anlamını yitirmeyecek şekilde bozulması ile yanlış imla kurallı veri kümesinin elde edilmesi.
2) Doğru ve yanlış imla kurallı veri kümelerinin “LSTM” yapsı ile “encode” edilmesi.
3) Encode edilmiş verilerin “decode” edilmesi.
4) Encode ve decode edilmiş verilerin input değerleri ile bir “Model” oluşturulması.
5) Oluşturulan Model’ in encode ve decode verileri ile eğitilmesi.
6) Model’ in kaydedilmesi ve doğruluk oranlarının hesaplanması.
7) Eğitilmiş modelin encode ve decode yapıları ile sonuçlarının tahmin edilmesi.

3.1. YANLIŞ İMLA KURALLI VERİ KÜMESİNİN OLUŞUMU

Python programlama dilinde doğru imla kurallı cümleler dosyadan okunarak kelime kelime işleme tabi tutulmuştur (Şekil 7).

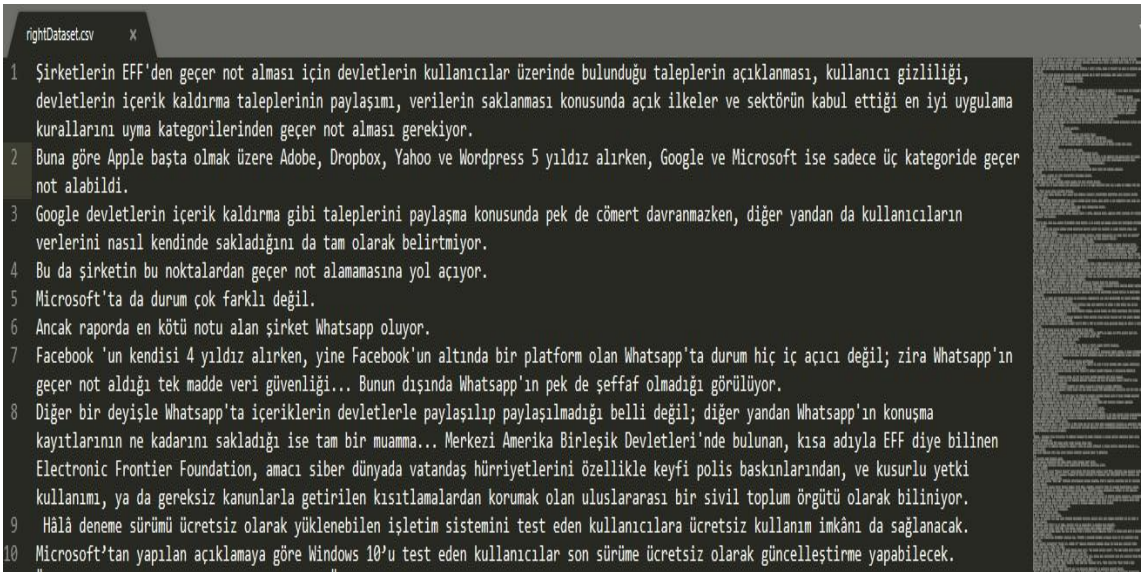
Bu işlemler:

1- Kelimenin içerisindeki harflerin klavye üzerindeki yakın harfler ile yer değiştirilmesi (Şekil 6) .

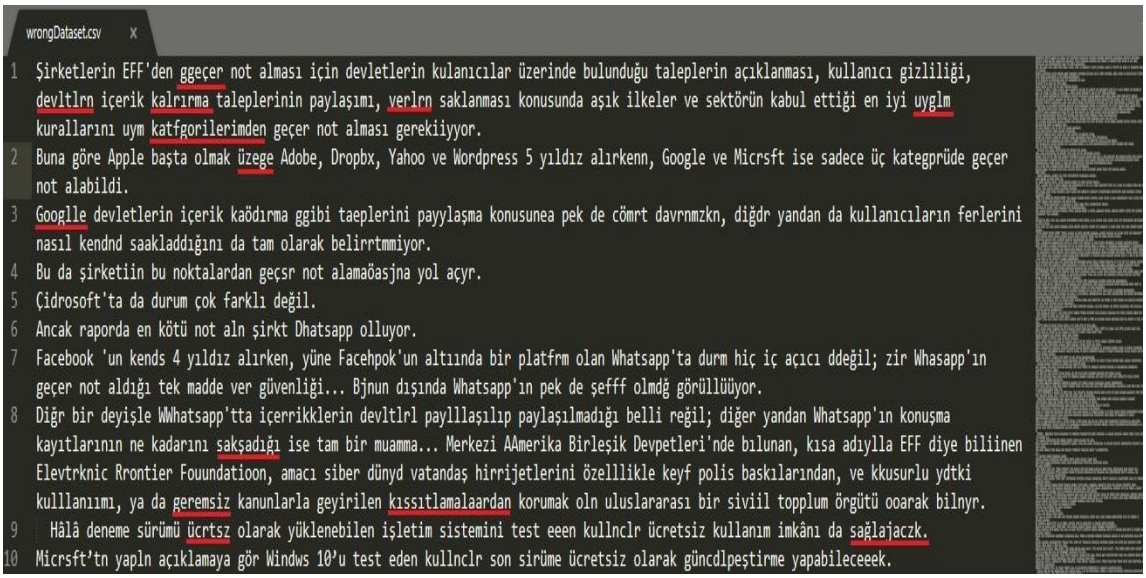
Şekil 6: Klavyede Yapılabilecek Hataları Gösteren Resim

2- Kelimenin uzunluğunun %15 kadar rastgele seçilen harflerin tekrar edecek şekilde eklenmesi.
3- Kelimenin içerisindeki ilk sesli harf hariç diğer sesli harflerin kelimeden çıkarılması.
4- Kelimenin uzunluğunun %15 kadar sessiz harfin kelimeden çıkarılması.
5- En az 4 ve üzeri karakterden oluşacak kelimelere yukarıdaki işlemlerin uygulanması. Daha az olanlarının doğrudan cümleye eklenmesi.

Bu işlemler tüm kelimelere uygulanmadan rastgele seçilerek belirlenen kelimelere uygulanmıştır. Uygulanan kelimelere hangi işlemin belirleneceği de aynı şekilde rastgele seçilmiştir (Şekil 8).



Şekil 7: Doğru İmla Kurallı Cümlelerden Oluşmuş Veri Kümesi



Şekil 8: Yanlış İmla Kurallı Cümlelerden Oluşmuş Veri Kümesi

3.2. LSTM YAPISI İLE ENCODE İŞLEMİ

Tablo 1: Veri Kümeleri İçerisindeki

a) Toplam kaç cümlenin eğitileceği.

b) Veri kümesindeki eşsiz (unique) karakterlerinsayısı.

c) Maksimum karakter sayısına sahip bir cümlenin uzunluğu.

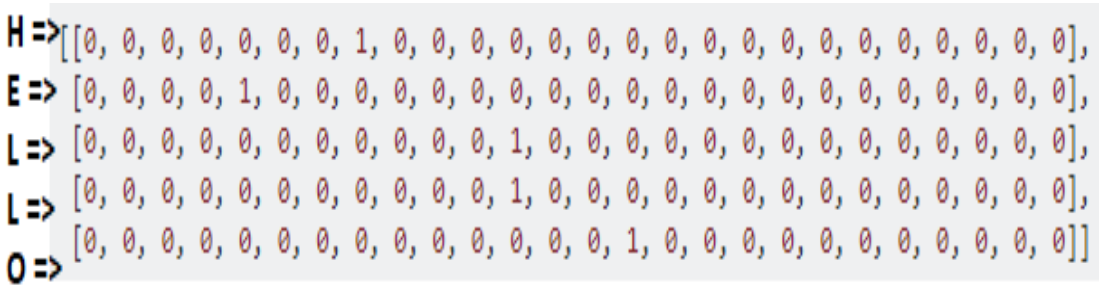
Number of Wrong Samples:	2000
Number of Right Samples:	2000
Number of unique input tokens:	111
Number of unique output tokens:	113
Max sequence length for inputs:	852
Max sequence length for outputs:	856

Tablo 1' de belirtilen değerler ile python programlamanın "numpy" kütüpanesinin "zeros" metodu ile veri kümesindeki toplam cümle adeti kadar en uzun cümle ve eşsiz (unique) karakterlerin uzunluğuna bağlı olarak, değerleri sıfır ("0") olan vektörler oluşturulur.

Örnek : "encoder_input_data = np.zeros((len(wrongSentences), max_encoder_seq_length, num_encoder_tokens), dtype='float32')"

Cümleler içerisinden belirlenen eşsiz karakterlere ve cümlenin içerisinde yer alma indeksine göre numpy.zeros metodu ile oluşturulan vektörlere 1 veya 0 atama işlemi yapılarak encode yapısı gerçekleştirilir. Bu yapı "One – Hot – Encode" yapısıdır.

Şekil 9' da görüldüğü gibi İngilizce bir kelimenin harflerinin nasıl "One – Hot – Encode" yapısına uygun olduğu gösterilmiştir. Tablo 2' de gösterilen yapıda ise belirlemiş olduğum yanlış imla kurallı bir kelimenin sadece ilk harfi olan "Ş" harfinin "One – Hot – Encode" yapısı gösterimi mevcuttur.



Şekil 9: One-Hot-Encode Yapısı (Stackoverflow' dan alınmıştır.)

Tablo 2: "Ş" Harfini Temsil Eden One - Hot - Encode Yapısı

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.,
       0., 0., 0., 0., 0., 0., 0., 0.], dtype=float32)
```

Yanlış imla kurallı cümlelerin oluşturduğu veri kümesinin eşsiz karakterleri ile "encoder input" oluşturulur. LSTM yapısına encoder input verilerek "encode output" ve "encode state" oluşturulur. Bu şekilde LSTM yapısı ile encode işlemi gerçekleştirilmiştir. Ayrıca deneme amaçlı LSTM ve Bidirectional LSTM yapıları kullanılmıştır[12].

3.3. LSTM YAPISI İLE DECODE İŞLEMİ

Tablo 1' de belirtilen değerler ile python programlamanın "numpy" kütüpanesinin "zeros" metodu ile veri kümesindeki toplam cümle adeti kadar en uzun cümle ve eşsiz (unique) karakterlerin uzunluğuna bağlı olarak, değerleri sıfır ("0") olan vektörler oluşturulur.

Örnek : "decoder_input_data = np.zeros((len(wrongSentences), max_decoder_seq_length, num_decoder_tokens), dtype='float32')"

Cümleler içerisinden belirlenen eşsiz karakterlere ve cümlenin içerisinde yer alma indeksine göre numpy.zeros metodu ile oluşturulan vektörlere 1 veya 0 atama işlemi yapılarak decode yapısı gerçekleştirilir.

Doğru imla kurallı cümlelerin oluşturduğu veri kümesinin eşsiz karakterleri ile "decoder input" oluşturulur. LSTM yapısına decoder input ve encoder state verilerek "decoder output" oluşturulur. Bu şekilde LSTM yapısı ile decode işlemi gerçekleştirilmiştir.

3.4. MODEL OLUŞTURULMASI

Model yapımızı oluşturmak için encoder input ve decoder input değerlerinin vektörü ve decoder output vektörü gerekmektedir.

Örnek: "model = Model([encoder_inputs, decoder_inputs], decoder_outputs)"

3.5. MODEL' İN EĞİTİLMESİ

Model oluşturulduktan sonra derlenilerek (compile) model yapısı oluşur ve eğitmeye hazır hale gelir. Daha sonra model.fit( ) fonksiyonu ile encode ve decode yapısına tabi tutulan cümleler eğitilmiştir (Tablo 3).
Eğitim aşamasında bilinmesi gereken terimler ve bu terimlerin yorumlanması aşağıdaki gibidir:

- **Eğitim Tur (Epoch) Sayısı:** Model eğitilirken verilerin tamamı aynı anda eğitime katılmaz. Belli sayıda parçalar halinde eğitimde yer alırlar. Eğitim adımlarının her birine "epoch" denilmektedir.
- **Batch Sayısı:** Model tasarlanırken batch parametresi olarak belirlenen değer; modelin aynı anda kaç veriyi işleyeceği anlamına gelmektedir.
- Epoch sayısı arttıkça modelin başarımı gözle görülür oranda artmaktadır.
- Batch size küçük olması iyileştirme (reguralization) etkisi yaratmaktadır. Modele veri büyük gruplar halinde verildiğinde ezberleme daha fazla olur.

Tablo 3: Eğitilecek ve Test Edilecek Verilerin Sayısı

Epoch, Loss,Accurancy,

Validation Loss ve Validation Accurancy

Değerlerini Gösteren Eğitim Tablosu

```
Train on 1000 samples, validate on 1000 samples
Epoch 1/200
1000/1000 [==============================] - 83s 83ms/step - loss: 0.5056 - acc: 0.0942 - val_loss: 0.5468 - val_acc: 0.0152
Epoch 2/200
1000/1000 [==============================] - 80s 80ms/step - loss: 0.4486 - acc: 0.0158 - val_loss: 0.5399 - val_acc: 0.0126
Epoch 3/200
1000/1000 [==============================] - 80s 80ms/step - loss: 0.4461 - acc: 0.0155 - val_loss: 0.5357 - val_acc: 0.0252
Epoch 4/200
1000/1000 [==============================] - 80s 80ms/step - loss: 0.4440 - acc: 0.0171 - val_loss: 0.5344 - val_acc: 0.0200
Epoch 5/200
1000/1000 [==============================] - 80s 80ms/step - loss: 0.4411 - acc: 0.0182 - val_loss: 0.5311 - val_acc: 0.0270

.                                          .                                          .
.                                          .                                          .
.                                          .                                          .

poch 195/200
1000/1000 [==============================] - 80s 80ms/step - loss: 0.0064 - acc: 0.1328 - val_loss: 0.6783 - val_acc: 0.0574
Epoch 196/200
1000/1000 [==============================] - 80s 80ms/step - loss: 0.0043 - acc: 0.1331 - val_loss: 0.6872 - val_acc: 0.0569
Epoch 197/200
1000/1000 [==============================] - 80s 80ms/step - loss: 0.0248 - acc: 0.1262 - val_loss: 0.6757 - val_acc: 0.0573
Epoch 198/200
1000/1000 [==============================] - 80s 80ms/step - loss: 0.0064 - acc: 0.1328 - val_loss: 0.6817 - val_acc: 0.0575
Epoch 199/200
1000/1000 [==============================] - 80s 80ms/step - loss: 0.0039 - acc: 0.1331 - val_loss: 0.6902 - val_acc: 0.0573
Epoch 200/200
1000/1000 [==============================] - 80s 80ms/step - loss: 0.0239 - acc: 0.1262 - val_loss: 0.6754 - val_acc: 0.0574
```

3.6. MODEL' İN KAYDEDİLMESİ

Tablo 4: Modelin Kaydedilmesi ve Modelin Yapısının Özet Bilgileri

```
model.save( 's2s_5750_LSTM.h5' )
model.summary( )
```

Model eğitildikten sonra bu eğitilen modeli belirli bir formattaki dosyaya kaydedilir (Tablo 4). Bu kaydedilmiş olan dosya tekrar çalıştırılabilir ve güncellenebilir.

Ayrıca ".h5" dosyasına kaydedilmesinin sebebi, .h5 dosyasının, Hiyerarşik Veri Formatında (HDF) kaydedilen bir veri dosyası olmasıdır. Bu dosya çok boyutlu bilimsel veri dizilerini içerir.

Tablo 5: Modelin Özet Bilgileri

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   Connected to
=================================================================
input_1 (InputLayer)         (None, None, 111)         0
_________________________________________________________________
bidirectional_1 (Bidirectional) [(None, 512), (None, 753664      input_1[0][0]
_________________________________________________________________
input_2 (InputLayer)         (None, None, 113)         0
_________________________________________________________________
concatenate_1 (Concatenate)  (None, 512)               0         bidirectional_1[0][1]
                                                                 bidirectional_1[0][3]
_________________________________________________________________
concatenate_2 (Concatenate)  (None, 512)               0         bidirectional_1[0][2]
                                                                 bidirectional_1[0][4]
_________________________________________________________________
lstm_2 (LSTM)                [(None, None, 512),       1282048   input_2[0][0]
                                                                 concatenate_1[0][0]
                                                                 concatenate_2[0][0]
_________________________________________________________________
dense_1 (Dense)              (None, None, 113)         57969     lstm_2[0][0]
=================================================================
Total params: 2,093,681
Trainable params: 2,093,681
Non-trainable params: 0
_________________________________________________________________
```

3.7. EĞİTİLMİŞ MODELİN ENCODE VE DECODE İLE TAHMİNİ

Tablo 6: Modelin Tahmini Sonuçları

- **Input sentence:** Şirketlerin EFF'den ggeçer not alması için devletlerin kulanıcılar üzerinde bulunduğu taleplerin açıklanması, kullanıcı gizliliği, devltlrn içerik kalrırma taleplerinin paylaşımı, verlrn saklanması konusunda aşık ilkeler ve sektörün kabul ettiği en iyi uyglm kurallarını uym katfgorilerimden geçer not alması gerekiiyyor. **Decoded sentence:** Şirketlerin EFF'den geler nok alması için devartelerine konulacaklara için tanatıla gerekildi.
- **Input sentence:** Buna göre Apple başta olmak üzege Adobe, Dropbx, Yahoo ve Wordpress 5 yıldız alırkenn, Google ve Micrsft ise sadece üç kategprüde geçer not alabildi. **Decoded sentence:** Buna göre Appler aht olası kamesi yaban etkiye bir ünümet olar kabul değildi.
- **Input sentence:** Googlle devletlerin içerik kaödırma ggibi taeplerini payylaşma konusunea pek de cömrt davrnmzkn, diğdr yandan da kullanıcıların ferlerini nasıl kendnd saakladdığını da tam olarak belirrtmmiyor. **Decoded sentence:** Google devletlerin içerik kaldırma gibi taleplerini paylaşma konusunda pek de cömert davranmazken, diğer yandan da kullanıcıların verlerini nasıl kendinde sakladığını da tam olarak belirtmiyor.
- **Input sentence:** Bu da şirketiin bu noktalardan geçsr not alamaöasjna yol açyr. **Decoded sentence:** Bu da şirketin bu noktalardan geçen not alamadam be balara olarak yaksızı oynasınla iygelemde yaşındı.
- **Input sentence:** Çidrosoft'ta da durum çok farklı değil. **Decoded sentence:** Microsoft'ta da durum çok farklı değil.

-

Input sentence: Bu ihbarı hızlı bir biççimde yyapılabiilmesi için telefonun IMEI nnumaarasının bilinmesi büüyük önm taşımaakktadır.

Decoded sentence: Bu ihbarı hızlı bir biçimde yapılabilmesi için telefonun IMEI numarasının bilinmesi büyük önem taşımaktadır.

-

Input sentence: Cihazın üzerine bulnn kendi SİM kartınızın kullnlmsn egellemek için ise GSM oprtrnz aryrk hattnz da kapttrmnz gerekmektedir.

Decoded sentence: Cihazın üzerinde bulunan kendi SİM kartınızın kullanılmasını engellemek için ise GSM operatörünüzü arayarak hattınızı da kapattırmanız gerekmektedir.

-

Input sentence: Bayatly anlaşmanın mali detylryl ilgili bilgi vermedi.

Decoded sentence: Bayatly anlaşmanın mali detaylarıyla ilgili bilgi vermedi.

-

Input sentence: Hırsılık, gaep ve lapkaç gib ddurumlar söz konısu is savcılıklara, ikametgahınnıza veya olajın gerçekleştiği ilçe emjiyet müdürlüğüne başvuruarak şikayetli olunmması önnem taşıimaaktadır.

Decoded sentence: Hırsızlık, gasp ve kapkaç gibi durumlar söz konusu ise savcılıklara, ikametgahınıza veya olayı.

-

Input sentence: Cihazı tekrar kendşsine iltln vey bulunan kişlrn telefounu hangi yolla kapattırdı ise (bilggi ve ihbar jerkezi veya sabcılık üzeriden) aunı yoll tekfar iöetişime açtırası geerrekiyor.

Decoded sentence: Cihazı tekrar kendisyar gelişmelirden doğuz ve bir kanmılma sarıklayan yaram sedim, yünde 75 senilekinde onundu.

-

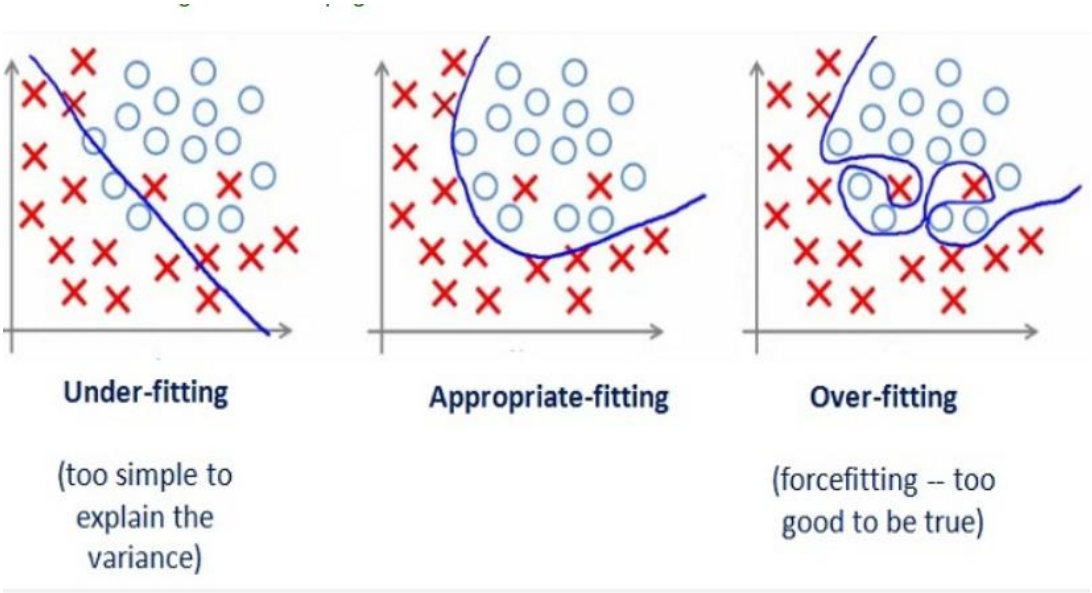
Input sentence: Yan telefon ile ihbada bulunulduysa yinne Bilg İhbar Merkezinin aranması, savccılık talimwtı ile inbarda bulunulduysa cihz bylununca yin saccılığa başvrlms gerekmektedir.”

Decoded sentence: Yani telefon ile ihbarda bulunulduysa için savatçılak dermeklenilerini de be krani olan karakladır. denişi daşı açılla ESSİy, 'din ve arşamanda olarak da ber olduğunu bilirti.

4. TARTIŞMA VE SONUÇ

İmla Hatalı Cümleleri Düzelten Sistem projesinin tamamlanması ile birlikte Doğal Dil İşleme dünyasında Türkçe için bir katkıya sahip olacaktır. Kullanım amacı doğrultusunda, iş hayatında yoğun ve kısıtlı zamana sahip olan insanların metinlerini düzenlemesi için bir alternatif olarak kullanılabilir.

Mevcut sistemin kurulması aşamasında birçok problem ile karşılaşılmıştır bu problemlerden en önemlileri eğitim sırasında büyük miktarda verinin aynı anda eğitilememesi ve eğitilen bölümde aşırı uyumun veya düşük uyumun gerçekleşmesidir. (Şekil 10)



Şekil 10: Eğitilmiş Verinin Uyumluluk Olasılıkları (medium.com' dan alıntı yapılmıştır.).

Aşırı ve düşük uyumluluk problemleri "epoch" ve "batch-size" değerleri takip edilerek ve optimal değerlere karar verilerek düzenlenmiştir. Epoch sayısı arttıkça modelin başarımı gözle görülür oranda artmaktadır. Batch-size küçük olması iyileştirme (reguralization) etkisi yaratmaktadır.

Modele veri büyük gruplar halinde verildiğinde ezberleme daha fazla meydana gelir ve bu yüzden tahmin gerçekleşmeyebilir. Bölüm 3.7.' de belirtilen verileri incelemek gerekirse, hiçbir hatalı cümlenin tamamen doğru bir şekilde düzeltilmediği fakat doğru cümleye çok yakın sonuçlar elde edildiği görülebilir.

Fakat eğitilmiş verinin doğruluk oranına ve hata oranına bakılırsa, bu eğitimin çokta sağlıklı bir eğitim olduğu söylenemez çünkü hata oranı azalmasına rağmen doğruluk oranları çok ufak miktarlarda artış göstermekle birlikte birbirlerine oldukça yakın değerler sergilemişlerdir.

KAYNAKLAR

[1] Altuğ TATLI, 25 Kasım 2014, "http://liveaplus.com/2014/11/is-hayatinda-ve-internette-kullanilan-kisaltmalar-nelerdir.html"

[2]"https://www.google.com.tr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiejOrfyv7QAhVWF8AKHX3XD2MQFggiMAA&url=https%3A%2F%2Fwww.macs.hw.ac.uk%2F~dwcorne%2FTeaching%2Fintrodl.ppt&usg=AFQjCNHaeX3OEmO_FNeCUR_32TWKyg3JXQ&sig2=EjSG8imBwPnVn8qE-nLvkA "

[3] "http://www.bb.itu.edu.tr/arastirma/dogal-dil-isleme"

[4] Margaret Rouse, Mayıs 2018, " https://searchenterpriseai . techtarget . com /definition/machine-learning-ML"

[5] " https://faroit.github.io/keras-docs/1.2.0/backend/ "

[6] " https://www.quora.com/What-is-an-Encoder-Decoder-in-Deep-Learning "

[7] " https://blog.algorithmia.com/introduction-natural-language-processing-nlp/"

[8] " https://www. wikizeroo. net/index.php?q =aHR0cHM6Ly9lbi 53 aW tpc GV k a Eub3JnL3dpa2kvTWFjaGluZV90cmFuc2xhdGlvbg"

[9]"http://www.wikizeroo.net/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvTG9uZ19zaG9ydC10ZXJtX21lbW9yeQ"
[10]http://www.wikizeroo.net/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvQmlkaXJlY3Rpb25hbF9yZWN1cnJlbnRfbmV1cmFsX25ldHdvcmtz

[11] https://colab.research.google.com/notebooks/welcome.ipynb

[12] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, Yoshua Bengio, 7 Oct 2014, "**On the Properties of Neural Machine Translation: Encoder-Decoder Approaches**"

[13] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, 19 May 2016, "**Neural Machine Translation by Jointly Learning to Align and Translate**"

[14] Yann LeCun, Yoshua Bengio & Geoffrey Hinton, 28 May 2015, "**Deep learning**"

[15] Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria, Aug. 2018, "**Recent Trends in Deep Learning Based Natural Language Processing**"