



DÖNEM SONU RAPORU

Derin Öğrenme İle Twitter Platformunda Yalan Haber Tespiti

Cansu AYTEN – 171180010

Gamze AKSU – 171180005

BM495 BİLGİSAYAR PROJESİ I

OCAK 2022

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER	1
KISALTMALAR.....	2
ÖZET	3
1. GİRİŞ	1
2. LİTERATÜR TARAMASI	2
3. YAPILAN ÇALIŞMALAR.....	4
3.1. Metodoloji	4
3.1.1. Platform ve Teknolojiler.....	4
3.1.2. Veri Seti.....	5
3.1.3. Verilerin Format Değişimi.....	5
3.1.4. Verilerin İncelenmesi	6
3.1.5. Veri Önışlemleri.....	8
3.1.6. Metin Vektörizasyonu	9
3.1.6.1. Word2Vec.....	9
3.1.6.2. FastText	10
3.1.6.3. Count Vectorizer	10
3.1.6.4. Hashing Vectorizer.....	10
3.1.7. Tokenizasyon	11
3.1.8. Modeller İçerisinde Kullanılan Yapılar.....	11
3.1.8.2 Bidirectional LSTM	11
3.1.9. Arayüz	12
3.2 Sonuçlar	13
4. TARTIŞMA VE YORUM	14
5. SONUÇ	14
KAYNAKÇA	16

KISALTMALAR

Kısaltmalar	Açıklamalar
GUI	Graphical User Interface
IDE	Integrated Development Environment
NLP	Natural Language Processing
TF-IDF	Term Frequency - Inverse Document
Frequency	
W2V	Word2Vec
SVM	Support Vector Machine
LSTM	Long Short-Term Memory
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
APS	Attentive Pooling Similarity
Bi-LSTM	Bidirectional LSTM
BCNN	Bidirectional CNN
RNN	Recurrent Neural Network
GloVe	Global Vectors
DSSM	Deep Semantic Similarity Model

ÖZET

Twitter platformu içerisinde bulunan haber içerikli tweetlerin sahte ya da gerçek olduğunun sınıflandırmasını yaparak bilgi kirliliğini önlemek amacıyla Twitter’da Gerçek-Sahte Haber Tespit projesi geliştirilmiştir. Bu projenin kapsamında veri seti olarak PHEME veri seti seçilmiştir. Verilerin format değişikliği ve verilere uygulanan ön işlemler açıklanmıştır. Daha sonra Word2Vec, FastText, Count Vectorizer ve Hashing Vectorizer yöntemleri kullanılmıştır. Modeller içerisinde LSTM ve Bidirectional LSTM modelleri kullanılmıştır. Modeller arasında en yüksek doğruluk değerine sahip olan model Bidirectional LSTM modelidir. Bu model kaydedilerek ara yüz için yapılan kodlama içerisinde bu modelden kaynak olarak yararlanılmıştır.

1. GİRİŞ

Teknolojinin günden güne gelişmesi ve gelişmeye devam etmesi, küçükten büyüğe herkesin en az bir akıllı cihaza sahip olma ihtiyacını da beraberinde getirmiştir. Akıllı cihazlar sayesinde günün herhangi bir anında akıllı cihazın kapasitesi dâhilinde istenilen bir işlem gerçekleştirilebilir. Bu işlemler arasında banka işlemleri, oyun ve eğlence, alışveriş, haber takipleri ve iletişim gibi işlemler bulunmaktadır. Her an herkesin elinde bulunan bu akıllı cihazlar ile İnternet'e olan ilgi de artmıştır. Bu da beraberinde bilgi ve haber kaynaklarının internet ortamında çoğalmasını sağlamıştır. İnsanların bilgiye erişim için kullandığı geleneksel araçlar olan gazete, radyo veya dergi gibi kaynakların yerini artık internet almıştır. İnternet kullanımının yaygınlaşmasıyla birlikte sosyal medya platformları da popüler hale gelmiştir. İnsanlar kolaylıkla ulaşabildikleri bu platformlar sayesinde gerçek zamanlı olarak yaşadıkları ya da duydukları olayları başkalarıyla paylaşabilirler. Sosyal medyanın insanların hayatının bir parçası haline gelmesiyle beraber haber ajanslarının da bu platformlarda bir hesap sahibi olup paylaştıkları haberleri duyurması kaçınılmaz olmuştur. Böylelikle hem insanlardan hem de haber ajansları tarafından paylaşılan bazen kesin olarak doğru olan bazen ise asılsız olan bazı haberler sosyal medya içerisinde hızla yayılabilir. Doğru olmayan haberlerin hızla yayılması bilgi kirliliğine neden olmaktadır. Yanlış bir bilgi bu platformlarda coğrafi açıdan sınırsız olarak paylaşıldığı için tüm dünyada etki yaratabilir. Örneğin Amerika’da bulunan bir haber ajansının Twitter hesabının ele geçirilmesi ile 2013 yılında bir son dakika haberi olarak Beyaz Saray’da iki patlamanın meydana geldiği ve Barack Obama’nın yaralandığı bilgisinin bulunduğu bir tweet atılmıştır. Bu tweet atılmasının ardından 5 dakikadan kısa bir süre içerisinde 4 binden fazla kez paylaşılmıştır. Her ne kadar bu haber ajansı hesabının ele geçirildiğini ve yayılan bu haberin sahte bir haber olduğunu açıklansa da haberin gerçek olma olasılığının yani ABD başkanının bir patlamada ölme olasılığının olması borsayı bir çöküşe uğratmıştır(Demir, 2021). Örnekte de görüldüğü üzere Twitter, haberlerin en hızlı yayıldığı sosyal medya platformlarından biridir. Bu platformda kullanıcılar kendilerine ait hesaplarda en fazla 280 karakter uzunluğundaki metinleri paylaşabilmektedirler. Ülkelere, dünyaya ve kişiye göre oluşturulmuş gündem ile o an içerisinde çok fazla konuşulan belirli sayıdaki başlıklara ulaşp bu başlıkların içerisinde yer alan tweetler görüntülenebilir. Bu şekilde gündemde olan herhangi bir olay daha hızlı şekilde insanlar arasında yayılabilir. Doğru olmayan haberlerin bu şekilde büyük bir hızla yayılması insanları, hayatı, ekonomi veya siyasi açıdan ülkeleri çok fazla etkileyebilmekle birlikte bir kargaşa ortamına sebep olabilmektedir.

Twitter’deki yalan haber barındıran tweetlerin zararlarının önlenmesi amacıyla kullanıcıların bir tweetin sahte ya da gerçek olduğunu öğrenebileceği bir sisteme ihtiyacı vardır. Bu soruna alternatif bir çözüm olarak geliştirilen Twitter’da Gerçek-Sahte Haber Tespit Projesi içerisinde haber içerikli bir tweetin gerçek ya da sahte olduğunu saptayabilen bir mekanizmaya sahiptir. Bu proje kapsamında Twitter’dan elde edilen haber verileri doğal dil işleme teknikleri kullanılarak işlenmektedir. Daha sonra derin öğrenme ile oluşturulan model bu veriler ile eğitilmektedir. Eğitilen modelin Twitter’dan alınan bir haberin gerçek ya da yalan bir haber olup olmadığını ayırt edebilmesi beklenmektedir.

2. LİTERATÜR TARAMASI

Son zamanlarda akıllı cihazların yaygınlaşmasıyla birlikte insanlar haberlere gazete, dergi gibi araçlarla ulaşmak yerine akıllı cihazları kullanarak daha hızlı ulaşım sağlamaktadırlar. Bu durum yalnızca hızlı ulaşmakla sınırlı kalmaz haberlerin hızlı yayılımına da sebep olur. Herkesin bilgiye kolaylıkla ulaşması ve bu bilgilerin kolaylıkla yayılması yalan haberlerin sayısını ve yayılımını da artırmıştır. Yalan haberlerin yayılımının artmasıyla birlikte toplumda olan kötü etkisi de kaçınılmaz olmuştur. Bu nedenle etkiyi azaltmak amacıyla yalan haberler bir sistemle tespit edilmelidir. Son zamanlarda yalan haber tespiti amacıyla birçok çalışma gerçekleştirilmiştir. Bu çalışmaların çoğunun içerisinde doğal dil işleme teknikleri, makine öğrenmesi algoritmaları ve derin öğrenme modellerine yer verilmiştir. Aşağıda bu çalışmalardan birkaçından bahsedilmiştir.

Vijayaraghavan vd. (2020) çalışmalarında verilerini ilk olarak bazı ön işlemlerden geçirmişlerdir. Bu ön işlemler verilerden sayıların, noktalama işaretlerinin, özel karakterlerin ve etkisiz kelimelerin kaldırılmasını kapsamaktadır. TF-IDFVectorizer, CountVectorizer (CV) ve Word2Vec (W2V) gibi üç farklı vektörleştirme modeli kullanmışlardır. Sınıflandırma işlemi için Lojistik Regresyon (LR), Support Vector Machine Classifier (SVM), Random Forest, Yapay Sinir Ağları (ANN) ve LSTM kullanmışlardır. K-fold cross validation yöntemini kullanıp k'ya üç değeri verilmiştir. Hiperparametre optimizasyonu yapmak amacıyla bazı parametrelerin denenebilmesi için Grid Search tekniğinden yararlanılmıştır ve böylelikle en optimum hiperparametrelere ulaşılmıştır. Count Vectorizer diğerler vektörleştirme modellerinden daha iyi sonuç verirken Word2Vec düşük sonuçlar vermiştir. Count Vectorizer ve LSTM kombinasyonu diğer tüm modeller arasında en iyi sonuçları vermiştir. Bunun yanında TF-IDF için LR'nin sinir ağlarına göre daha iyi bir performansa sahip olduğu söylenebilir(Vijayaraghavan ve diğerleri,2020).

Lee vd. (2019) çalışmalarında Word2Vec ve FastText modelleri kullanmışlardır. BCNN, LSTM-BCNN, Bi-LSTM-BCNN ve Attentive Pooling Similarity ile BCNN (APS-BCNN) modelleri oluşturmuşlardır. Çalışma içerisinde iki farklı görev belirlemişlerdir. Bunlardan Görev 1 başlık ve gövdenin tutarsız olduğu durumlar için ve Görev 2 gövde ve içeriğinin bağlamla alakasız olduğu durumlar olarak belirlemişlerdir. Görev 1 için sonuçlar düşük çıkarken Görev 2 için yeterli bir doğruluk oranı ortaya çıkmıştır. FastText sonuçları Word2Vec sonuçlarından daha iyi çıkmıştır. Çünkü bu çalışma Korece üzerinde gerçekleştirilmiştir. Korece'de her bir hecenin kendi anlamı olmakla birlikte birleştiklerinde ise farklı bir kelime oluşturduğu için FastText bu dil üzerinde daha iyi sonuç vermiştir(Lee ve diğerleri, 2019).

Bahad vd. (2019) çalışmalarında haber gövdesi ve başlığı arasındaki ilişkiyi ölçmek için GloVe modeli kullanmışlardır. Embedding katmanı içinde rastgele ağırlıklar ile başlamak yerine GloVe'dan ağırlıklar eklenmiştir. CNN, Vanilla RNN, Unidirectional LSTM-RNN ve Bidirectional LSTM-RNN modelleri kullanılarak iki farklı veri seti eğitilmiştir. Performans metrikleri arasından doğruluk metriği baz alınarak sonuçlar karşılaştırılmıştır. Eğitimler sonrasında en iyi sonucu Bidirectional LSTM-RNN vermiştir(Bahad,KimY.,KimH.,Park,Yang, 2019).

Jadhav vd (2019) çalışmalarında LIAR veri setinde eğitim için DSSM, LSTM ve DSSM-LSTM kullanmışlardır. Ön işleme aşamasında tokenizasyon ve stemming işlemlerinden sonra TF-IDF Vectorizer ve Count Vectorizer kullanmışlardır. Verileri eğitim ve test olarak üç farklı oranda ayırıp üç farklı şekilde eğitim yapılmıştır. Eğitimler sonucunda doğruluk metriğini baz alarak en yüksek performansı %75-25 dağılımına sahip veriler için DSSM-LSTM modeli vermiştir. Model %99 doğruluk oranıyla sahte haber tespiti yapar(Jadhav ve Tepade, 2019).

Wang (2017) çalışmasında LIAR veri setini kullanmıştır. Bu çalışmada Lojistik Regresyon (LR), Convolution Neural Network (CNN) model, Long Short-Term Memory (LSTM) ve Support Vector Machine Classifier (SVM) gibi yaklaşımlar kullanarak bir sınıflandırma gerçekleştirmiştir. Çalışma içerisinde meta-verilerin de metinler ile birlikte modele verilmesi sonucu modelin daha iyi değerler verebileceği sonucuna varmıştır. Bunlarla birlikte önceden

eğitilmiş 300 boyutlu bir Word2Vec kullanmıştır. Grid Search tekniği kullanılarak SVM ve LR için hiperparametreler optimize edilmeye çalışılmıştır. LIAR veri seti düzgün dağılıma sahip olan bir veri seti olduğu için f-measure değerine eş değer görüldüğünden doğruluk değerine bakılarak sonuçlara varmışlardır. Bidirectional LSTM overfit olduğu için yapılan iyileştirmeler sonucunda CNN modelinin 0.27 ile en iyi sonuç verdiği görülmüştür(Wang, 2017).

3. YAPILAN ÇALIŞMALAR

3.1. Metodoloji

3.1.1. Platform ve Teknolojiler

Twitter’da Gerçek-Sahte Haber Tespit Projesi içerisinde PHEME veri setinin kullanılmıştır. Bu veri seti .json formatında verilere sahiptir. Proje içerisinde PHEME veri setinin kullanılabilmesi için .csv formatına dönüştürülmesi gerekmektedir. Bu işlem için Jupyter Notebook kullanılmıştır. Jupyter Notebook açık kaynak kodlu bir yazılım olmakla birlikte kullanıcı dostu bir platformdur. Python, R ve Julia programlama dillerini destekler. Jupyter Notebook içerisinde notlar alınabilen bir markdown'lar kolaylıkla oluşturulabilir. Ayrıca markdown ile başlıklandırma da yapılarak düzenli bir kodlama imkanı sunar(“Project Jupyter,” 2021).

Veriler .csv formatında bir dosyada toplandıktan sonra Google Colab platformuna geçilmiştir. Google Colab, Google’ın geliştiricilere sunduğu ücretsiz bir bulut servisi. Google Drive ile entegre bir yapıdır. Drive’da bulunan bir dosyaya kod içerisinde erişebilmemize imkan sağlar. Python kodunun Jupyter Notebook benzeri bir ortamda çalıştırılmasına olanak tanır. Jupyter Notebook gibi markdown yapısına sahip olduğu için benzer şekilde içerisinde not tutabilme imkanı sağlar. Colab, içerisinde bazı hazır kütüphanelere sahiptir. Bu kütüphaneleri indirmeye gerek kalmadan kolayca kullanımına olanak tanır. Colab içerisinde oluşturulan Notebooklar belirli kısıtlamalar altında belirli kişilere düzenleme imkanı sağlayan bir paylaşma yapısına da sahiptir.

Proje içerisindeki arayüz işlemleri Spyder IDE’si kullanılarak yapılmıştır. Çünkü ara yüz oluşturmak için kullanılan tkinter kütüphanesini Google Colab desteklememektedir. Spyder IDE’si Python programlama dilini destekleyen açık kaynak kodlu bir platformdur. Spyder GUI oluşturmak için Qt kullanır(“Spyder (Software),” 2021).

Proje içerisinde .json dosyası formatında olan verileri .csv formatına dönüştürmek için json ve os kütüphanelerinden faydalanılmıştır. Verilerin bir dataframe'de toplanması ve .csv formatında indirilebilmesi için Pandas kütüphanesinden yararlanılmıştır. Daha sonraki aşamada bu verileri inceleme, görselleştirme ve ön işleme işlemleri için Matplotlib, Seaborn, Wordcloud, NLTK, String ve Gensim kütüphaneleri, model oluşturma ve eğitme işlemleri için ise Sklearn, Tensorflow ve Keras kütüphanelerinden yararlanılmıştır.

3.1.2. Veri Seti

Proje içerisinde PHEME veri seti kullanılmıştır . Bu veri seti içerisinde 6425 kayıt bulunmaktadır. Bu kayıtlar dokuz farklı haber başlığı altında toplanmıştır. Bu haber başlıkları; Charlie Hebdo, Ebola Essien, Ferguson, Germanwings Crash, Gurlitt, Ottawa Shooting, Prince Toronto, Putin Missing ve Sydney Siege'dir. Bulundurdukları gerçek, sahte ve tweet sayıları aşağıdaki gibidir:

- Charlie Hebdo: 1621 gerçek, 458 sahte tweet
- Ebola Essien: 0 gerçek, 14 sahte tweet
- Ferguson: 859 gerçek, 284 sahte tweet
- Germanwings Crash: 230 gerçek, 238 sahte, 1 belirsiz tweet
- Gurlitt: 77 gerçek, 61 sahte tweet
- Ottawa Shooting: 420 gerçek, 470 sahte tweet
- Prince Toronto: 4 gerçek, 229 sahte tweet
- Putin Missing: 112 gerçek, 126 sahte tweet
- Sydney Siege: 699 gerçek, 522 sahte tweet

Bu haberler içerisinde sahte, gerçek ve belirsiz olmak üzere üç farklı haber türü mevcuttur. Veriler içerisinde toplam sahte haber sayısı 2402, toplam gerçek haber sayısı 4022 ve toplam belirsiz haber sayısı 1'dir. Her bir haber başlığı için farklı bir dizin oluşturulmuştur. Bu dizinler içerisinde her bir tweet için de farklı bir dizin oluşturulmuştur. Dizinler içindeki veriler .json dosyası olarak depolanmıştır. Dizin içerisinde tweetin kendisi "source-tweet" dizininde bulunur. Bu tweete verilen cevaplar ise "reactions" dizininde bulunur. Bu proje kapsamında reactions dizininde bulunan tweetler kullanılmamıştır. Tweetlere ait sahte, gerçek haber ya da belirsizlik bilgisi ise "annotation.json" dosyasında bulunmaktadır.

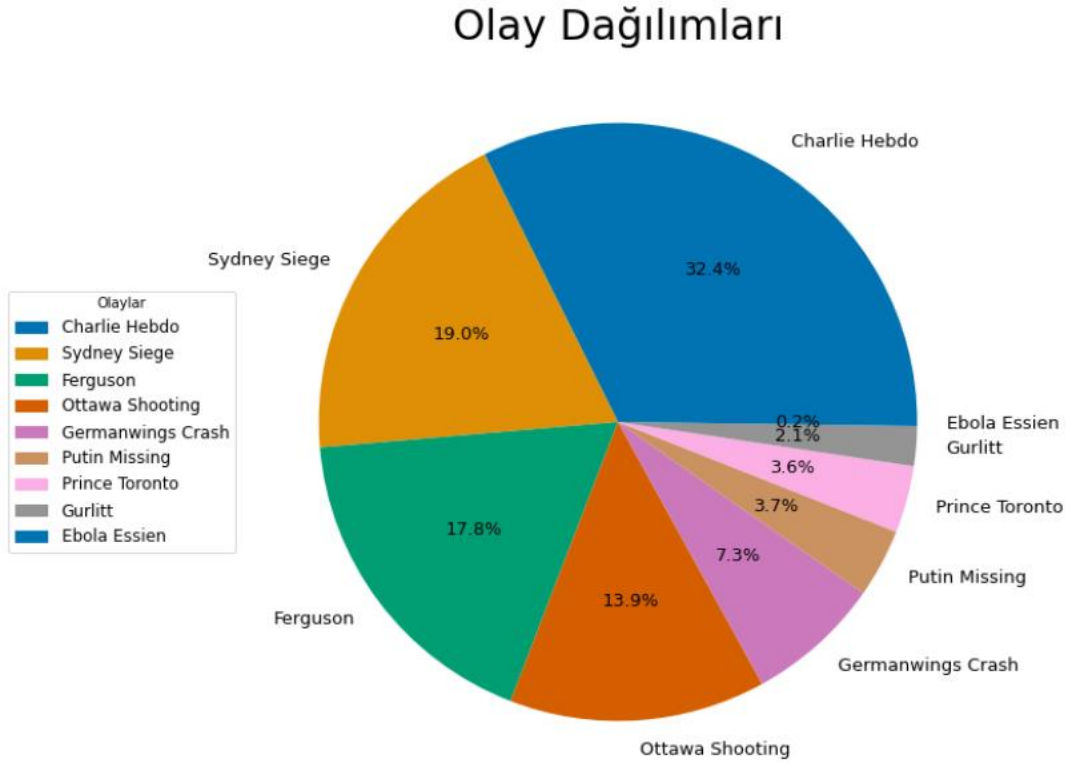
3.1.3. Verilerin Format Değişimi

Veri seti birçok farklı dizin ve dosya bulunmaktadır. Bu durum karmaşıklık yaratmaktadır. Bunun yanında veri seti içerisindeki tüm dizinler ve attribute'lar proje içerisinde

kullanılmayacaktır. Oluşan karmaşıklıkları gidermek amacıyla proje içerisinde yalnızca “source-tweet” dizininde bulunan tweetlerin text attribute’u ile “annotation.json” dosyası içerisinde bulunan is_rumor attribute’u kullanılmıştır. Diğer aşamalarda daha kolay bir kullanım için bu veriler toplanıp bir .csv dosyasında birleştirilmiştir. Ayrıca konu başlıklarındaki olayların incelenmesi için bu olaylar da dosya içerisine eklenmiştir.

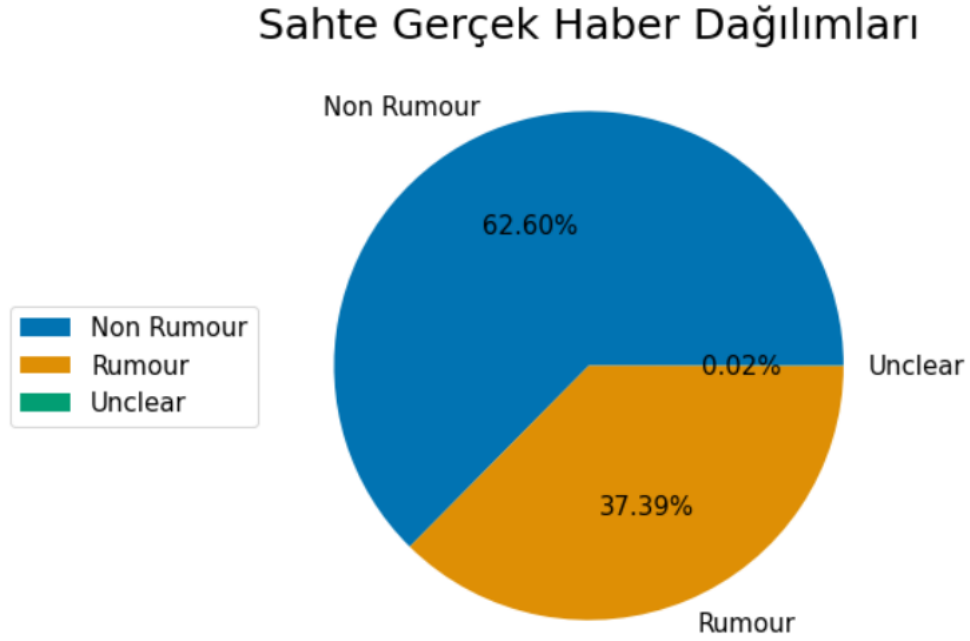
3.1.4. Verilerin İncelenmesi

Verilerin incelenmesi aşaması Google Colab ortamında gerçekleştirilmiştir. Bu yüzden veriler drive’a kaydedilmiştir. Veriler okunduktan sonra olayların dağılımını incelemek için olaylar pasta grafiği ve sütun grafiği olarak görselleştirilmiştir. Olay dağılımlarına ait pasta grafiği Şekil 1’deki gibidir. Haberlerin büyük çoğunluğu Charlie Hebdo olayı içerisinde bulunmaktadır. Bu başlıktaki veriler tüm verilerin %32.4’ünü oluşturmaktadır. Veri setine en az katkı sağlayan olay ise Ebola Essien olayıdır. Bu başlıktaki veriler tüm verilerin %0.2’sini oluşturmaktadır. Konu başlıklarına göre haber dağılımlarına bakıldığında düzgün bir dağılım olmadığı görülmektedir. En çok haber bulunduran Charlie Hebdo başlığı ile en az haber bulunduran Ebola Essien başlığı arasındaki haber sayısı farkının çok fazla olduğu anlaşılmaktadır.



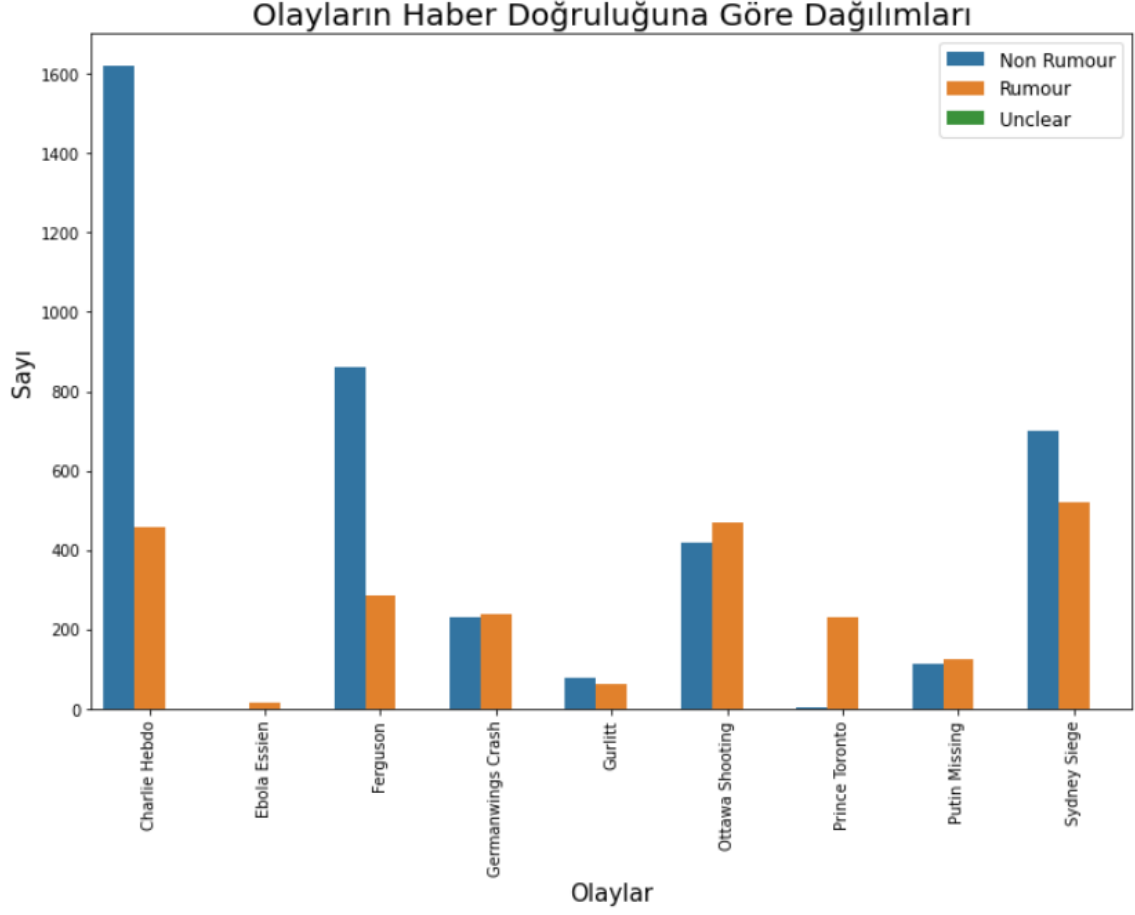
Şekil 1: Olay Dağılımları

Veriler içerisindeki sahte, gerçek ve belirsiz haberlerin sayılarının dağılımını incelemek amacıyla pasta grafiği ve sütun grafiği oluşturulmuştur. Bu grafiklerden pasta grafiği Şekil 2’deki gibidir. Grafikte verilerin büyük çoğunluğunun gerçek haberlerden oluştuğu görülmektedir. Gerçek haberler tüm haberlerin %62.6’sını oluşturmaktadır. Veriler içerisindeki yalan haber sayısı tüm haberlerin %37.4’ünü oluşturmaktadır. Veriler içerisinde bulunan bir tane belirsiz haber tüm haberlerin %0.02’sini oluşturmaktadır. Bu belirsiz haber proje içerisinde silinmiştir.



Şekil 2: Sahte Gerçek Haber Dağılımları

Sahte ve gerçek haberlerin olaylar içerisindeki dağılımlarını görmek amacıyla bir sütun grafiği oluşturulmuştur. Bu grafikte olayların haber doğruluğuna göre dağılımları Şekil 3’te gösterilmiştir. Burada en çok haber sayısına sahip olan konu başlığı Charlie Hebdo olmakla birlikte en çok doğru haber sayısı da Charlie Hebdo konu başlığı altında bulunmaktadır. Germanwings Crash, Gurlitt, Ottawa Shooting ve Putin Missing konu başlıkları altında bulunan haberlerin gerçek ve sahte haber sayıları birbirine yakın gelmiştir. Bu durum o başlıkların dengeli dağılıma sahip olduklarını göstermektedir. Charlie Hebdo, Ferguson ve Prince Toronto haber başlıklarının altındaki haberlerin sahte ve gerçek haber sayıları arasında büyük bir fark olduğu için dengeli dağılıma sahip değildir. Ebola Essien haber başlığı altında neredeyse hiç haber bulunmamaktadır. Bu olayın veri setine olan etkisi çok azdır.



Şekil 3: Olayların Haber Doğruluğuna Göre Dağılımları

3.1.5. Veri Önışlemleri

Twitter’da Gerçek-Sahte Haber Tespit Projesi için kullanılan veri setindeki tweetlere bir takım ön işlemler uygulanmıştır. Bu önışlemler şu şekilde sıralanabilir:

1. Noktalama kaldırma işlemi
2. Tokenizasyon işlemi
3. Tüm harflerin küçük harfe dönüştürülmesi işlemi
4. Stopword kaldırma işlemi
5. Linkleri kaldırma işlemi
6. Kelimeleri köklerine dönüştürülmesi işlemi

Öncelikle veriler içerisindeki noktalama işaretleri kaldırılmıştır. Bunun nedeni noktalama işaretleri kelimelere dâhil değildir. Bunun sonucunda noktalama işaretleri verilerde gürültüye neden olmaktadır. Gürültülü veri, veri içerisindeki hatalar olarak açıklanabilir. Bu nedenle bu hataların veriden çıkarılması gerekmektedir. Noktalama işaretleri kaldırılmasının ardından veri içerisindeki büyük harfler küçük harfe dönüştürülür. Çünkü büyük harfle başlayan Ağaç kelimesi ile küçük harfle başlayan ağaç kelimesi aynı anlama gelmektedir ve aynı şeyi temsil etmektedir. Sayısallaştırma yapılırken bu kelimelere aynı sayılar atanmalıdır. Bu işlem yapılmazsa sayısallaştırma işleminde bu kelimeler bilgisayar tarafından farklı olarak algılanacağı için farklı sayılar atanır. Bununla birlikte verilere tokenizasyon işlemi de uygulanmaktadır. Bu işlem yapılırken split fonksiyonundan yararlanılmıştır. İşlem sonunda verilerdeki kelimeler birbirinden ayrılır ve kelimelere ulaşım kolaylaşır. Ardından verilerdeki etkisiz kelimeler kaldırılmıştır. Etkisiz kelimeler cümleden çıkarıldığında cümlenin anlamında bir değişikliğe neden olmaz. İngilizce’de bu kelimelere “and”, “the”, “to”, “so” ve “for” gibi kelimeler örnek verilebilir. Böylelikle metin içerisinde daha önemli kelimeler kalır. Sonrasında tweetlerin içerisinde bulunan linklerin bir kelime olarak düşünülmemesi için tweet içerisinde linklerin kaldırılması işlemi yapılmıştır. Son olarak stemming işlemi yapılmıştır. Stemming kelimelerdeki eklerin kaldırılması işlemidir. Stemming yapılırken ekli bir kelimeye bulunabilen ortak ön ve son eklerin bir listesini dikkate alınarak kelimenin başlangıcı veya sonu kesilmeye çalışılır. Kelimelerin fazla kesilmesi sonucu aşırı sıkılama meydana gelebilir. Aşırı sıkılama sonucu farklı kelimeler çok fazla kesilme sonucu aynı steme sahip olabilirler. Ayrıca aşırı kesilen kelimenin bütün anlamı kaybolabilir. Örneğin verilerde bulunan “became” kelimesi stemming işlemi sonucunda “becam” haline dönüşerek anlamsızlaşmıştır. Proje içerisinde Snowball Stemmer kullanılmıştır.

3.1.6. Metin Vektörizasyonu

Önişlemlerden geçmiş verileri bilgisayarların anlayabileceği hale getirmek amacıyla verilere sayısallaştırma işlemi uygulanmıştır. Bu işlem için proje içerisinde kullanılan modeller Word2Vec, FastText, CountVectorizer ve HashingVectorizer şeklinde sıralanabilir. Bu modellerin kullanımıyla metinler vektör modelleri haline dönüştürülür.

3.1.6.1. Word2Vec

Twitter'da Gerçek-Sahte Haber Tespit Projesi kapsamında birçok model oluşturulmuştur. Bu modellerden biri oluşturulurken Word2Vec kelime temsil yönteminden faydalanılmıştır. Bu teknik 2013 yılında Google tarafından geliştirilmiştir. Word2Vec temel olarak birer adet

girdi, çıktı ve gizli katmandan oluşan bir yapay sinir ağıdır(Keskin, 2018). Word2Vec modelinde bir pencere boyutu vardır ve bu pencere boyutu içerisinde bulunan kelimeler arasında bir ilişki kurulur. Bunu sonucunda vektör uzayında bu kelimeler birbirine yakın olarak konumlandırılır. Bu yüzden Word2Vec kelimeler arasında bulunan anlam ilişkilerini çıkarma işlemi yapabilir. Word2Vec’in iki çeşit alt yöntemi bulunmaktadır. Bu yöntemler CBOW ve Skip-Gram’dır. Bu yöntemlerden biri olan CBOW modelinde pencere boyutunun merkezinde bulunmayan kelimeler girdi olarak alınırken merkezde bulunan kelime tahmin edilmeye çalışılır. Bunun tam tersi olarak Skip-Gram modelinde ise merkezinde bulunan kelimeler girdi olarak alınmaktadır. Bu modelde merkezde bulunmayan diğer kelimeler tahmin edilmeye çalışılır. Bu işlemler bütün cümlelere uygulanmaktadır. CBOW yöntemi genel olarak büyük veri kümelerinde, Skip-Gram ise daha küçük veri kümelerinde iyi çalışmaktadır.

3.1.6.2. FastText

Diğer bir model oluşturulurken FastText tekniğinden yararlanılmıştır. Bu teknik 2016 yılında Facebook tarafından geliştirilmiştir. Word2Vec’in bir uzantısıdır. Kelimeler tek tek işlenmek yerine n-gram yöntemiyle bölünerek işlenir. Bu yöntemde n harf sayısını temsil etmektedir. $n=2$ ise bigram adı verilmekte ve örneğin ağaç kelimesi ağ, ğa, aç şeklinde parçalanmaktadır. Kelimeler bölündüğü için eğer yanlış yazılan bir kelime varsa bu kelimenin doğru kelimeye benzemesi durumunda yakın bir konumlandırmaya sahip olur. FastText, Word2Vec’ten daha hızlı çalışmaktadır.

3.1.6.3. Count Vectorizer

Count Vectorizer kelimeleri One Hot Encoding ile bernzer olarak bir kelimenin karşılığı kelimenin cümle içerisinde geçme sıklığı ile temsil edilir. Matriste her sütun birbirinden farklı bir kelimeyi temsil etmektedir. Projede ise matristeki her satır bir tweete karşılık gelmektedir. Yani matrisin boyutu veride bulunan benzersiz kelime sayısı ile verideki tweet sayısı kadardır. Bu da verinin büyük olması durumunda çok fazla yer kaplayabileceği anlamına gelmektedir. Word2Vec ve FastText aksine bu yöntemde kelimeler arasındaki ilişkiler çıkarılamamaktadır.

3.1.6.4. Hashing Vectorizer

Hashing Vectorizer Count Vectorizer yöntemine benzerlik göstermektedir. Aralarındaki farklardan biri metnin içerisinde geçen kelimelerin cümle içinde geçme sayısına bir hash fonksiyonunun uygulanmasıdır. Diğerisi ise benzersiz kelimeleri bellekte tutmamasıdır.

3.1.7. Tokenizasyon

Veriler içerisinde bulunan kelimelerin sayılara dönüştürülmesi için keras kütüphanesinin önişleme modülü kullanılmıştır. Burada her bir kelime sayılarla kodlanır. Daha sonra her bir cümle içerisinde farklı sayıda kelime olduğundan her bir listenin uzunluğu farklı olur. Bunların modele verilebilmesi için listelerin eşit uzunlukta olması gerekmektedir. Bu yüzden padding işlemi uygulanır. Tweetler içerisinde en çok kelimeye sahip olan cümlelerin uzunluğu maksimum uzunluk olarak belirlenip tüm listelere bu uzunluğa kadar sıfır eklenir. Bu şekilde tüm listelerin boyutu eşit olur. Daha sonra padding uygulanmış veri Word2Vec ve FastText modelleri ile ağırlıklı matrise dönüştürülür. Daha sonra bu ağırlıklı matris derin öğrenme modelinde Embedding katmanında kullanılmıştır

3.1.8. Modeller İçerisinde Kullanılan Yapılar

3.1.8.1. LSTM

Uzun kısa süreli bellek ağları (LSTM) uzun süreli bağımlılıkları öğrenebilen özel bir RNN türüdür. Doğal dil işleme problemlerinde genellikle RNN yapısı geçmiş bilgileri de hatırladığı için iyi sonuçlar verir. Bunun nedeni örneğin biri kitap okurken okunulan kelimeleri anlayabiliyor olmasının sebebi o okuduğu kelimedenden önceki kelimeleri de hatırlayabiliyor olmasıdır. Yani RNN de geçmiş bilgiler hatırlanabildiği için doğal dil işleme problemlerinde iyi çalışır. RNN kısa vadeli bir hafızaya sahiptir. Bu yüzden bazı problemlerde işe yaramaz. Bu sorunu aşmak için LSTM geliştirilmiştir. LSTM bilgileri daha iyi depolar. Standart bir RNN tek bir tanh katmanı içerirken, LSTM'ler iletişim halinde olan 4 farklı katman içerirler. Bu katmanlar cell state, unutma kapısı (forget gate), giriş kapısı (input gate) ve çıkış kapısı (output gate) olarak sıralanabilir. Unutma kapısında hangi bilgilerin silineceği hangi bilgilerin depolanacağı kararını veren kapıdır. Giriş kapısında Cell State güncellemesi yapılır. Çıkış kapısında ise bir sonraki hücrenin girişinin belirlendiği kapıdır. Hücrenin çıkış sonucu belirlenir. Standart LSTM ağları dizileri sırayla işler, gelecekteki bağlamı görmezden gelirler(Akköse, 2020)(Olah, 2015).

3.1.8.2 Bidirectional LSTM

Bir RNN türü olan çift yönlü LSTM'ler iki ayrı RNN'i bir araya getirmektedirler. Aynı anda çift yönde ilerleme yaparlar. Çift yönlü işleme mantığı sayesinde hem önceki hem de sonraki zamana ait bilgilere sahip olabilmektedirler. Böylelikle bir kelimenin anlaşılabilmesi için hem önceki hem de sonraki kelime bilgisine sahip olunmaktadır. Çift yönlü LSTM'ler bağlamı iyi anlayabilmektedirler.

3.1.9. Arayüz

Grafiksel kullanıcı arayüzü kullanıcıların uygulama ile etkileşime girmesini sağlar. Kullanıcılar arayüz sayesinde belirtilen metin alanına bir tweet girerek bu tweetin gerçek ya da sahte olduğu bilgisini öğrenebilir. Arayüz tasarlanırken kullanıcı dostu olmasına önem verilmiştir. Bu nedenle arayüz karmaşıklıktan uzak, açıklayıcı ve kullanımı kolay olacak şekilde tasarlanmıştır. Arayüz içerisinde kullanıcıların metin girebileceği bir metin kutusu bulunmaktadır. Metin kutusunun alabileceği tweetin karakter sayısı bir tweette en fazla bulunacak karakter sayısı olacak şekilde sınırlandırılmamıştır. Fakat kullanıcının 280 karakterden fazla uzunlukta olan bir metin girdiği durumda bir Message Box ile hata mesajı döndürülmektedir. Bu hata mesajı içerisinde “Girdiğiniz tweetin boyutu 280 karakteri aşmamalıdır.” yazmaktadır. Ardından metin kutusu içerisindeki tüm metinler silinerek kullanıcıdan yeni bir girdi bekleme durumuna geçilir. Belirtilen metin kutusuna herhangi bir tweet girilmeden Tahmin butonuna tıklanması halinde kullanıcıya Message Box içerisinde bir bilgilendirme mesajı gösterilmektedir. En fazla 280 karakter uzunluğunda bir tweet girilmesinin ardından tahmin butonuna basıldığında ise girilen tweetin sahte veya gerçek olma bilgisi bir Message Box içerisinde kullanıcıya döndürülmektedir. Kullanıcı metin kutusu doluyken yeni bir tweet girmek isterse Sayfayı Temizle butonuna tıklayarak olaylıkla metin kutusu içerisindeki metni silebilmektedir. Böylelikle metin kutusu yeni kullanıma hazır hale gelmektedir.



Şekil 4: GUI tasarımı

Arayüzde kullanıcı tarafından girilen tweete tahmin işleminin yapılabilmesi için birtakım işlemler uygulanması gerekmektedir. Bu işlemler 3.1.5. Veri Önışlemleri başlığında anlatılan veriye uygulanan noktalama işaretlerinin kaldırılması, tüm harflerin küçük harfe dönüştürülmesi, cümle içerisindeki kelimelerin birbirlerinden ayrılması, etkisiz kelime kaldırılması, linklerin kaldırılması ve kelimelere köklerine dönüştürülmesi işlemleridir. Bu işlemlerin ardından 3.1.6 Tokenizasyon başlığında belirtilen işlemler yapılmaktadır.

3.2 Sonuçlar

Modelleri eğitmek için adam optimizery kullanılmıştır. En popüler gradyan iniş optimisasyon algoritmalarından biri olan adam optimizery Adagrad ve RMSprop algoritmalarının bir kombinasyonundan oluşur. Adam optimizerynin kullanılmasının nedeni araştırmalar sonucu nlp problemlerinde en iyi sonucu veren optimizery olmasıdır(Ruder, 2017).

Değerlendirme metriği olarak doğruluk seçilmiştir. Doğruluk ise doğru tahmin edilen verilerin toplam veriye oranıdır. Proje içerisinde eğitim ve validasyon doğruluğu olmak üzere iki farklı doğruluk ele alınmıştır. Eğitim doğruluğu veriler eğilirken yapılan sınıflandırma sonucu ortaya çıkan doğruluk değeridir. Validasyon doğruluğu ise modelin test verileri gibi daha önce görmediği verileri sınıflandırdığında ortaya çıkan doğruluk değeridir.

Name	Acc	Loss	Val_Acc
bidirect_lstm_w2v	0.974	0.075	0.857
dropout_w2v	0.974	0.072	0.855
count_vec	0.940	0.167	0.855
lstm_fast	0.979	0.060	0.851
dropout_fast	0.907	0.228	0.843
bidirect_lstm_fast	0.975	0.070	0.840
lstm_w2v	0.710	0.575	0.715
hash_vec	0.650	0.624	0.626

Şekil 5: Doğruluk Değerleri Tablosu

Veriler eğitim ve test verisi olarak eğitilmeden önce 80,20 olarak şekilde bölümlendirilmiştir. Eğitimler sonucu en yüksek eğitim doğruluk değerine Embedding katmanında FastText kullanılmış olan LSTM modeli sahiptir. En yüksek validasyon doğruluk değerine Embedding katmanında Word2Vec kullanılmış olan Bidirectional LSTM modeli sahiptir.

4. TARTIŞMA VE YORUM

Wang (2017) LIAR veri seti üzerinde yaptığı çalışmasında kullandığı Bidirectional LSTM modelinin 0.223 doğruluk değeriyle overfit olduğunu gözlemlemiştir. Bahad vd. (2019) çalışmalarında CNN, Vanilla RNN, Unidirectional LSTM ve Bidirectional LSTM kullanmışlardır. Bu çalışma sonucunda en iyi doğruluk değerini veren Bidirectional LSTM olmuştur. Aynı şekilde bu proje içerisinde de en yüksek validasyon doğruluk değerine sahip olan model Bidirectional LSTM modeli olmuştur. Çift yönde ilerleme yapması ve hem önceki hem de sonraki zamana ait kelimeleri sahip olabilmesi sonucu bağlamı daha iyi anlamış olabilmesi sebebiyle en iyi doğruluk değerine sahip olan modelin Bidirectional LSTM modeli olduğu düşünülmektedir.

5. SONUÇ

Sosyal medya üzerinde hızla yayılan bilgilerin doğruluğunun tespit edilememesi sonucu oluşan bilgi kirliliğinin önüne geçebilme amacıyla Twitter’da Gerçek-Sahte Haber Tespit projesi gerçekleştirilmiştir. Bu proje içerisinde doğal dil işleme teknikleri ve derin öğrenme teknikleri birlikte kullanılarak haber içerikli bir tweetin gerçek ya da sahte olup olmadığı en doğru şekilde tahmin edilmeye çalışılacaktır. Bunun için veriler üzerinde noktalama

işaretlerinin kaldırılması, tüm harflerin küçük harfe dönüştürülmesi, cümle içerisindeki kelimelerin birbirlerinden ayrılması, etkisiz kelime kaldırılması, linklerin kaldırılması ve kelimelere köklerine dönüştürülmesi gibi işlemler uygulanmıştır. Bu işlemlerin ardından Word2Vec ve FastText modelleri oluşturulmuştur. Bidirectional LSTM, LSTM ve Embedding katmanlarından oluşan farklı modeller oluşturulmuştur. Bu modellerin eğitilmesinin ardından en yüksek doğruluk değerinin Bidirectional LSTM modeliyle elde edildiği gözlemlenmiştir. Bu model kaydedilerek arayüz içerisinde kullanılmıştır.

KAYNAKÇA

1. Demir, Ş. (2021, Şubat 8). *Çağımızın çevrimiçi salgını: Yalan haber*. TRT Haber. <https://www.trthaber.com/haber/bilim-teknoloji/cagimizin-cevrimici-salgini-yalan-haber-554683.html>
2. Vijayaraghavan, S., & Guo, Z., & Wang, Y., & Voong, J., & Xu, W., & Nasser, A., & Cai, J., & Li, L., & Vuong, K., & Wadwa, E. (2020). *Fake News Detection with Different Models*. arXiv, 2003, 04978. <https://arxiv.org/pdf/2003.04978.pdf>
3. Lee, D., & Kim, Y., & Kim, H., & Park, S., & Yang, Y. (2019). *Fake News Detection Using Deep Learning*. JIPS, 15(5), 1119-1130. <https://www.koreascience.or.kr/article/JAKO201932569395106.pdf>
4. Bahad, P., Saxena, P., & Kamal, R. (2019). *Fake News Detection using Bi-directional LSTM-Recurrent Neural Network*. Procedia Computer Science. 165, 74–82. <https://sci-hub.se/https://doi.org/10.1016/j.procs.2020.01.072>
5. Jadhav, S. S., & Thepade, S. D. (2019). *Fake News Identification and Classification Using DSSM and Improved Recurrent Neural Network Classifier*. Applied Artificial Intelligence, 1–11.
6. Wang, W.Y. (2017). “*Liar, Liar Pants on Fire*”: A New Benchmark Dataset for Fake News Detection. arXiv, 1705, 00648. <https://arxiv.org/pdf/1705.00648.pdf>
7. Project Jupyter. (2021, Aralık 15). In Wikipedia. https://en.wikipedia.org/wiki/Project_Jupyter
8. Spyder (software). (2021, Aralık 16). In Wikipedia [https://en.wikipedia.org/wiki/Spyder_\(software\)](https://en.wikipedia.org/wiki/Spyder_(software))
9. Keskin, M. (2018, Ağustos 6). *Word2Vec, FastText, GloVe*. Medium. <https://medium.com/codable/word2vec-fasttext-glove-d4402fa8cce0>
10. Akköse, O. (2020, Aralık 22). *Uzun-Kısa Vadeli Bellek(LSTM)*. Medium. <https://medium.com/deep-learning-turkiye/uzun-k%C4%B1sa-vadeli-bellek-lstm-b018c07174a3>
11. Olah, C. (2015, Ağustos 27). *Understanding LSTM Networks*. GitHub. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
12. Ruder, S. (2017, Temmuz 25). *Deep Learning for NLP Best Practices*. Ruder. <https://ruder.io/deep-learning-nlp-best-practices/index.html#optimization>