دانشکده مهندسی کامپیوتر

تمرین درس

پردازش زبان طبیعی

استاد درس: دکتر مینایی

سجاد رمضانی 96471298
یاسمین مدنی 97532265

نیم‌سال دوم
سال تحصیلی ۰۱-۰۰

فاز ۱

۱.۱ مقدمات

در این پروژه ما قصد داریم بتوانیم اخبار را بر اساس عنوان آنها دسته بندی کنیم موضوع داده تیتر خبرهای موجود در سایت های خبری و دسته بندی آن است. که ما به دو زبان فارسی و انگلیسی دیتا جمع آوری کرده ایم از این جهت که اگر بخواهیم مدل خوبی را پیشنهاد دهیم باید به گونه ای باشد که قابلیت تعمیم پذیری به چند زبان را داشته باشد از آنجا که داده ها به صورت تیتر خبر هستند پس زبان آنها رسمی بوده و محاوره ای نیست.

۲.۱ داده

داده های گرده آوری شده در دو زبان فارسی و انگلیسی و همینطور به دو صورت دیتا های اسکریپ شده از وب سایت های خبری و داده های استاندارد برای این تسک بوده. در قسمت داده های اسکریپ شده از ابزار Scrapy برای جمع آوری داده ها استفاده شده و چهاراسپایدر برای وب سایت های متفاوت نوشته شده است. که به صورت خلاصه در زیر بیان شده است

- HuffingPostSpider برای استخراج داده های سایت huffpost که شامل اخبار از دسته های گوناگون می باشد.
- Techcrunch Scrapper: اسپایدر برای استخراج داده های سایت تک کرانچ که شامل اخبار مرتبط با تکنولوژی است
- Nytimes Rss Spider: اسپایدر پارس کننده اخبار سایت New York times

- اسپایدر سایت خبر انلاین: برای یکی از مهم ترین وب سایت های خبری ایران هم یک اسپایدر نوشته ایم

جدا از موارد بیان شده از دیتا ست های آماده برای این تسک نیز استفاده شده است که به شرح زیر است

- News Category Dataset که دیتاستی از همان سایتhuffpost که بخشی از دیتا را از آن به دست آورده ایم می باشد برای گرفتن دیتاست می توانید به اینجا مراجعه فرمایید
- BBC news Data: داده های اخبار سایت BBC برای گرفتن دیتاست می توانید به اینجا مراجعه فرمایید
- دیتا ست دیگر داده های اخبار افتصادی است که درکگل منتشر شده است که می توان در اینجا مشاهده کنید

۱.۲.۱ جمع آوری دیتا

در این قسمت توضیح کوتاهی درمورد نحوه جمع آوری دیتا و اسپایدر های نوشته شده می پردازیم

همانطور که گفته شد با استفاده از فریم ورک Scrapy به راحتی می توان اسپایدر برای وب سایت های متفاوت نوشت و سپس با نوشتن پایپلاین مناسب این دیتا ها را در فرمت مناسب نگه داری کرد. که در اینجا یک نمونه از اسپایدر ها را بررسی می کنیم با استفاده از این فریم ورک ابتدا یک کلاس می نویسیم که با ارث بری از کلاس های موجود تنها کافی است چند متد آن را پیاده سازی کنیم برای این کار به مثال گردآوری داده از وب سایت خبر انلاین توجه کنید

ابتدا لینک و فرمت پارامتر های ان در متغیر urls آورده شده است که در ان دو پارامتر tp که در واقع ایدی موضوع است و pi که شماره صفحه است را مشاهده می کند که برای شروع ۱۰۰۰ تا از ایدی ها برای موضوعات متفاوت داده شده است. سپس هر صفحه ایی که درخواست داده می شود پس از گرفتن html ان به تابع parse داده می شود تا قسمت های مورد نیاز از آن استخراج شود که با نگاه کردن به فرمت صفحه و استخراج css selector های مناسب این موارد استخراج می شود و تا صفحه مشخص شده در page limit از اخبار این دسته بندی آورده می شود.

و در اخر هم یک ابجکت از نوع Title Item برگردانده می شود

برای نوشتن این موارد هم یک pipeline ارایه شده است که در فرمت جیسون دیتا ها را بنویسید که کد آن هم در زیر ارایه شده است. برای مشاهده دقیق تر این موارد به ریپومراجعه شود

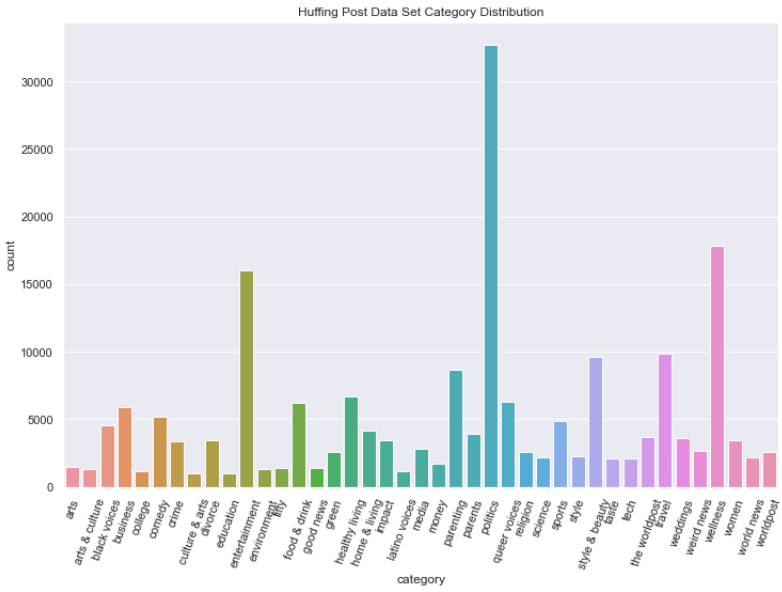
```python
class KhabarOnlineScraper(scrapy.Spider):
    name = "khabaronline"
    page_limit = 200
    def start_requests(self):
        urls = [
            f"https://www.khabaronline.ir/archive?pi=1&tp={i}"
            for i in range(10000)
        ]
        for url in urls:

            yield scrapy.Request(url=url, callback=self.parse)
    def parse(self, response):
        for item in response.css('section[id="box202"]').css("ul")[0].css("li"):
            title = item.css("h3").css("a::text")[0].get()
            category = item.css("p").css("span").css("a::text").get()
            yield TitleItem(title=title, category=category)
        try:
            if int(re.search("pi=(\d+)", response.url)
            .groups(1)) < self.page_limit:
                yield scrapy.Request(
                    re.sub(
                        "pi=(\d+)",
                        lambda exp: "pi={}".format(int(exp.groups()[0]) + 1),
                        response.url,
                    ),
                    self.parse,
                )
        except:
            pass
```

```python
class JsonWriterPipeline:

    def open_spider(self , spider):
        self.file = open(ITEM_OUTPUT_PATH , 'w',encoding='utf-8')

    def close_spider(self , spider):
        self.file.close()

    def process_item(self, item, spider):
        line = json.dumps(ItemAdapter(item).asdict(),ensure_ascii=False) + '\n'
        self.file.write(line)
        return item
```

۲.۲.۱ دیتاست های آماده

news-category-dataset

این مجموعه داده شامل حدود ۲۰۰ هزار عنوان خبری از سال ۲۰۱۲ تا ۲۰۱۸ است که از HuffPost به دست آمده است. هر عنوان خبری یک دسته بندی مربوطه دارد که دسته بندی ها را در شکل زیر مشاهده می کنید.



Dataset: BBC

شامل ۲۲۲۵ داده از وب‌سایت خبری بی‌بی‌سی در پنج حوزه موضوعی از سال ۲۰۰۴ تا ۲۰۰۵ است. دارای ۵ کلاس از نوع (کسب و کار، سرگرمی، سیاست، ورزش، فناوری) می باشد

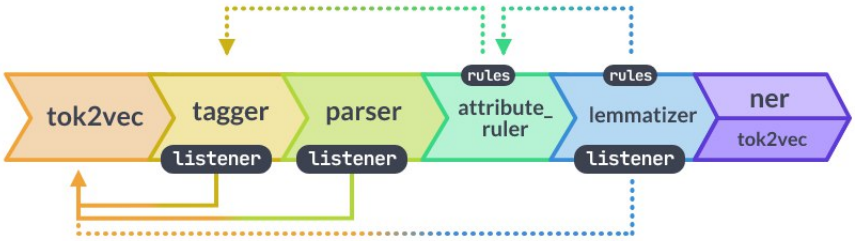
Category	Heading	Article
Business	510	510
Entertainment	386	386
Politics	417	417
Sport	511	511
Tech	401	401

US Financial News Articles

داده های مربوط به اخبار اقتصادی جمع آوری شده از سایت های Bloomberg.com, CNBC.com, reuters.com, wsj.com, fortune.com

۳.۱ پیش پردازش

در این مرحله نیاز داریم تا داده های جمع آوری شده را تمیز و پیش پردازش کنیم که روش های گوناگون و تسک های مختلفی از جمله ریشه گیری کلمات، حذف استاپ وردها و ... دارد. در اینجا از پایپ لاین آموزش داده شده ی en_core_web_sm استفاده می کنیم این پایپ لاین ابتدا متن را نشانه‌گذاری می‌کند تا یک شی Doc تولید کند. سپس Doc در چندین مرحله مختلف پردازش می‌شود. خط لوله معمولاً شامل یک tagger یک lemmatizer، یک parserو یک entity recognizer است. که هر یک از این بخش ها متن پردازش شده را به بخش بعدی می فرستد. برای مشاهده داک مدل به اینجا مراجعه کنید.



در ادامه در تابع زیر باقی عملیات لازم جهت پاکسازی متن را انجام می دهیم. که این شامل حذف استاپ وردها،اعداد،فاصله هاو..می شود.

```
def clean_doc(d):
    doc = []
    for t in d:
        if not any([t.is_stop, t.is_digit,
        not t.is_alpha, t.is_punct, t.is_space,
        t.lemma_ == '-PRON-']):
            doc.append(t.lemma_)
    return ' '.join(doc)


def preprocess(articles):
    iter_articles = (article for article in articles)
    clean_articles = []
    for i, doc in
    enumerate(nlp.pipe(iter_articles, batch_size=100,n_process=8), 1):
        if i % 1000 == 0:
            print(f"{i / len(articles):2%.}", end=" ", flush=True)
        clean_articles.append(clean_doc(doc))
    return clean_articles
```

برای داشتن دید بهتر تعدادی مثال از تیتر اخبار پیش و پس از پیش پردازش در تصاویر زیر به نمایش گذاشته شده است.

داده پیش و پس از پردازش

raw: Captive Medic's Bodycam Shows Firsthand Horror Of Mariupol
cleaned: Captive Medic Bodycam show Firsthand Horror Mariupol

داده پیش و پس از پردازش

raw: Russia Is Firing Its Senior Commanders. What Does That Mean For Ukraine War?
cleaned: Russia fire senior commander mean Ukraine War

داده پیش و پس از پردازش

raw: LinkedIn Settles With U.S. Over Alleged Pay Discrimination
cleaned: LinkedIn Settles Alleged Pay discrimination

داده پیش و پس از پردازش

raw: Starbucks Workers Have Unionized More Than 50 Stores In The U.S.
cleaned: Starbucks Workers unionize Stores

یک مثال از شرایطی که می تواند در تصمیم گیری مدل در آینده تاثیر بگذارد

raw: LinkedIn Settles With U.S. Over Alleged Pay Discrimination
cleaned: LinkedIn Settles Alleged Pay discrimination

۴.۱ آمار مربوط به دیتاهای جمع آوری شده

از آنجا که دیتاهای مختلفی جمع آوری کرده ایم در بخش زیر تعدادی نمودار مربوط به هر یک از این دیتاستهارا به نمایش گذاشته ایم اما از آنجا که انتظار می رود داده های جمع شده از سایت huffpostو دیتاست آن پاسخ بهتری در فازهای بعدی برای ما فراهم کند آمار خواسته شده در داک پروژه را در مورد این دیتاست نوشته ایم. برای بدست آوردن این اماره ها از توابع موجود در nltk استفاده شده است.

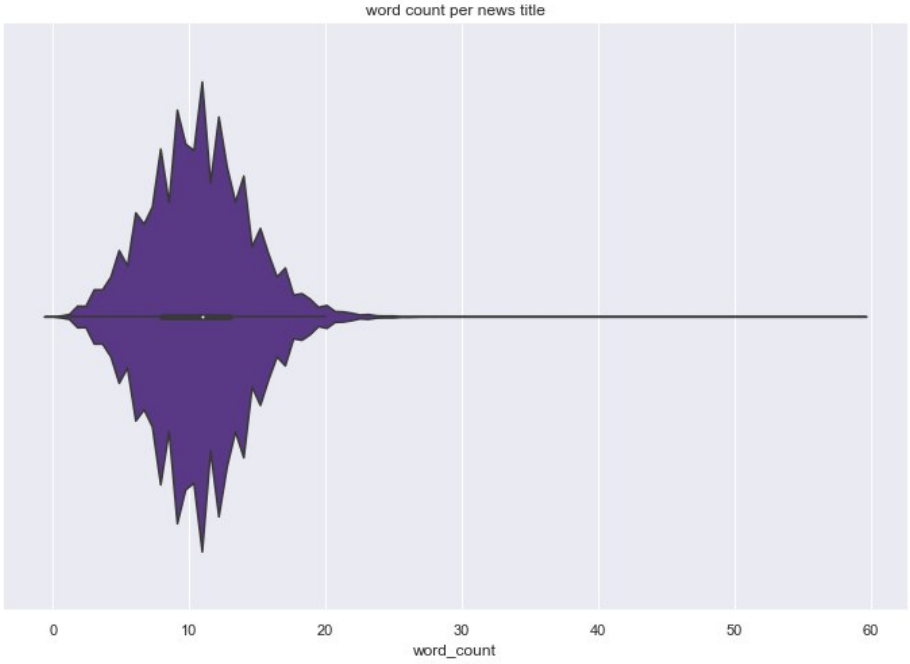
برای تفکیک جملات از sent_tokenize استفاده شده است. که در صورت پیش فرض با استفاده از punctuation ها می تواند جملات را تفکیک کند برای مطالعه بیشتر به اینجا مراجعه کنید

برای تفکیک کلمات از word_tokenize استفاده شده است. که در صورت پیش فرض با استفاده از TreebankWordTokenizer استفاده می کند.

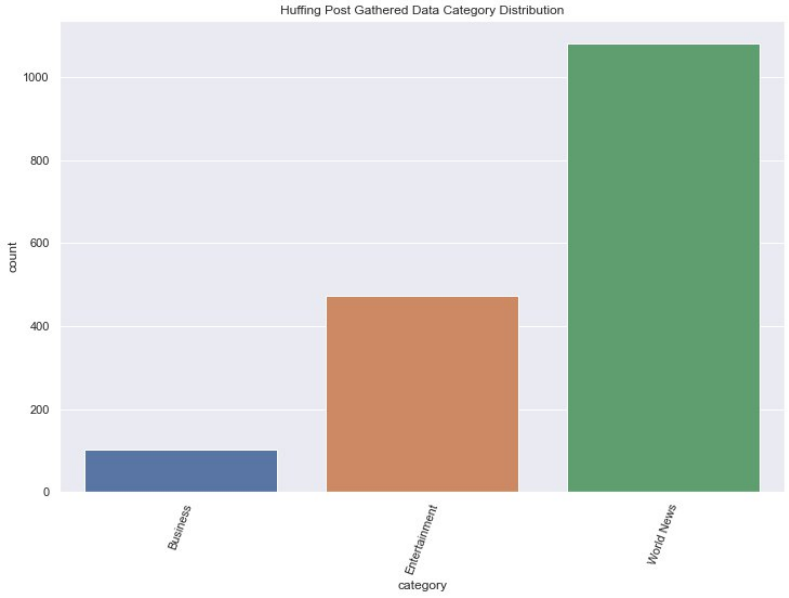
huffpost ۱.۴.۱

- تعداد واحد داده : ۲۰۱۰۵۹ عنوان خبر
- تعدا جملات : ۲۰۹۵۵۷
- تعداد کلمات ۲۱۴۶۸۹۱
- تعداد کلمات منحصر به فرد: ۶۴۷۳۲(بعد تمیز کردن)
- تعداد کلمات منحصر به فرد: ۷۴۵۰۵(قبل تمیز کردن)

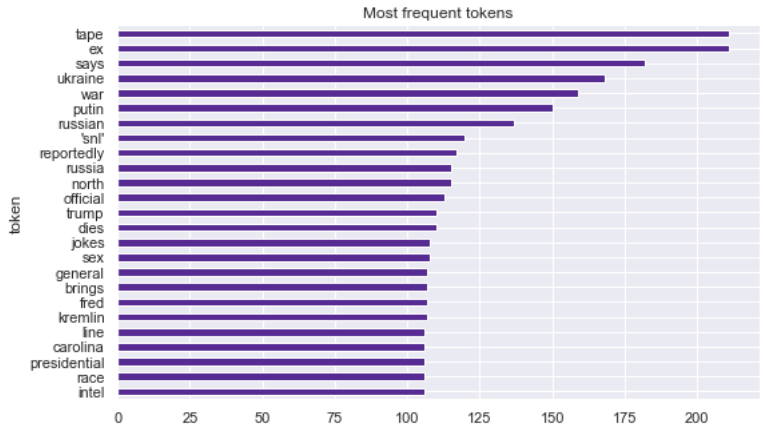
تعداد توکنهای موجود برحسب هدلاین

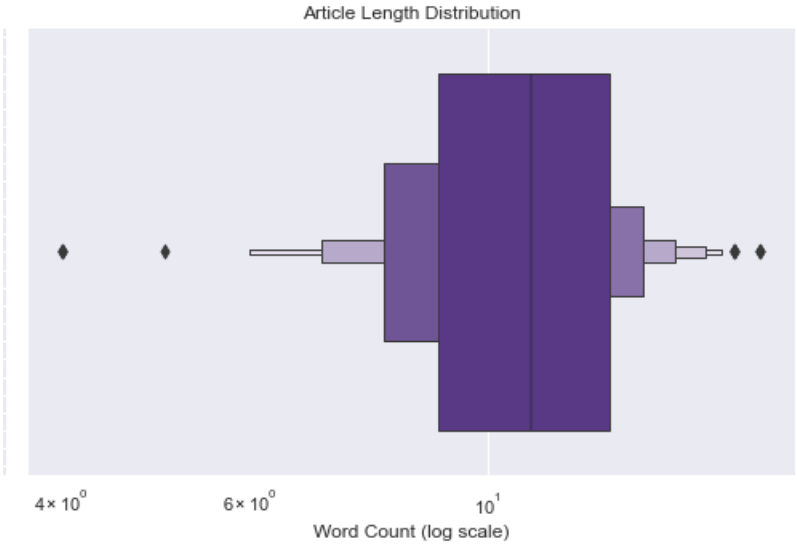


برچسب داده ها و تعداد آنها در داده های کرال شده از سایت

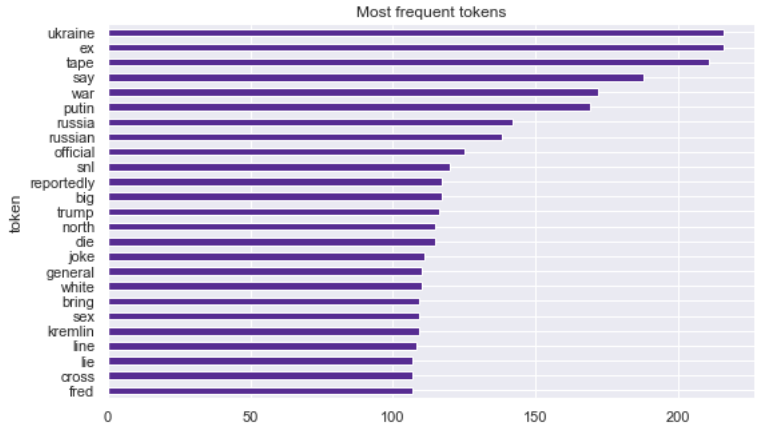


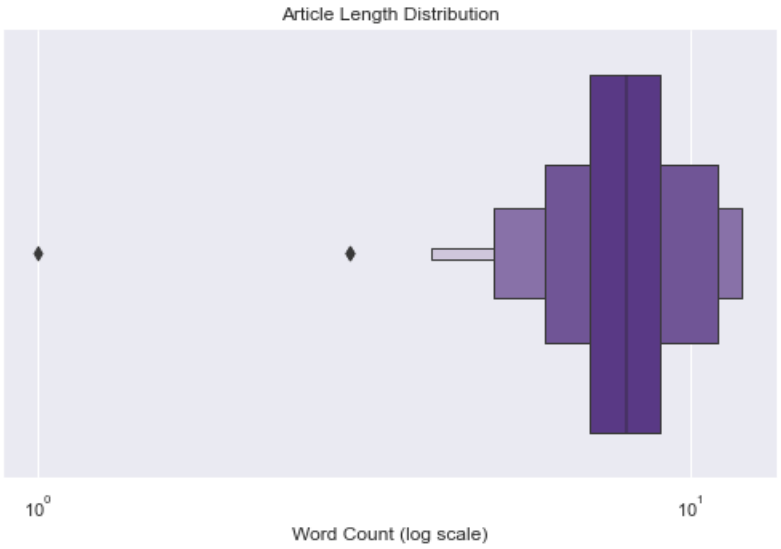
در شکل زیر نمودارهای توکن های پرتکرار و طول جملات داده های جمع آوری شده را مشاهده می کنیم.



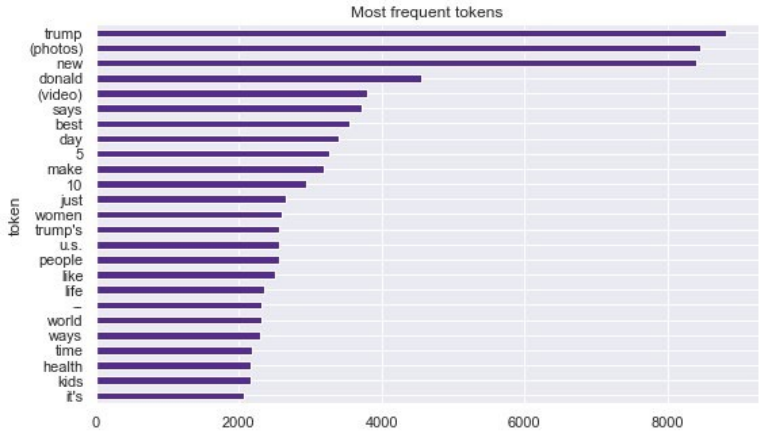


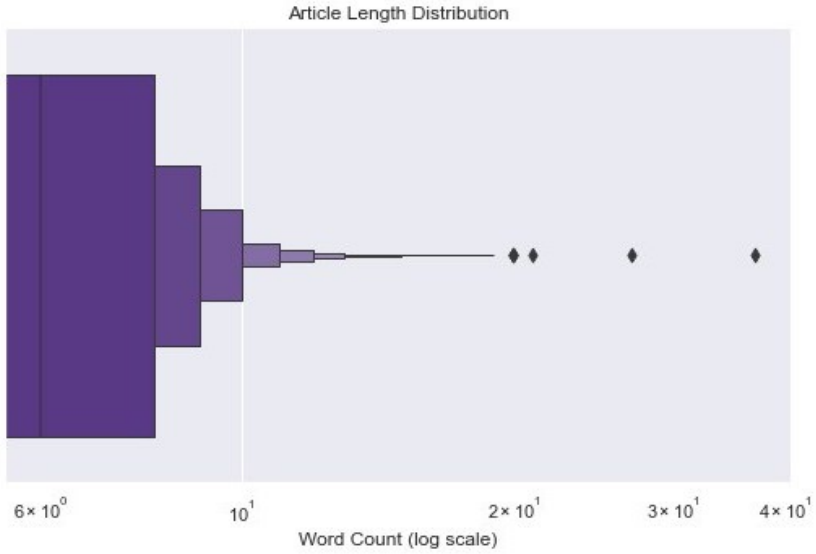
همچنین در شکل زیر نمودارهای مربوطه را پس از عملیات پیش پردازش می توان مشاهده کرد

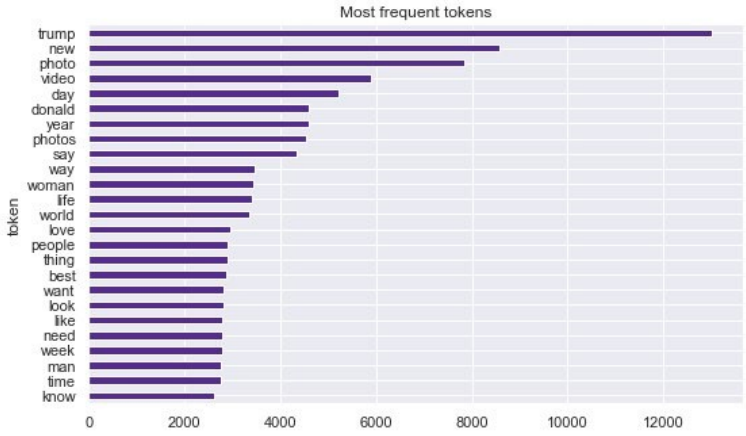


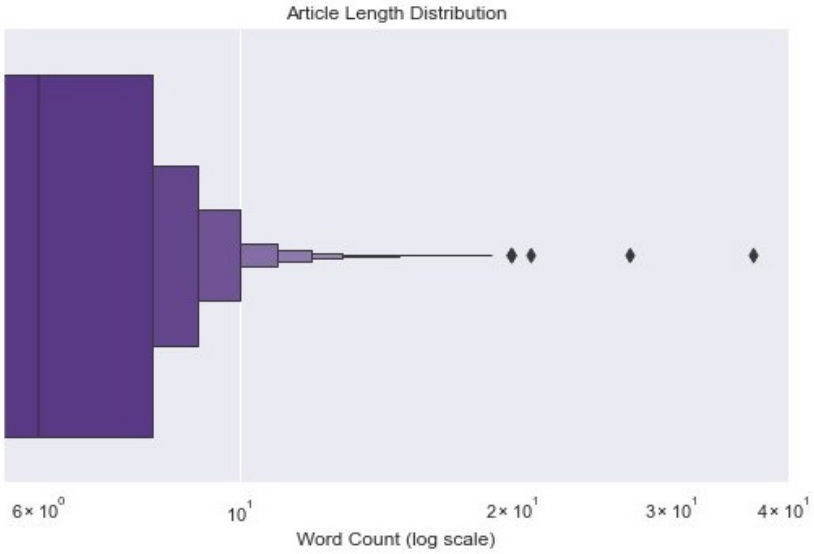


به علاوه بخشی از داده های ما از طریق دیتاست توضیح داده شده پیش از این تامین شده از این رو این عملیات پیش پردازش را روی داده های موجود در دیتاست نیز اجرا می کنیم همان طور که در نمودارهای زیر مشاهده می شود داده های موجود در دیتا بیس حاوی مقادیر اطلاعات از قبیل اعداد و ... اند که با انجام عملیات پیش پردازش حذف شده و اولویت در ترتیب توکن های پر تکرار تغییر می کند.


Most frequent tokens
token
trump
(photos)
new
donald
(video)
says
best
day
5
make
10
just
women
trump's
u.s.
people
like
life
–
world
ways
time
health
kids
it's
0
2000
4000
6000
8000

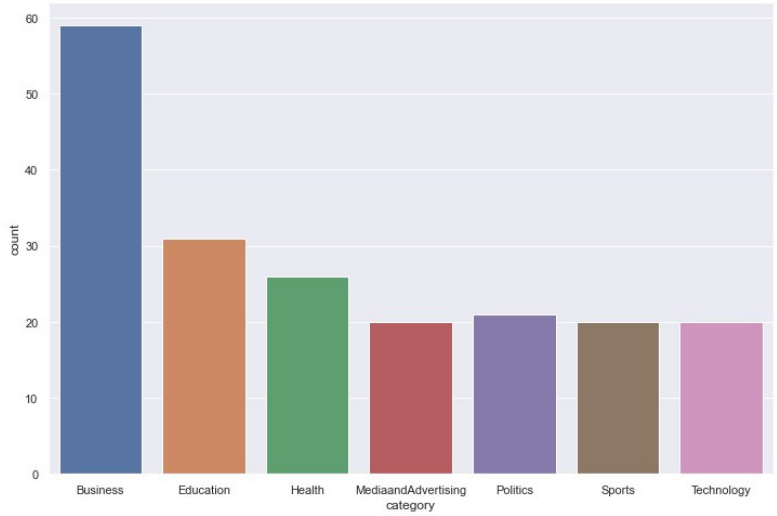

Article Length Distribution
6× 10^0
10^1
2× 10^1
3× 10^1
4× 10^1
Word Count (log scale)


Most frequent tokens
token
trump
new
photo
video
day
donald
year
photos
say
way
woman
life
world
love
people
thing
best
want
look
like
need
week
man
time
know
0
2000
4000
6000
8000
10000
12000


Article Length Distribution
6× 10^0
10^1
2× 10^1
3× 10^1
4× 10^1
Word Count (log scale)

۲.۴.۱ nyt

برچسب داده ها و تعداد آنها در داده های کرال شده از سایت



۳.۴.۱ خبرآنلاین

برچسب داده ها و تعداد آنها در داده های کرال شده از سایت