

T. C.
EGE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

STAJ RAPORU

Öğretim Yılı :2020/2021

Stajerin

Adı Soyadı : Erdal Kutay Şeşen

Öğrenci No'su: 05160000675

Staj Raporunu Onaylayan Kişinin

Adı ve Soyadı : Dr. Öğr. Üyesi Özgür Gümüş

İmzası :

.....

İçindekiler

A.	Staj yapılan kuruluşun tanıtımı, organizasyonu, faaliyet tarihçesi Kurumun Tarihçesi	2
B.	Kuruluştaki bilgisayar altyapısının tanıtımı	2
C.	Kuruluşun organizasyon şeması	5
D.	Hafta bazında, staj süresi içinde yapılan işlerin özeti	5
E.	Staj süresi içinde yapılan çalışmalar	6
F.	Staj değerlendirmesi	23
G.	Kaynak Kod ve Dosyalar	23
H.	Referanslar	24

A. Staj yapılan kuruluşun tanıtımı, organizasyonu, faaliyet tarihçesi Kurumun Tarihçesi

Başlangıçta sadece Ziraat ve Tıp Fakültelerini içeren Ege Üniversitesi (E.Ü.), 1955 yılında kurulmuştur. Daha sonraları, E. Ü.'nin kapsamında bulunan fakültelere yenileri de eklenmiştir. Bu kuruluşlarda yapılan bilimsel araştırmaların matematiksel ve istatistiksel değerlendirmeleri için, bilgi işlem aygıtları olarak mekanik ve elektrikli hesap makineleri kullanılmıştır. 1970 yılında Ege Üniversitesi'nin bilgi işlem gereksinimini karşılamak ve aynı zamanda üniversiteye başvuran öğrencilerin fakültelere yerleştirme işlemini yapmak amacıyla, E.Ü. Rektörlüğü'ne bağlı olarak, Elektronik Hesap Merkezi kurulmuştur. Bu dönemde, Ege Üniversitesi'nin mühendislik eğitimi veren birimlerinin öğretim programları kapsamına, Bilgisayar Programlama adı altında bir ders eklenmiş ve bu dersle birlikte, bazı fakültelerin istatistik derslerinin Elektronik Hesap Merkezi'nce yürütülmesi düşüncesi benimsenmiştir. Merkez, Elektronik Hesap Merkezi statüsü ile akademik bir işlev yüklenemeyeceği için, 1975 yılında yeniden organize olmuş ve adı Elektronik Hesap Bilimleri Enstitüsü'ne dönüştürülmüştür. Böylelikle kuruluş akademik ve idari kadrolara kavuşmuştur.

B. Kuruluştaki bilgisayar altyapısının tanıtımı

Kullanılan Bilgisayar

Lenovo IdeaPad L340 Intel Core i7 9750H 8GB 256GB SSD GTX1650 15.6" FHD
Windows 10 Home

Kullanılan Kütüphaneler ve Teknolojiler

- NumPy v1.19.0 [\[1\]](#)
- Pandas version: 1.1.1 [\[2\]](#)
- Sklearn 0.23 [\[3\]](#)
- Tweepy 3.9.0 [\[4\]](#)
- GPT 2 Natural Language Processing Model [\[5\]](#)
- Jupyter Notebook 1.0 [\[6\]](#)
- Python version 3.7 [\[7\]](#)
- Anaconda [\[8\]](#)
- Tensorflow 1.15.0 [\[9\]](#)
- Beautiful Soup 4.9.0 [\[10\]](#)
- Colab [\[11\]](#)
- Translate [\[12\]](#)
- Pickle [\[13\]](#)
- Gfycat [\[14\]](#)
- WordCloud [\[15\]](#)

1. Numpy

Python programlama dilinin bir kütüphanesi olup, büyük, çok boyutlu diziler ve matrisler için destek eklerken, bu dizilerde çalışmak için yüksek düzeyli matematiksel fonksiyonların geniş bir koleksiyonudur. [\[16\]](#)

2. Pandas

Pandas, Open Source lisanslı bir Python kütüphanesidir. Python programlama dili için yüksek performanslı veri yapıları ve veri analizi araçları sunmaktadır. [\[17\]](#)

3. Sklearn

Scikit-learn, Python programlama dili için ücretsiz bir yazılım makinesi öğrenme kütüphanesidir. [\[18\]](#)

4. Tweepy

Tweepy, Twitter'ın Python için yazılan bir API kütüphanesidir. Tweet atmak, çekmek veya kullanıcı bilgilerine erişmek gibi Tweet API özellikleri için kullanılır.

5. GPT 2

GPT-2 Open A.I. tarafından geliştirilen bir State of Art Natural Language Processing kütüphanesidir.

6. Jupyter Notebook

Jupyter not defterleri, kod ve veriler için web tabanlı etkileşimli bir geliştirme ortamıdır.

7. Python

Python, yorumlanmış, etkileşimli, nesne yönelimli bir programlama dilidir. Linux ve Mac OS dahil birçok Unix varyantında ve Windows'ta çalışır. [\[19\]](#)

8. **Anaconda**

Anaconda, Python ve R programlama dillerinin bilimsel hesaplama kullanımında paket yönetimini kolaylaştırmayı amaçlayan Open Source dağıtımdır. Conda paket yönetim sistemi ile yönetilir.

9. **Tensorflow**

TensorFlow, makine öğrenimi için uçtan uca açık kaynaklı bir platformdur. Araştırmacıların son teknoloji ürünü makine öğrenimi geliştirmesine ve geliştiricilerin makine öğrenimi destekli uygulamaları kolayca oluşturup dağıtmasına olanak tanıyan kapsamlı ve esnek araçlar, kitaplıklar ve topluluk kaynakları olan bir ekosistemine sahiptir.

10. **Beautiful Soup**

Beautiful Soup, HTML ve XML dosyalarından veri çekmek için yazılmış bir Python kütüphanesidir. Ayrıştırma ağacında gezinmek, aramak ve değiştirmek için kullanılır.

11. **Colab**

Colab, makine öğrenimi eğitimi ve araştırmasının yaygınlaştırılmasına yardımcı olmak için oluşturulmuş bir Google araştırma projesidir. Tamamen bulutta çalışan bir Jupyter dizüstü bilgisayar ortamıdır.

12. **Translate**

Translate, birden çok çeviri sağlayıcısını destekleyen python ile yazılmış basit ama güçlü bir çeviri aracıdır.

13. **Pickle**

Pickle, sıralama veya düzleştirme olarak da adlandırılan Python nesne yapılarını serileştirmek ve serileştirmek için kullanılır.

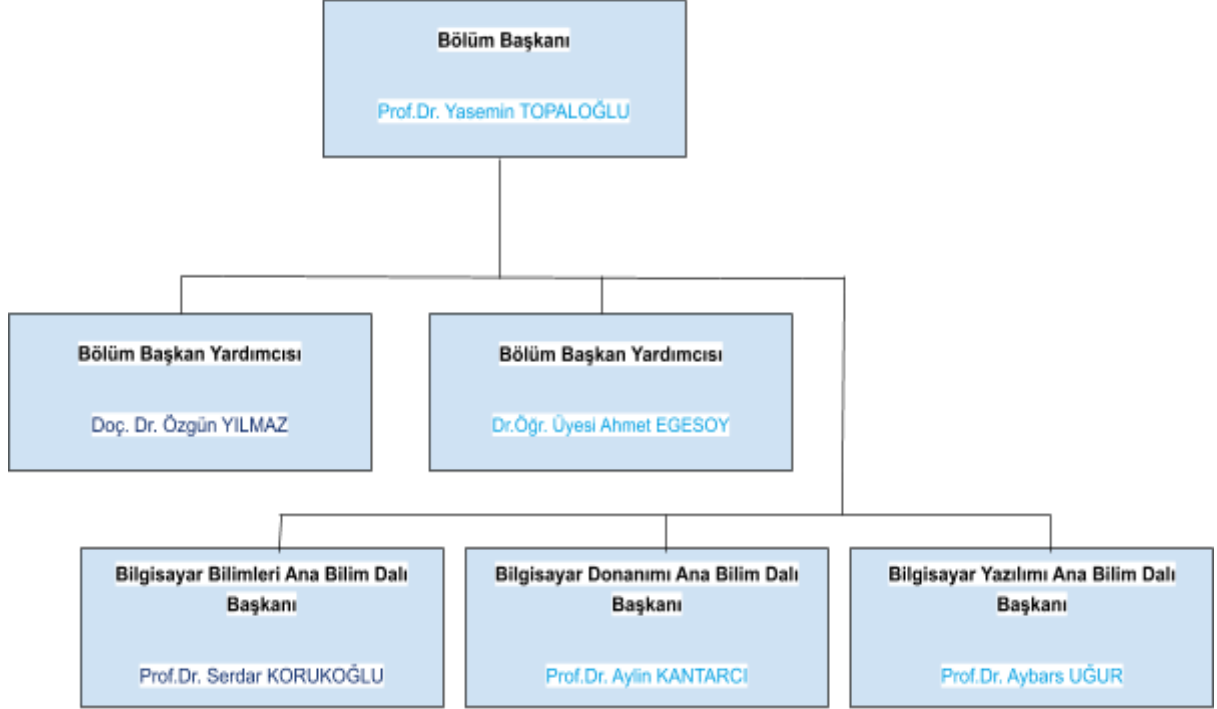
14. **Gfycat**

Gfycat, Richard Rabbat, Dan McEleney ve Jeff Harris tarafından kurulan, kullanıcı tarafından oluşturulan kısa bir video barındırma şirkettir. Geliştiriciler için ücretsiz API sunmaktadır.

15. WordCloud

WordCloud, kelimelerin boyutunun sıklıklarını temsil ettiği bir metindeki sık kelimeleri görselleştirmek için kullanılan bir tekniktir.

C. Kuruluşun organizasyon şeması



D. Hafta bazında, staj süresi içinde yapılan işlerin özeti

Birinci Hafta (20 - 27 Ağustos 2020)

Beautiful Soup kullanılarak veriler toplandı ve GPT 2 Finetune işlemi gerçekleştirildi. Tweet API ile GPT 2 üretilen Tweetler yollanabilir hale getirildi.

İkinci Hafta (1 - 5 Eylül 2020)

Twitter'dan çekilen kullanıcı bilgileri ile Sınıflandırma yapıldı ve geri takip edeceği tahmin edilen Twitter hesapları takip edildi.

Üçüncü Hafta (6 - 12 Eylül 2020)

Sentiment Analiz yapıldı. Bir tweetin pozitif veya negatif olmasına bakıldı. Eğer Tweet pozitif ise Twitter API kullanılarak beğenildi.

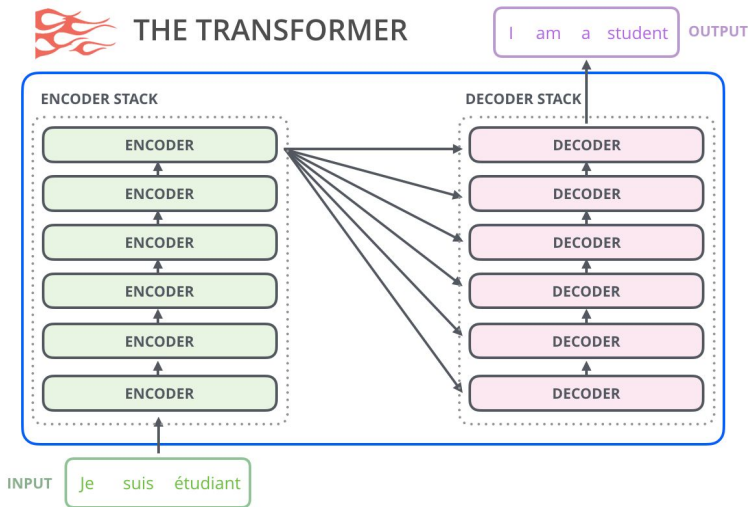
Dördüncü Hafta (13 - 18 Eylül 2020)

Twitter API üzerinde API kullanım koşullarını ihlal etmeden Tweet atan, takipleşen, Tweet beğenen, Tweet Retweetleyen bot yazıldı. Daha önceki haftalarda yapılan işlemler otomatize edildi.

E. Staj süresi içinde yapılan çalışmalar

Twitter üzerinde daha önceden Finetune edilerek elde edilen yeni GPT 2 modeli ile Tweet atarak, diğer Twitter accountları ile etkileşime giren bir yapay zeka Twitter hesabı yapmaktır. Aynı zamanda, Sklearn ile Twitter üzerinden toplanılmış olan kullanıcı verilerini Sınıflandırma yöntemleri kullanarak bir hesabın takibe dönüp dönmeyeceği tahmin edilerek gerçek zamanlı takip yapabilecektir.

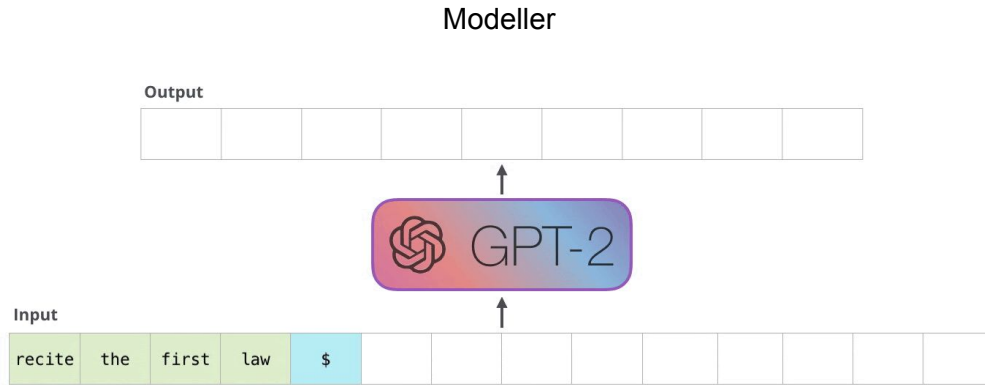
GPT 2 Language Processing Modelinin Çalışma Mantığı



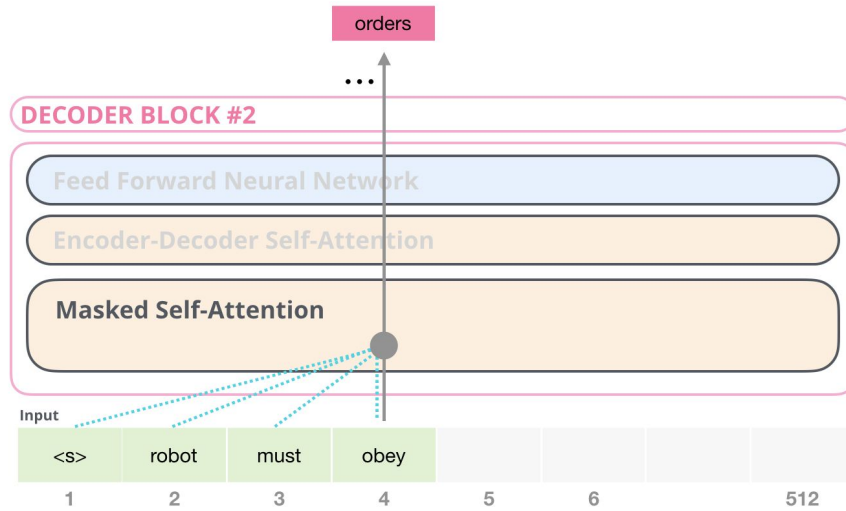
GPT-2 Language Modelinin çalışma mantığı bir sonraki kelimeyi bulmaya dayanmaktadır, böylelikle yapay zeka modeli anlamlı cümleler kurulabilmektedir. Arkasındaki teknoloji Google tarafından yayınlanan "Attention Is All You Need" [20] makalesine dayanır. Burada bahsi geçen yapıya Transformer ismi verilmektedir. Basitçe bahsetmek gerekirse bir transformer Encoder ve Decoder olmak üzere 2 kısma ayrılır (Burada bahsi geçen Open A.I. tarafından kullanılan yalnızca Decoder kullanılmaktadır.). Kendi içlerinde de katmanlara ayrılırlar.

Encoder aldığı girdiyi Encoder Stack üzerinde işleyerek Decoder için anlamlı bir veri üretir. Decoder ise bu veriden anlamlı bir çıktı üretir. Bun yaparken daha önceden eğitilmiş Model kullanılır. GPT-2'nin gücü ise tam olarak burada ortaya çıkar. Yaklaşık 40 Gb ham veri, Reddit ve Wikipedia üzerinden toplandığı belirtilmektedir. GPT-2 modeli Transformer'ın Decoder blokları üzerine inşa edilmiştir.

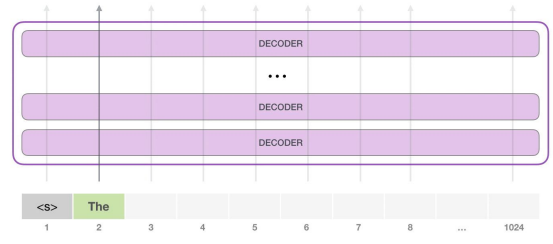
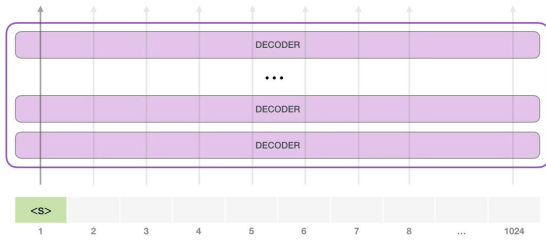
Şimdiye kadar bir çok farklı büyüklükte model Open A.I. tarafından Open Source olarak sunulmuştur bunun sebebi modelin büyüklüğü arttıkça kullanabilmek için gerekli kaynak miktarı da artmaktadır. Ayrıca tam modeli kötüye kullanım ihtimali olabileceği düşünülerek uzun bir süre yayınlanmamıştı.



GPT-2 modeli her bir token üretildikten sonra bu token da girdiye eklenerek yeni bir token yaratmaktadır. Bu yöneme auto-regression ismi verilmektedir. Bahsettiğimiz gibi GPT-2 modeli yalnızca Transformer'ın decoder kısmı üzerine inşa edilmiştir bu modeller Transformer-Decoder olarak adlandırılmaktadır. Burada GPT-2 Decoder'ında klasik Decoder'ların aksine Self-Attention katmanı yerine Masked Self-Attention katmanı bulunmaktadır. Self-Attention katmanları inputtaki tüm tokenları dikkate alırken Masked Self-Attention bunun önüne geçer. Kısaca özetlemek gerekirse Encoder yükü sistemden atılmıştır.

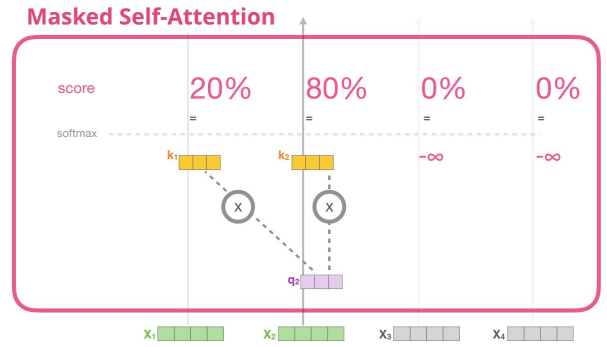


Modelin çalışma mantığını daha iyi anlamak için şu örneğe bakabiliriz. Modelin çalışmaya başlaması için aslında bir token göndermemiz gerekmektedir fakat bunun anlamlı bir kelime olmasına gerek yok zira vereceğimiz <s> gibi bir girdi ile de GPT-2 kendiliğinden bir cümle kurabilir. Eğer anlamlı bir girdi verirsek devamını kendiliğinden getirecektir.



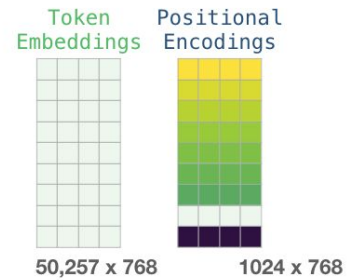
Biraz daha detaya indersek model gelen her bir input için Embedding Matrix'e bakar, burada modelde bulunan her bir kelimeye karşılık gelen bir anlam vardır. GPT-2 Vocabulary modeli burada buna karşılık gelmektedir. Her modelin embedding uzunluğu farklıdır. Yine modelde gömülü bulunan Positional Encoding de token embedding'e eklenerek elde edilen vektör işlenmek üzere Decoder'a gönderilir.

Self-Attention katmanı ise girdiler arasındaki konteksi belirliyor. Örnek vermek gerekirse daha önce verdiğimiz "A robot must obey the orders given it" cümlesindeki "it" girdisi için Self-Attention bunun "robot" ile ilişkili olduğunu belirliyor. Masked Self-Attention'da ise her bir adımda sadece o adıma kadar olan girdiler varmış gibi davranarak diğer girdileri perdeler. Daha sonraki adımda Feed Naturel Network katmanı devreye giriyor.



Feed Naturel Network 2 katmanlı bir yapıya sahiptir. Self-Attention bilgisi de eklendikten sonra oluşturulan vektörü alıp her Decoder grubunun kendisinde bulunan ağırlık matrisi ile işleyerek bir çıktı üretir. Her Decoder'ın kendi ağırlık matrisi olmasına rağmen modelde yalnızca birer Token Embedding ve Positional Encoding matrisi bulunmaktadır.

 GPT-2 SMALL



Veri Toplama (Beautiful Soup)

GPT modelini Finetune edebilmek için öncelikle internet üzerinden “Steins;Gate” adlı anime ile ilgili wiki sitesi [21] ve tüm bölümlerin yazılı dökümünün olduğu parchive [22] web sitesi seçildi. Daha sonra Beautiful Soup yardımı ile gerekli bağlantılar ayıklanıp içerikleri text dosyasına kaydedildi.

```
In [1]: import requests
from bs4 import BeautifulSoup
r = requests.get('https://1parchive.org/SteinsGate/')
source = BeautifulSoup(r.content,"lxml")
source.title

Out[1]: <title>Steins;Gate</title>

In [2]: links = []
source = source.find_all(class_="toc")
for x in source:
    for a in x.find_all('a', href=True):
        print( "Found the URL: " + "https://1parchive.org/SteinsGate/" + a['href'])
        links.append("https://1parchive.org/SteinsGate/" + a['href'])

Found the URL: https://1parchive.org/SteinsGate/Update%2001/
Found the URL: https://1parchive.org/SteinsGate/Update%2002/
Found the URL: https://1parchive.org/SteinsGate/Update%2003/
Found the URL: https://1parchive.org/SteinsGate/Update%2004/
Found the URL: https://1parchive.org/SteinsGate/Update%2005/
Found the URL: https://1parchive.org/SteinsGate/Update%2006/
Found the URL: https://1parchive.org/SteinsGate/Update%2007/
Found the URL: https://1parchive.org/SteinsGate/Update%2008/
Found the URL: https://1parchive.org/SteinsGate/Update%2009/
Found the URL: https://1parchive.org/SteinsGate/Update%2010/
Found the URL: https://1parchive.org/SteinsGate/Update%2011/
Found the URL: https://1parchive.org/SteinsGate/Update%2012/
Found the URL: https://1parchive.org/SteinsGate/Update%2013/
Found the URL: https://1parchive.org/SteinsGate/Update%2014/
Found the URL: https://1parchive.org/SteinsGate/Update%2015/
Found the URL: https://1parchive.org/SteinsGate/Update%2016/
Found the URL: https://1parchive.org/SteinsGate/Update%2017/
Found the URL: https://1parchive.org/SteinsGate/Update%2018/
Found the URL: https://1parchive.org/SteinsGate/Update%2019/
```

BeautifulSoup for parchive.org Jupyter Notebook

Yukarıdaki ekran görüntüsünde görülebileceği üzere sitedeki tüm sayfaların bağlantılar çekildi.

```
In [3]: content = []
for link in links:
    r = requests.get(link)
    source = BeautifulSoup(r.content,"html.parser")
    print(source.title)
    source = source.find(class_="cblock")
    content.append(source.text.replace("\n", ""))
    print(source.text)
```

Daha sonra bağlantıların içeriğinde HTML Classı “cblock” olan blokları tek tek çekilerek “output.txt” isimli bir dosyaya kaydedildi.

```
: #Save them to file...
with open('scripts.txt', 'w', encoding='utf-8') as f:
    for x in content:
        f.write(x)
    f.close

: with open('scripts.txt',encoding='utf-8') as infile, open('output.txt', 'a',encoding='utf-8') as outfile:
    for line in infile:
        if not line.strip(): continue
        outfile.write(line) # non-empty line. Write it to output
```

Aynı işlemler Wiki sitesi içinde uygulandı ve tüm çıktılar “output.txt” dosyasına kaydedildi.

Colab Üzerinde Retrain (Finetune)

GPT 2 modelini Finetune edebilmek için gerekli GPU gücünü Google'nın Colab servisi ile sağlıyoruz. Çıktı dosyasını Colab'a yükledikten sonra sırası ile şu işlemleri yapıldı.

Downloading GPT-2

If you're retraining a model on new text, you need to download the GPT-2 model first.

```
In [ ]: gpt2.download_gpt2(model_name="355M")

Fetching checkpoint: 1.05Mit [00:00, 161Mit/s]
Fetching encoder.json: 1.05Mit [00:00, 111Mit/s]
Fetching hparams.json: 1.05Mit [00:00, 559Mit/s]
Fetching model.chkpt.data-00000-of-00001: 1.42GiB [00:06, 227Mit/s]
Fetching model.chkpt.index: 1.05Mit [00:00, 244Mit/s]
Fetching model.chkpt.meta: 1.05Mit [00:00, 97.8Mit/s]
Fetching vocab.bpe: 1.05Mit [00:00, 163Mit/s]
```

Mounting Google Drive

```
In [ ]: gpt2.mount_gdrive()

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc41.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aob&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fhttps%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly&response_type=code

Enter your authorization code:
.....
Mounted at /content/drive
```

GPT 2 355M orta büyüklükteki modelini Colab makinemize indiriyoruz, Finetune işlemi bu model üzerinden gerçekleşecek. Ayrıca en son üretilen modeli Google Drive'a yüklemek için Google Drive hesabı da bağlandı.

Finetune GPT-2

The next cell will start the actual finetuning of GPT-2. It creates a persistent TensorFlow session which stores the training config, then runs the training for the specified number of steps.

```
In [ ]: sess = gpt2.start_tf_sess()

gpt2.finetune(sess,
               dataset=file_name,
               model_name='355M',
               steps=1000,
               restore_from='fresh',
               run_name='run1',
               print_every=10,
               sample_every=200,
               save_every=500
               )
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/gpt_2_simple/src/sample.py:17: where (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/gpt_2_simple/src/memory_saving_gradients.py:62: get_backward_walk_ops (from tensorflow.contrib.graph_editor.select) is deprecated and will be removed after 2019-06-06.
Instructions for updating:
Please use tensorflow.python.ops.op_selector.get_backward_walk_ops.
Loading checkpoint models/355M/model.ckpt
INFO:tensorflow:Restoring parameters from models/355M/model.ckpt

0% | 0/1 [00:00<, ?it/s]

Loading dataset...

100% |██████████| 1/1 [00:04<00:00, 4.32s/it]

dataset has 734849 tokens
Training...
[10 | 26.15] loss=3.14 avg=3.14
```

1000 iterasyonun ardından modelimiz hazır. [1000 | 1734.37] loss=2.39 avg=2.43

TextGenerator Notebook

Bu notebook üzerinde GPT2 ile text üretmek için kullanılacak olan generate fonksiyonu bulunmaktadır. Bu fonksiyon random bir şekilde Tweet üretebilir, bunun için daha önce üretilen yeni modeli kullanılmaktadır. Ayrıca Prefix vererek verilen söze de devam edebilir.

```
In [9]: def generate(prefix_ = None, prefix_return=False, length = None, model_name = "scripts_wiki"):
    text = gpt2.generate(sess,
        model_name = model_name,
        length = length if length != None else random.randint(30,60),
        prefix= prefix_,
        include_prefix= prefix_return,
        return_as_list=True)[0]
    while(not "." in text):
        text = gpt2.generate(sess,
            model_name = model_name,
            length = length if length != None else random.randint(30,60),
            prefix= prefix_,
            include_prefix= prefix_return,
            return_as_list=True)[0]
    lines = text.split('\n')
    tweet=""
    for x in lines:
        if x != '':
            tweet= tweet + x
    index = -1
    punctuation = [".", "?", "!", ".com", ".net", ".org"]
    punct = ""
    for p in punctuation:
        if tweet.rfind(p) > index:
            index = tweet.rfind(p)
            punct = p
    if punct in tweet:
        tweet = tweet[:index] + punct
    if prefix_ != None:
        return tweet if prefix_return else tweet[len(prefix_):].replace(' ','')
    else:
        return tweet.replace(' ','')
```

Ayrıca bu notebook üzerinde bir de test fonksiyonu bulunmaktadır bu da modeli test etmek için kullanılıyor.

Twitter Entegrasyonu

Tweepy kütüphanesini kullanabilmek için ilk önce yeni bir Twitter APP'i açıyoruz. Daha sonra Twitter hesabımızın bu APP'e izin vermesi gerekir. Bunun için Twitter Finding Keys Notebook üzerinde Authorization linki üreterek hesabı APP'e bağladık. Daha sonra API için gerekli anahtarları çekiyoruz.

```
In [9]: #Twitter App
CONSUMER_KEY = "u0eF31p60Q6Nkt31m9sjl0tq0"
CONSUMER_SECRET = "f5ESVjA5sBamjGvXzA6wKX8kuUL392rXPtvNEmUZLIjdXcNa8o"
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
# the following command will print a URL to visit:
print(auth.get_authorization_url())

https://api.twitter.com/oauth/authorize?oauth_token=RlbWjGAAAAAAv-TYAAABdBN7r0s
```

```
In [10]: #Access pin for account is 0557712
token = auth.get_access_token(verifier = "0557712")
mycreds = {
    'consumer_key': CONSUMER_KEY,
    'consumer_secret': CONSUMER_SECRET,
    'access_token': token[0],
    'access_token_secret': token[1]
}
```

Twitter Notebook üzerinde Tweepy API ile Tweet atmak, takipçi ve takip listesini getirmek, ana sayfadaki ve profildeki son 20 Tweet çekmek ve profil bilgisini değiştirmek için fonksiyonlar bulunmaktadır. TextGenerator notebook burada import edilerek Tweet oluşturulur ve gönderilir.

Functions

There're functions of Twitter class.

- followers -> Return Dataframe of Followers of Accounts
- following -> Return Dataframe of Account's Following accounts
- sendtweet -> Send tweet
- updateBio -> Update Bio field of Account
- timeline -> return timeline tweets of Account
- gettweets -> return tweets of one Account

Main Notebook

Main Notebook tüm programın çalıştığı notebooktur, Twitter Notebook burada kullanılır. Hangi hesap ile bağlandığımızı buradan görebiliriz.

Twitter Account

This is the Twitter Account we are using right now.

```
In [2]: tw.Account
Out[2]: {'Name': 'Kurusu Makise',
'Screen Name': 'KurusuM64305605',
'Following': 2,
'Followers': 1,
'Bio': 'I am Kurisu Makise. I am the mad scientist of the year 2000.' "You're a mad scientist, huh. What are you saying?" "I a
m Hououin Kyouma.'}
```

```
In [3]: tw.following()
Out[3]:
```

	Name	Screen Name	Following	Followers	Bio
0	CERN	CERN	435	2576646	#CERN, the European Organization for Nuclear Research, is the world's largest particle physics lab, home of the #LHC. In French: @CERN_FR
1	Crunchyroll	Crunchyroll	443	1417845	Welcome to the official account for Crunchyroll Bringing you the latest & greatest anime at the speed of Japan ~!

```
In [4]: tw.followers()
Out[4]:
```

	Name	Screen Name	Following	Followers	Bio
0	Kutay Şeşen	Kutayeen	476	1128	Laugh hard, Run fast, Be Kind There is not such thing as coincidence in this world. the only thing is hitsuzen. God is in his heaven All's right with the world

Burada test ve Tweet gönderme işlemleri gerçekleştirilir.

Test for GTP 2

```
In [7]: tw.test(2)
----- Sample: 1-----
Two years ago, a group of thieves broke into the lab and took my phone. They broke into it to steal the IBN 5100.
----- Sample: 2-----
More than half of the Dr. Jekyll and the Ms.

In [8]: tw.test(Text = "Time machine is ", Prefix_turn = True)
----- Sample: 1-----
Time machine is !" "You say it's not here, but if you go to the other side of the wall, you can come out here..." "I see! And t
hen you can go back to the other side. But it's not that simple.
```

Sending a Tweet

```
In [10]: tw.sendtweet()
Tweet: MoekaThis is an OVA, not a movie. We have no connection with the movie.Okabe, this isn't the first time you've done... ht
tps://t.co/4v425fAigC
```


Twitter Accounts

Twitter Accounts Notebook’unda Twitter API ile Twitter hesaplarının verileri çekildi. Bu veriler takipçi sayısı, takip ettikleri hesap sayısı, attıkları Tweet sayıları, takipçi ve takip listeleridir. 640 farklı Twitter hesabının bu bilgileri çekilmiştir. Bu veriler daha sonra “Accounts.csv” dosyasına kaydedildi.

Twitter Follower Sınıflandırma

Bu notebook dosyasında daha önce kaydettiğimiz “Accounts.csv” dosyası kullanılarak bu Twitter hesaplarını verileri üzerinde Sınıflandırma işlemi yapılmıştır.

	Name	Following	Followers	Description	Tweet Count	Following List	Followers List
0	nabi_kun	468	431	NaN	12203	['GalaksimsiBiris', 'kasvetlisenfoni', '14Rubi14', 'aseutel', 'nutzezforliezez', 'sakuras_soul', ...]	['Kutayeen', 'beyzanguney', 'theKarpuz', 'sorasayanora', 'abxushi', 'Usagidoodles', 'cevimenimb...
1	cmatrixs	180	213	NaN	232	['muyagci', 'realsnowgoons', 'anililungor', 'nurgaviria', 'avczxpo', 'yasminclueless', 'Kutayeen...]	['clkemrh', 'Samil_cicekk', 'avczxpo', 'nurgaviria', 'Yagizbirkan23', 'realsnowgoons', 'anililun...
2	fecitekme	77	60	forlorn and impetuous; lovely	158	['StendhalNara', 'sonrasizlik', 'marbmallow', 'Kutayeen', 'CrazyinRussia', 'kaotikfiona', 'c0eur...	['sonrasizlik', 'Kutayeen', 'allahhummm', 'kaotikfiona', 'c0eurcitr0n', 'adeushh', 'konuomusimdi...
3	couplangee	197	236	NaN	229	['lostinart_', 'HamzaMacabago', 'Kutayeen', 'morosanpper', 'shi_ningyou', 'nulldotcom', 'bes_kil...	['Nazar_yvz', 'Kutayeen', 'morosanpper', 'bes_kil', 'calapoks', 'HaloTheAnswer', 'nykslyssa', '...
4	inkedowl	232	202		4822	['diablosai', 'TsukinoHikarii', 'ArtistOlan', 'dndkyckk', 'dmdgr_', 'kyokosei', 'Kutayeen', 'am...]	['diablosai', 'TsukinoHikarii', 'Kutayeen', 'ArtistOlan', 'dndkyckk', 'dmdgr_', 'gal4cticunt', ...]

Bu hesapların takip ettikleri hesapların takipçi listesinde olup olmasına göre “Return Following” kolonu oluşturuldu. Her bir satırda Twitter hesabının ve takipçilerinin bilgileri bulunmaktadır. Burada daha önceden Accounts arasında olmayan hesaplar da olduğu için her yeni hesabın bilgileri Twitter API ile çekilmiştir. Bu işlem yaklaşık 3 gün gibi bir süre almıştır bunu nedeni Twitter API’nin günlük ve saatlik istek sınırındır.

```
In [102]: Data.head()
```

```
Out[102]:
```

	Name	Following	Followers	Tweet Count	Follower Name	Follower Following	Follower Followers	Follower Tweet Count	Return Following
0	nabi_kun	468	431	12203	GalaksimsiBiris	195	178	627	1
1	nabi_kun	468	431	12203	kasvetlisenfoni	324	302	847	1
2	nabi_kun	468	431	12203	14Rubi14	213	69	161	1
3	nabi_kun	468	431	12203	aseutel	219	135	25	1
4	nabi_kun	468	431	12203	nutzezforliezez	580	197	472	1

```
In [105]: Data["Return Following"] = Data["Return Following"].astype(object)
Data["Following"] = Data["Following"].astype(object)
Data["Followers"] = Data["Followers"].astype(object)
Data["Tweet Count"] = Data["Tweet Count"].astype(object)
Data["Follower Following"] = Data["Follower Following"].astype(object)
Data["Follower Followers"] = Data["Follower Followers"].astype(object)
Data["Follower Tweet Count"] = Data["Follower Tweet Count"].astype(object)
Data.describe(include= ['O'])
```

```
Out[105]:
```

	Name	Following	Followers	Tweet Count	Follower Name	Follower Following	Follower Followers	Follower Tweet Count	Return Following
count	311780	311780	311780	311780	311780	311780	311780	311780	311780
unique	532	427	390	391	210839	20640	60508	47420	2
top	rezil2015	1744	373	0	Kutayeen	0	3	0	0
freq	1990	3476	3457	10171	523	9643	861	4133	184938

Sonuç olarak toplanan verilerin bulunduğu Dataframe'de 311780 farklı girdi toplandı. Daha sonra Sınıflandırma işlemi için test ve train olmak üzere ikiye bölündü burada yalnızca 20% test için ayrıldı.

İlk olarak Linear Support Vector kullanılarak eğitim yapıldı. Sonuç olarak 0.75165 doğruluk oranı elde edildi.

```
In [126]: print("Accuracy:", accuracy_score(y_test, y_pred))
          print(confusion_matrix(y_test, y_pred))
          print(classification_report(y_test, y_pred))

Accuracy: 0.7516518057604721
[[26215 10802]
 [ 4684 20655]]
           precision    recall  f1-score   support

    0           0.85       0.71       0.77       37017
    1           0.66       0.82       0.73       25339

 accuracy
macro avg       0.75       0.76       0.75       62356
weighted avg     0.77       0.75       0.75       62356
```

2. olarak Decision Tree kullanılarak yapılan eğitim daha iyi sonuç verdi ve 0.832766 doğruluk elde edildi.

```
In [128]: print("Accuracy:", accuracy_score(y_test, y_pred))
          print(confusion_matrix(y_test, y_pred))
          print(classification_report(y_test, y_pred))

Accuracy: 0.8327666944640452
[[31743  5274]
 [ 5154 20185]]
           precision    recall  f1-score   support

    0           0.86       0.86       0.86       37017
    1           0.79       0.80       0.79       25339

 accuracy
macro avg       0.83       0.83       0.83       62356
weighted avg     0.83       0.83       0.83       62356
```

En son KNeighbors kullanılarak tekrar eğitim yapıldı ve onun sonucu Decision Tree'den biraz daha az olarak 0.827410 doğruluk oranı elde edildi.

```
In [132]: print("Accuracy:", accuracy_score(y_test, y_pred))
          print(confusion_matrix(y_test, y_pred))
          print(classification_report(y_test, y_pred))

Accuracy: 0.8274103534543589
[[31771  5246]
 [ 5516 19823]]
           precision    recall  f1-score   support

    0           0.85       0.86       0.86       37017
    1           0.79       0.78       0.79       25339

 accuracy
macro avg       0.82       0.82       0.83       62356
weighted avg     0.83       0.83       0.83       62356
```

En son olarak Decision Tree Classifier Pickle kütüphanesi kullanılarak "Classifier.pickle" dosyası olarak kaydedildi.

PredictFollow

Burada daha önceden kaydedilen “Classifier.pickle” ile takip edilen bir Twitter hesabının geri takibe dönüp dönmeyeceğini tahmin etme işlemini gerçekleştirecek fonksiyon bulunmaktadır.

```
In [ ]: def predict(Account):  
        y_pred = classifier.predict([[info['Following'],info['Followers'],info['Tweet Count']],Account['Following'],Account['Followers']])  
        return y_pred[0]
```

Main Notebook Üzerindeki İşlemler

Twitter üzerinde kullanıcı araması yapılarak takip edilebilecek yeni hesaplar bulundu. Bunun için Twitter Notebook’una yeni bir fonksiyon “search_user” eklenmiştir. Daha sonra bu fonksiyon kullanılarak potansiyel takip edilebilecek hesaplar seçilmiştir. “Steins Gate” anahtar kelimesi ile yapılan arama sonucu gelen hesaplar aşağıdaki Dataframe’de görülebilir.

	Name	Screen Name	Following	Followers	Tweet Count	Bio
995	シャロン	gate2010_steins	0	0	0	
996	José Julio	Gate0Steins	55	0	1	
997	kenshin1530	Gate2010Steins	0	0	0	
998	美幸	Gate2310Steins	5	4	0	
999	ふっか	Gate254Steins	228	153	7677	精神が不安定になるとたつやかニャンちゅうになるよご了承ください。RADWIMPSを愛してる。 WIMPERに片足突っ込んでます

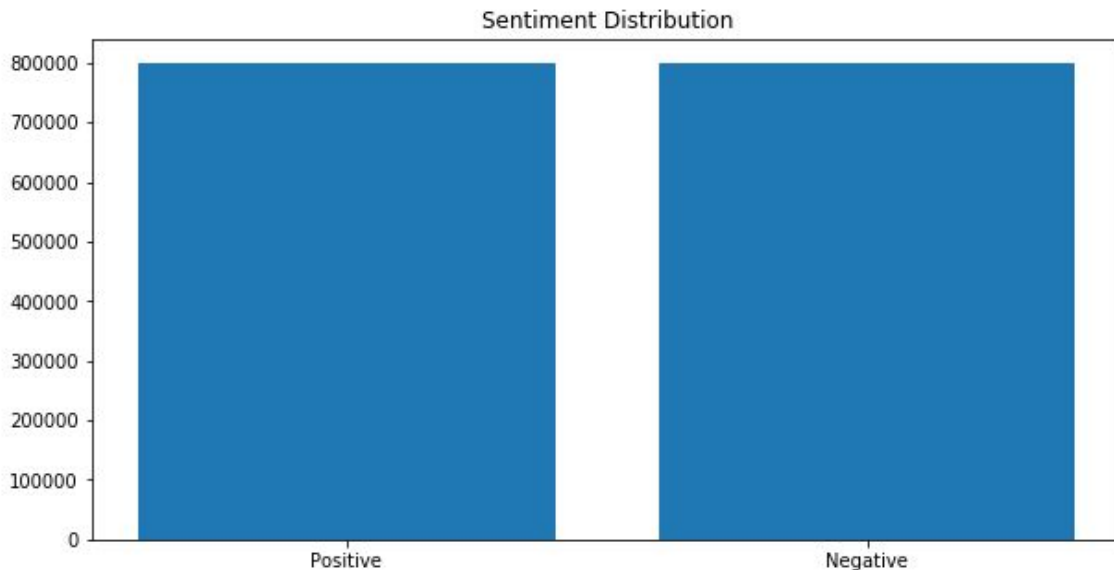
Bu hesaplar üzerinde tahmin yapıldı ve takibe geri dönebilecek olanlar hesaplar bulundu. Daha sonra rastgele hesap seçilerek takip edildi. 60 hesap takip edildi ve 4 hesap takibe geri döndü.

Sentiment Analysis

Sentiment Analiz yapmak için supervised veri seti olan Sentiment140[23] seçildi. Bu veri setinde 1.6 Milyon Tweet daha önceden “Negative” ve “Positive” olmak üzere 2 gruba göre ayrılmıştır. (0 Negatif Tweetler, 1 Pozitif Tweetler için.)

Sentiment	Tweet Id	Tweet Date	Flag	User Name	Tweet
0	0 1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Thi...
1	0 1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by texting it... and might cry as a result School to...
2	0 1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Managed to save 50% The rest go out of bounds
3	0 1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0 1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all. i'm mad. why am i here? because I can't see you a...

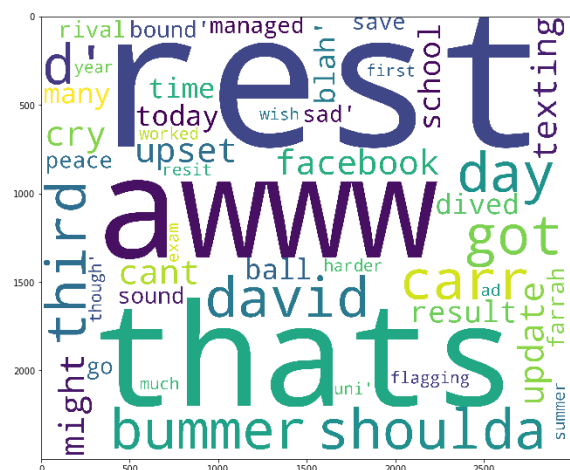
Sadece Sentiment ve Text kolonlarını alarak, Sentiment kolonundaki 0 ve 1 değeri “Negative” ve “Positive” e dönüştürüldü. Daha sonra veri dağılımına bakarak eş sayıda negatif ve pozitif tweet olduğu belirlendi.



Sentiment Analiz için öncelikle Tweetler üzerinde Post Processing işlemi yapıldı. Burada StopWords'ler Tweetlerden çıkarılıp gereksiz her türlü kullanıcı ismi, alfabe dışı karakterler silindi. Daha sonra her kelime Lemmatize edilerek en sade haline indirildi ve Cleaned Text kolonuna eklendi.

Sentiment	Tweet	Cleaned Text
0 Negative	@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Thi...	awww thats bummer you shoulda got david carr third day d
1 Negative	is upset that he can't update his Facebook by texting it... and might cry as a result School to...	upset cant update his facebook by texting and might cry result school today also blah
2 Negative	@Kenichan I dived many times for the ball. Managed to save 50% The rest go out of bounds	dived many time ball managed save 50 rest go out bound
3 Negative	my whole body feels itchy and like its on fire	my whole body feel itchy and like it fire
4 Negative	@nationwideclass no, it's not behaving at all. i'm mad. why am i here? because I can't see you a...	no it not behaving im mad am because cant see you over there
5 Negative	@Kwesidei not the whole crew	not whole crew
6 Negative	Need a hug	need hug
7 Negative	@LOLTrish hey long time no see! Yes.. Rains a bit ,only a bit LOL , I'm fine thanks , how's you ?	hey long time no see yes rain bit only bit lol im fine thanks hows you
8 Negative	@Tatiana_K nope they didn't have it	nope they didnt
9 Negative	@twittera que me muera ?	que muera

Pozitif ve Negatif Tweetlerde en çok sık geçen kelimeleri bulmak için Wordcloud kütüphanesi kullanıldı.



Daha sonra bu veri train ve test olmak üzere ikiye bölündü, 20% test için ayrıldı.

Splitting Train and Test

```
In [14]: from sklearn.model_selection import train_test_split

train, test = train_test_split(data, test_size = 0.2, random_state=0)
```

Veriler Sentiment Analiz’de kullanılabilmesi için verileri vektöre çevirmemiz gerekir. Bunun için ilk önce TfidfVectorizer fit edildi. Böylece train ve test verileri TfidfVectorizer yardımı ile vektöre dönüştürüldü.

TfidfVectorizer

```
In [15]: from sklearn.feature_extraction.text import TfidfVectorizer

In [16]: vectoriser = TfidfVectorizer(ngram_range=(1,2), max_features=500000)
vectoriser.fit(train["Cleaned Text"])
print('feature_words: ', len(vectoriser.get_feature_names()))

feature_words:  500000

In [17]: TweetTrain = vectoriser.transform(train["Cleaned Text"])
TweetTest = vectoriser.transform(test["Cleaned Text"])
```

BernoulliNB Model

BernoulliNB Model kullanılarak yapılan eğitim sonucu:

```
y_pred = BNB.predict(TweetTest)
print(classification_report(test["Sentiment"], y_pred))
```

	precision	recall	f1-score	support
Negative	0.81	0.76	0.79	159815
Positive	0.78	0.82	0.80	160185
accuracy			0.79	320000
macro avg	0.79	0.79	0.79	320000
weighted avg	0.79	0.79	0.79	320000

```
confusion_matrix(test["Sentiment"], y_pred)
```

```
array([[121867, 37948],
       [ 28512, 131673]], dtype=int64)
```

LinearSVC Model

LinearSVC Model kullanılarak yapılan eğitimin sonucu:

```
print(classification_report(test["Sentiment"], y_pred))
```

	precision	recall	f1-score	support
Negative	0.80	0.79	0.80	159815
Positive	0.79	0.80	0.80	160185
accuracy			0.80	320000
macro avg	0.80	0.80	0.80	320000
weighted avg	0.80	0.80	0.80	320000

```
confusion_matrix(test["Sentiment"], y_pred)
```

```
array([[126331, 33484],
```

Logistic Regression Model

Logistic Regression Model kullanılarak yapılan eğitimin sonucu:

```
LRM = LogisticRegression(C = 2, max_iter = 1000, n_jobs=-1)
LRM.fit(TweetTrain, train["Sentiment"])

y_pred = LRM.predict(TweetTest)
print(classification_report(test["Sentiment"], y_pred))
```

	precision	recall	f1-score	support
Negative	0.82	0.80	0.81	159815
Positive	0.80	0.82	0.81	160185
accuracy			0.81	320000
macro avg	0.81	0.81	0.81	320000
weighted avg	0.81	0.81	0.81	320000

```
confusion_matrix(test["Sentiment"], y_pred)

array([[127863, 31952],
       [ 28318, 131867]], dtype=int64)
```

En iyi sonucu verdiği için Logistic Regression Model üzerinde GridSearchCV kullanarak parametre geçişi yapıldı. C değeri üzerinde 1, 10, 100 ve 1000 değerleri denendi. En iyi sonuç 1 değerinde alındı. Son olarak Cross Validation işlemi yapılarak Doğruluk: 0.810 (0.001) elde edildi. Model Daha sonra kullanılmak üzere Pickle kütüphanesi yardımı ile kaydedildi.

PredictSentiment

Bu Notebook'da daha önce elde edilen Logistic Regression Modeli ile verilen Tweet'in Negatif veya Pozitif olduğu tahmin eden fonksiyon bulunmaktadır.

Main Notebook Üzerindeki İşlemler

Sentiment Analiz için öncelikle akıştaki Tweet'ler çekildi.

Sentiment Prediction

```
In [24]: Timeline = tw.timeline()
```

```
In [25]: Timeline.head()
```

Out[25]:

	Tweet	Tweet Id	Account
0	"That's what heroes are for" (via Fire Force Season 2) https://t.co/TPtkuw7dkv	130523479850443905	Crunchyroll
1	, I faint at the sight of a beautiful https://t.co/v6P0K1WtB's Suwa, right? She's been waiting ...	1305234533885894664	KurisuM64305605
2	Mayuri gets in touch with her father. The two are on their way to a summer camp.Mayuri and Daru ...	1305234474268086273	KurisuM64305605
3	Ninja Collection - Episode 9 - Critical Condition just launched! https://t.co/c7Vk3mRLBr https://...	1305234089335808006	Crunchyroll
4	A very small fraction of the world's population is alive in this era. If you take away that pers...	1305233379114319875	KurisuM64305605

Bu Tweetler üzerinde tek tek tahmin işlemi yapıldı. Daha sonra Sentiment sütunu oluşturularak Dataframe'e eklendi.

```
In [26]: TimeLine["Sentiment"] = TimeLine["Tweet"].apply(lambda tweet: ps.predict(tweet))
```

```
In [27]: TimeLine
```

```
Out[27]:
```

	Tweet	Tweet Id	Account	Sentiment
0	"That's what heroes are for" (via Fire Force Season 2) https://t.co/TPtkuw7dkv	1305234798504443905	Crunchyroll	Positive
1	, I faint at the sight of a beautiful https://t.co/v6P0KtWtIB's Suwa, right? She's been waiting ...	1305234533885894664	KurisuM64305605	Positive
2	Mayuri gets in touch with her father. The two are on their way to a summer camp.Mayuri and Daru ...	1305234474268086273	KurisuM64305605	Positive
3	Ninja Collection - Episode 9 - Critical Condition just launched! https://t.co/c7Vk3mRLBr https://...	1305234089335808006	Crunchyroll	Positive
4	A very small fraction of the world's population is alive in this era. If you take away that pers...	1305233379114319875	KurisuM64305605	Negative
5	RT @wlvpond: part 2! https://t.co/yJa243Feuw	1305232953631477765	themotherofants	Positive
6	RT @wlvpond: here she is!! donna noble once said (part 1) https://t.co/q6rQIEkHZN	1305232507823063040	themotherofants	Positive
7	RT @ReZero_En: Puppy or rabbit? https://t.co/4qemTfGpuH	1305230342543020033	Crunchyroll	Positive
8	Mahsuru yoksa bir adet soru alabilir miyim https://t.co/GPfOgUI13Q	1305227616329199616	themotherofants	Positive
9	NEWS: Sony Will Finally Have a PS5 Showcase on September 16!n/n More: https://t.co/UuLNSI3jdH h...	1305227317640335362	Crunchyroll	Positive
10	şöyle bir collab olsa güzel olmaz mıydı https://t.co/eH98eOX985	1305226844334100482	themotherofants	Positive
11	RT @FoodWarsAnime: Five Cuisine Fried Rice https://t.co/qSYjzOXIRF	1305226577425453057	Crunchyroll	Positive
12	RT @BestBoyEN: The Flavor of Success \n#FoodWars #ShokugekinoSoma https://t.co/UAYoF2e8uN	1305226543845773312	Crunchyroll	Positive
13	valla ben terk etmedim https://t.co/rFiTusjZUy	1305225617777061889	themotherofants	Positive
14	The newest Hello Panda flavor is here! Enjoy the same delicious, crème-filled crunchy treat now ...	1305223471853641730	Crunchyroll	Positive
15	It's Super Hacker, not Supah Hakah, duh.	1305222353920626693	labmem009_bot	Positive
16	Ternyata nggak ngantuk nggak enak ya	1305222053318987776	ivalfull	Positive
17	RT @ddawnchorus: Akira/アキラ (Katsuhiro Otomo, 1988) https://t.co/PUX38rLEpO	1305221883437092864	ivalfull	Positive

Bu Tweet'ler arasından kendi hesabımıza ait olmayan ve pozitif olarak tahmin edilen bir Tweet rastgele seçilip beğenildi.

```
In [30]: like = TimeLine[(TimeLine["Sentiment"] == "Positive") & (TimeLine["Account"] != "KurisuM64305605")]
like = like.sample(n = 1)
like
```

```
Out[30]:
```

	Tweet	Tweet Id	Account	Sentiment
11	RT @FoodWarsAnime: Five Cuisine Fried Rice https://t.co/qSYjzOXIRF	1305226577425453057	Crunchyroll	Positive

```
In [31]: for index,row in like.iterrows():
tw.like(row["Tweet Id"])
```

Sentiment Analiz haricinde Gif eklenmiş Tweet atma ve Tweet'e cevap verebilme özelliği Twitter Notebook'una eklendi.

Sending Gif Tweet

```
In [10]: tw.sendgif("Steins Gate",50,True)
```

Tweet: , I faint at the sight of a beautiful https://t.co/v6P0KtWtIB's Suwa, right? She's been waiting for you. https://t.co/DNng8ACqgD

Reply to Tweet from TimeLine

```
In [32]: reply = TimeLine[(TimeLine["Sentiment"] == "Positive") & (TimeLine["Account"] != "KurisuM64305605")].sample(n = 1)
reply
```

```
Out[32]:
```

	Tweet	Tweet Id	Account	Sentiment
14	The newest Hello Panda flavor is here! Enjoy the same delicious, crème-filled crunchy treat now ...	1305223471853641730	Crunchyroll	Positive

```
In [33]: tw.reply(tweetid = reply["Tweet Id"].values[0], Text =reply["Tweet"].values[0])
```

Tweet: @Crunchyroll @meijiamerica - Nintendo of America (@NintendoAmerica) July 26, 2017Hello Kitty is back in the spotlight after a while.

Dördüncü hafta daha önce yapılan tüm işlemleri otomatik bir şekilde, sürekli yapabilen ve Twitter API'nin kullanım şartlarını ihlal etmeyen Automatic Bot adında yeni bir Notebook yazıldı.

Automatic Bot

Notebook açılıştta daha önce yapılmış olan Mention ve Like'ları Pickle dosyasından çeker, bunun sebebi daha önce beğenilmiş veya cevaplanmış tweetlere tekrar cevap vermemek veya beğenmemek. (Beğenilmiş bir Tweet'i tekrar beğenmeye çalışmak API tarafından exception döndürür.)

```
In [5]: if os.path.exists("mentions_replied.pickle"):
        file = open('mentions_replied.pickle', 'rb')
        mentions_replied = pickle.load(file)
        file.close()
    else:
        mentions_replied = []
```

```
In [6]: if os.path.exists("retweet.pickle"):
        file = open('retweet.pickle', 'rb')
        retweets = pickle.load(file)
        file.close()
    else:
        retweets = []
```

Follow fonksiyonu verilen sayıda, anahtar kelime üzerinde yaptığı Twitter araması sonucu hesabı geri takibe dönüp dönmeyeceğini tahmin ederek rastgele takip eder.

SendTweet fonksiyonu rastgele bir şekilde gif eklenmiş veya eklenmemiş Tweet'ler atar.

```
In [8]: def sendtweet(hashtags):
        if(random.randint(1,5) <= 4):
            tw.sendtweet(hashtags = ["SteinsGate"])
        else:
            tw.sendgif(["Steins Gate"],random.randint(1,100),random.randint(0,1) == 0,0)
        return tw.gettweets()
```

Followback fonksiyonu hesabı takip etmiş hesapları takip eder.

```
In [9]: def followback():
        following_accounts = following["Screen Name"].values
        followers["Followed"] = followers["Screen Name"].apply(lambda x: x in following_accounts)
        for index, row in followers[followers["Followed"] == False].iterrows():
            tw.follow(row["Screen Name"])
        return tw.following(),followers[followers["Followed"] == False]
```

Replymentions fonksiyonu gelen Mention'lara cevap verir.

```
In [10]: def replymentions():
        mentions = tw.mentions()
        replies = []
        for index, row in mentions.iterrows():
            if row["Tweet Id"] not in mentions_replied:
                replies.append(tw.reply(tweetid = row["Tweet Id"], Text = translator.translate(row["Tweet"])))
                mentions_replied.append(row["Tweet Id"])
        file = open('mentions_replied.pickle', 'wb')
        pickle.dump(mentions_replied, file)
        file.close()
        return replies
```

Reply fonksiyonu hesabın ana akışındaki veya Twitter aramasından geri dönen rastgele Tweet'lere yanıt verir.

Like fonksiyonu hesabın ana akışındaki veya Twitter aramasından geri dönen Tweet'ler üzerinde Sentiment Analysis yaparak Tweet'lerin pozitif veya negatif olduklarını bulur ve pozitif olanlardan rastgele seçilenleri beğenir.

```
In [12]: def like(count = 1):
    if(random.randint(0,1) == 0):
        tweets_ = tw.timeline()
    else:
        tweets_ = tw.searchtweet(hashtags[random.randint(0,len(hashtags)-1)])
    liked = tw.getlikes()
    tweets_ = tweets_[~tweets_["Tweet Id"].isin(liked["Tweet Id"].values)]
    tweets_["Sentiment"] = tweets_["Tweet"].apply(lambda tweet: ps.predict(translator.translate(tweet)))
    if len(tweets_[(tweets_["Sentiment"] == "Positive") & ((tweets_["Account"] != account["Screen Name"])] & ((not "RT" in tweet:
        n = count % len(tweets_) if count % len(tweets_) != 0 else 1
        tweets_ = tweets_[(tweets_["Sentiment"] == "Positive") & ((tweets_["Account"] != account["Screen Name"])] & ((not "RT" in
        for index,row in tweets_.iterrows():
            tw.like(row["Tweet Id"])
        return tweets_
    else:
        return pd.DataFrame()
```

Retweet fonksiyonu hesabın ana akışındaki veya Twitter aramasından geri dönen Tweet'lerden rastgele seçilenleri yorumlu veya yorumuz retweet yapar.

```
In [13]: def retweet(count = 1):
    if(random.randint(0,1) == 0):
        searchtweet = tw.timeline()
    else:
        searchtweet = tw.searchtweet(hashtags[random.randint(0,4)])
    searchtweet = searchtweet[~searchtweet["Tweet Id"].isin(retweets)]
    searchtweet = searchtweet.sample(n = count % len(searchtweet))
    for index,row in searchtweet.iterrows():
        if row["Tweet Id"] not in retweets and row["Account"] != account["Screen Name"]:
            tw.retweet(Text = row["Tweet"], tweet_id = row["Tweet Id"], screenname = row["Account"])
            retweets.append(row["Tweet Id"])
    file = open('retweet.pickle', 'wb')
    pickle.dump(retweets, file)
    file.close()
    return searchtweet["Tweet"][0:count]
```

Geriye kalan kısımda ise bu fonksiyonlar belli süre aralıkları beklenerek çalıştırılır.

Tweet örnekleri





Kurisu Makise
@KurisuM64305605

So basically, it is the beginning of the end, and the beginning of the beginning.



Darkstorm @Meepok1 · Sep 15

Replying to @chinatsu_stan

No. Steins Gate 0 happens after SG, but ends at the beginning of SG.

11:33 PM · Sep 17, 2020 · Meydan oku

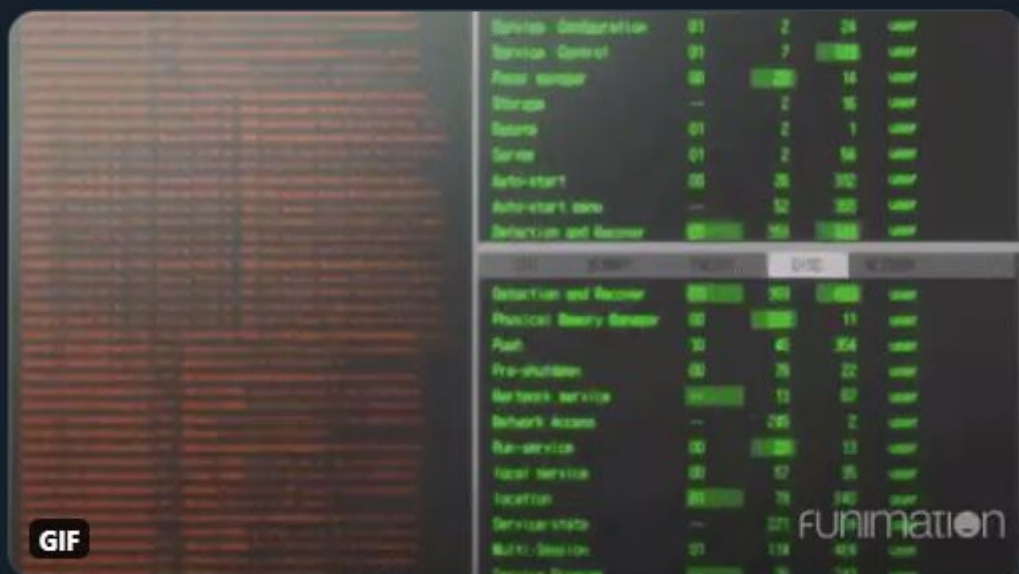
1 Retweet 1 Like



Kurisu Makise @KurisuM64305605 · 2h

.I try to force down on the answers, but they're just ideas. The only one who knows them all is me. I'll try to organize it. I'll try my best to fix it.

#SteinsGate



F. Staj deęerlendirmesi

Staj boyunca Beautiful Soup ile veri toplama, verileri dzenleme ve Sklearn ile Sınıflandırma ve Sentiment analiz gibi farklı makine ęrenmesi iřlemleri yapıldı. Twitter API kullanıldı ve Twitter zerinden veri toplaması gerekleřtirildi. Veri toplama ařaması staj boyunca sren en uzun iřlemlerden biri oldu bunun sebebi ise Twitter'ın API kurallarının belli bir srede kısıtlı istek kabul etmesidir.

G. Kaynak Kod ve Dosyalar

https://drive.google.com/file/d/1Va6Wxvedl1O_4bwJniRYcsZx9QpDduhN/view?usp=sharing

H. Referanslar

1. "Numpy", <https://numpy.org/> (Son Erişim: 25 Ağustos 2020)
2. "Pandas", <https://pandas.pydata.org/> (Son Erişim: 26 Ağustos 2020)
3. "Scikit-learn: machine learning in Python", <https://scikit-learn.org/stable/> (Son Erişim: 26 Ağustos 2020)
4. "Tweepy", <https://www.tweepy.org/> (Son Erişim: 26 Ağustos 2020)
5. "GPT 2 Natural Language Processing Model", <https://openai.com/blog/better-language-models/> (Son Erişim: 26 Ağustos 2020)
6. "Jupyter Notebook", <https://jupyter.org/> (Son Erişim: 26 Ağustos 2020)
7. "Welcome to Python.org", <https://www.python.org/> (Son Erişim: 26 Ağustos 2020)
8. "Anaconda | The World's Most Popular...", <https://www.anaconda.com/> (Son Erişim: 26 Ağustos 2020)
9. "TensorFlow", <https://www.tensorflow.org/> (Son Erişim: 28 Ağustos 2020)
10. "Beautiful Soup Documentation", <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (Son Erişim: 28 Ağustos 2020)
11. "Welcome To Colaboratory", <https://colab.research.google.com/> (Son Erişim: 28 Ağustos 2020)
12. "translate .Pypl", <https://pypi.org/project/translate/> (Son Erişim: 28 Ağustos 2020)
13. "pickle — Python object serialization", <https://docs.python.org/3/library/pickle.html#module-pickle> (Son Erişim: 28 Ağustos 2020)
14. "Gfycat Developer Portal", <https://developers.gfycat.com/> (Son Erişim: 28 Ağustos 2020)
15. "WordCloud for Python documentation", https://amueller.github.io/word_cloud/ (Son Erişim: 28 Ağustos 2020)
16. "NumPy Introduction", https://www.w3schools.com/python/numpy_intro.asp (Son Erişim: 29 Ağustos 2020)
17. "Pandas'a Hızlı Başlangıç", <https://veribilimcisi.com/2017/07/14/pandasa-hizli-baslangic/> (Son Erişim: 29 Ağustos 2020)
18. "scikit-learn/scikit-learn", <https://github.com/scikit-learn/scikit-learn> (Son Erişim: 29 Ağustos 2020)

19. "What is Python?", <https://docs.python.org/3/faq/general.html#what-is-python> (Son Erişim: 29 Ağustos 2020)
20. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, "Attention Is All You Need", [arXiv:1706.03762](https://arxiv.org/abs/1706.03762) [cs.CL] (Son Erişim: 29 Ağustos 2020)
21. "Steins;Gate Wiki", https://steins-gate.fandom.com/wiki/Steins;Gate_Wiki (Son Erişim: 29 Ağustos 2020)
22. "parchive", <http://parchive.org/> (Son Erişim: 29 Ağustos 2020)
23. "Sentiment140", <http://help.sentiment140.com/for-students> (Son Erişim: 1 Eylül 2020)