تمرین اول NLP

یاسمین مدنی

۹۷۵۳۲۲۶۵

فهرست

سوالات تئوری

سوال ۱

Natural Language Tool Kit

یکی از محبوب ترین کتابخانه های NLP در پایتون است. از تعداد زیادی کار پشتیبانی می‌کند و می‌تواند برای انجام هر کاری از تکنیک‌های پیش‌پردازش متن مانند توقف حذف کلمه، توکن‌سازی، ریشه‌یابی و واژه‌سازی تا ساخت n-gram استفاده شود.

NLTK رابط کاربری آسانی را برای بیش از ۵۰ مجموعه و منبع واژگانی فراهم می کند. این ابزار دارای عملکردهای ضروری مورد نیاز برای تقریباً همه انواع وظایف پردازش زبان طبیعی با پایتون است وتقریباً هر مؤلفه ای از NLP را که نیاز دارید اجرا می کند، مانند طبقه بندی، نشانه گذاری، ریشه یابی، برچسب گذاری، تجزیه و استدلال معنایی.اغلب بیش از یک پیاده سازی برای هر کدام وجود دارد، بنابراین می توانید الگوریتم یا روش دقیقی را که می خواهید استفاده کنید انتخاب کنید. همچنین از بسیاری از زبان ها پشتیبانی می کند.

fastText

یک کتابخانه منبع باز، رایگان و سبک برای یادگیری جاسازی کلمات و طبقه بندی متن است که توسط آزمایشگاه تحقیقاتی هوش مصنوعی فیس بوک ایجاد شده است. این مدل به فرد امکان می دهد یک الگوریتم یادگیری بدون نظارت یا یادگیری نظارت شده برای بدست آوردن نمایش های برداری برای کلمات ایجاد کند. فیس بوک مدل های از پیش آموزش دیده را برای ۲۹۴ زبان در دسترس قرار می دهد.

SpaCy

سریع، آسان برای استفاده، به خوبی مستند شده، و برای پشتیبانی از حجم زیادی از داده ها طراحی شده است،دارای مجموعه ای از مدل های NLP از پیش آموزش دیده است که کار شما را آسان تر می کند. برخلاف NLTK یا CoreNLP که تعدادی الگوریتم را برای هر کار نمایش می دهند، SpaCy منوی خود را کوتاه نگه می دارد و بهترین گزینه موجود را برای هر کار در دست ارائه می دهد.

اگر می خواهید متنی را برای یادگیری عمیق آماده کنید، این کتابخانه یک گزینه عالی است و در کارهای استخراج عالی است. در حال حاضر، فقط به زبان انگلیسی در دسترس است.

Scikit-learn

یک کتابخانه NLP منبع باز رایج در میان دانشمندان داده به دلیل مستندات عالی است. علاوه بر این، روش های کلاس بصری را ارائه می دهد و الگوریتم های متعددی را برای ساخت مدل های یادگیری ماشین ارائه می دهد.با این حال، Scikit-learn شبکه های عصبی را برای پردازش متن ارائه نمی دهد.

CoreNLP

هدف آن این است که استفاده از ابزارهای تحلیل زبانی را برای یک متن آسان و کارآمد کند. با CoreNLP، می‌توانید انواع ویژگی‌های متن (مانند تشخیص هویت نام، برچسب‌گذاری بخشی از گفتار، و غیره) را تنها در چند خط کد استخراج کنید.

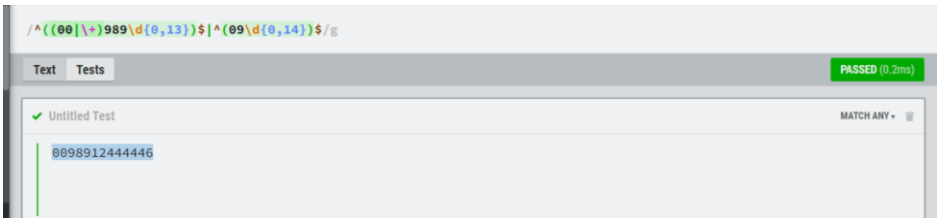
از آنجایی که CoreNLP به زبان جاوا نوشته شده است، نیاز به نصب جاوا بر روی دستگاه شما دارد. با این حال، رابط های برنامه نویسی را برای بسیاری از زبان های برنامه نویسی محبوب از جمله پایتون ارائه می دهد. این ابزار شامل چندین ابزار NLP استندفورد مانند تجزیه کننده، تجزیه و تحلیل احساسات، یادگیری الگوی بوت استرپ، برچسب بخشی از گفتار (POS) و...است.

علاوه بر این، CoreNLP از زبان‌های عربی، چینی، آلمانی، فرانسوی و اسپانیایی علاوه بر انگلیسی پشتیبانی می کند.

سوال ۲

الف)

^((00|\+)989\d{0,13})$|^(09\d{0,14})$

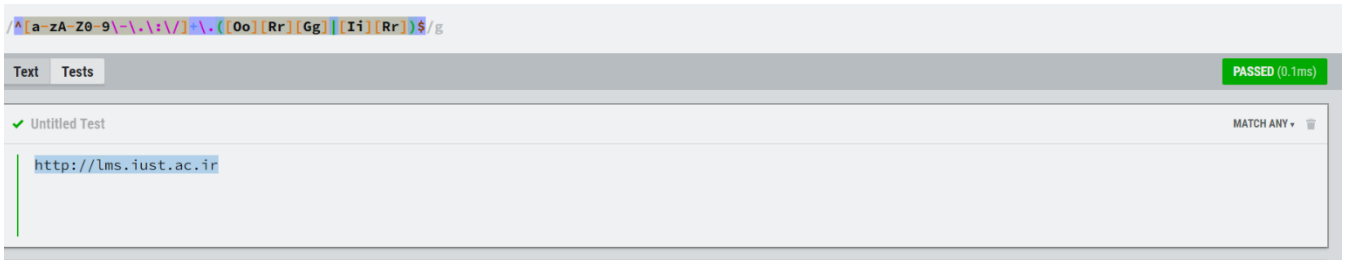


ب)

^((0? [1-9])| [12][0-9]| 3[01]) - (0? [1-9] | 1[012]) - [0-9]{4}$

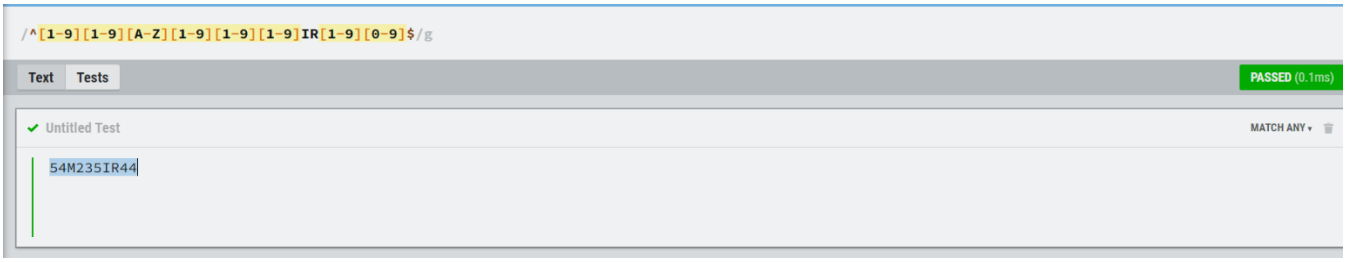
ج)

^[a-zA-Z0-9\-\.\:\/]+ \. ([Oo][Rr][Gg]|[Ii][Rr])$



(د

^[1-9][1-9][A-Z][1-9][1-9][1-9]IR[1-9][0-9]$



سوال ۳

روش Maximal Matching

یکی از راه‌های جلوگیری از تطابق کوتاه‌ترین کلمه، یافتن طولانی‌ترین دنباله کاراکترها در فرهنگ لغت است. این روش یک الگوریتم حریصانه است که با طولانی ترین کلمه مطابقت دارد.

مثلا در زبان انگلیسی ما این سری کاراکتر را داریم:

“themendinehere”

برای کلمه اول کلمات زیر را می یابیم و بعد از آن کلمه ای پیدا نمی شود.

the, them, theme

حالا ما طولانی ترین را انتخاب می کنیم که «theme » است، سپس دوباره از «n» شروع می کنیم. اما در حال حاضر ما هیچ کلمه ای در رشته زیرنداریم.

“ndineh…”

وقتی نمی توانیم یک کلمه را مطابقت دهیم، فقط اولین کاراکتر را به عنوان ناشناخته علامت گذاری می کنیم.پس اینجا n را به عنوان ناشناخته علامت گذاری و جدا می کنیم و کلمه بعد را با d شروع می کنیم.

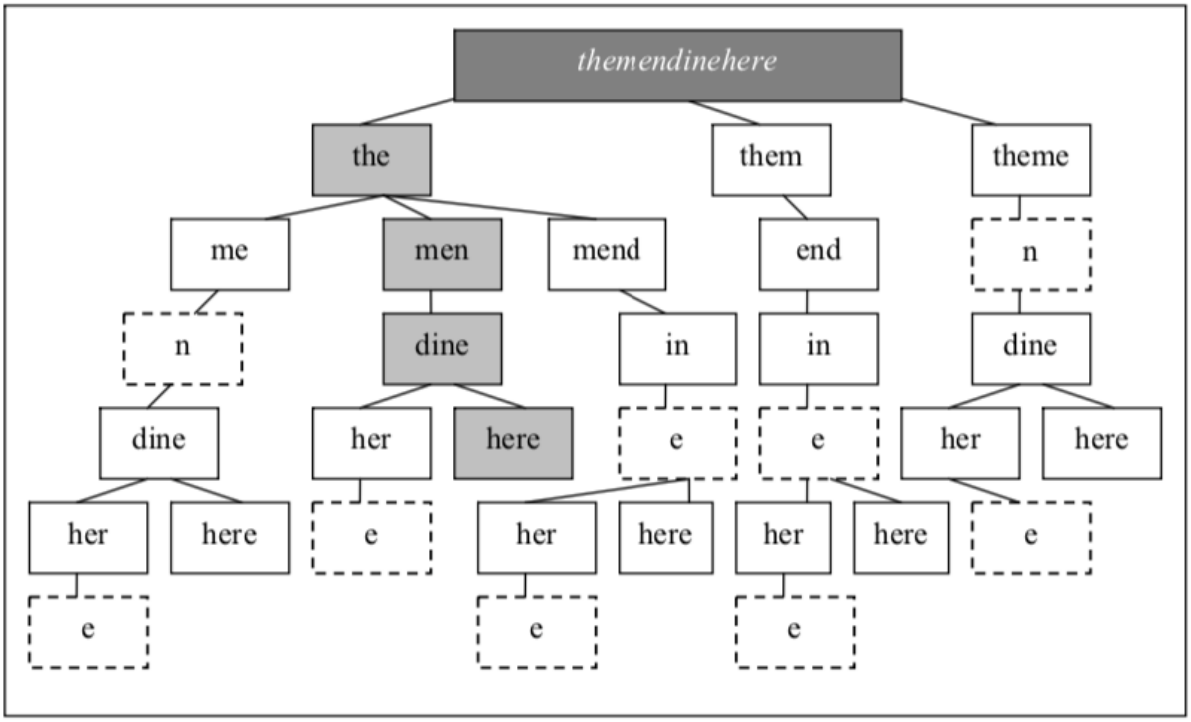
بافرض آنکه "din" یا "re" در فرهنگ لغت ما نباشد، مجموعه کلمات را به عنوان "theme n dine" به دست می‌آید.

اما همانطور که می بینید طولانی ترین کلمه می تواند تقسیم بندی نادرستی ایجاد کند. این امر منجر به گسترش بیش از حد کلمه اول "theme " به بخشی از کلمه دوم "men" و ناشناخته شدن کلمه بعدی می شود.

روش Maximum Matching

روش دیگر برای حل ماهیت حریصانه Maximal Matching الگوریتمی به نام «تطبیق حداکثر» یا Maximum Matching است.

این رویکرد چندین احتمال را تقسیم بندی می کند و گزینه ای را انتخاب می کند که کلمات کمتری در جمله داشته باشد. همچنین داشتن کلمات ناشناخته کمتر را در اولویت قرار می دهد. با استفاده از این رویکرد، تقسیم بندی صحیح را از متن مثال مانند شکل زیر بدست می آوریم.



اولین کلمه می تواند "them" ,"the", "theme" باشد. از هر یک از این گره ها، چندین انتخاب وجود دارد که همانطور که مشاهده می‌شود سایر مسیرها به تولید کلمات ناشناخته منجر شده و کارآمد نیستند.

این رویکرد از طریق تمام ترکیبات مختلف بر اساس فرهنگ لغت ما انجام می شود. بنابراین کلمات ناشناخته هنوز مشکلی است که به آن رسیدگی نکرده ایم.

علاوه بر این، این روش نیز می تواند دارای مشکل باشد اگر متن مورد بررسی تعداد کلمات بیشتری داشته باشد اما الگوریتم مسیری با تعداد کلمات کمتر پیدا کرده باشد.

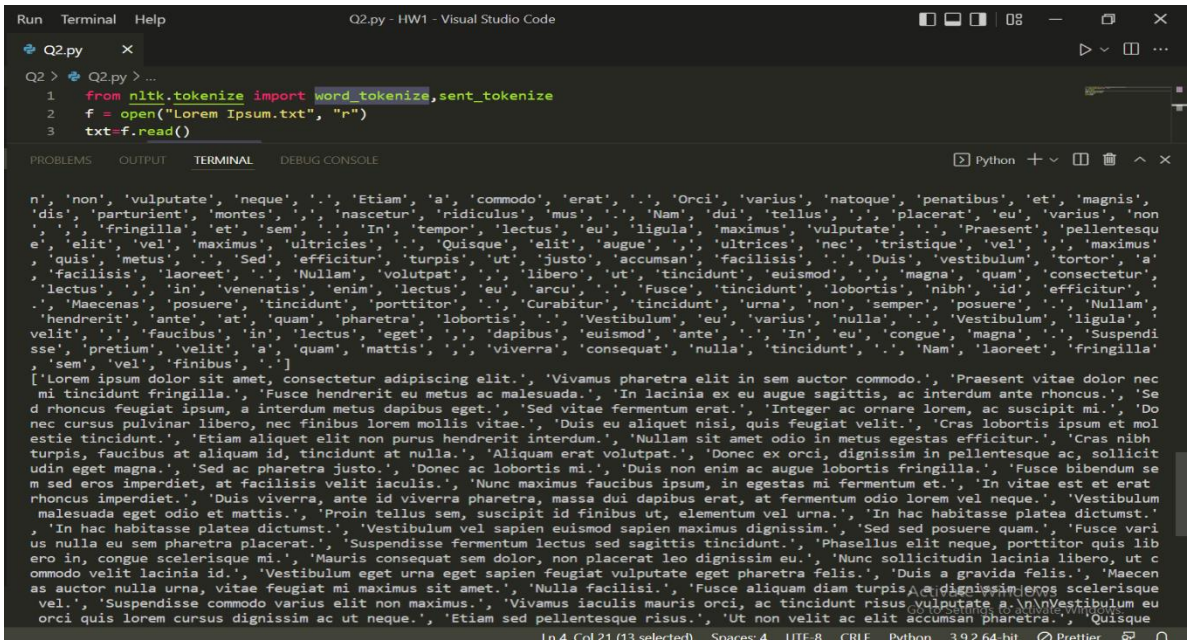
سوالات عملی

سوال ۱

کلمات مورد نظر در فایلی با نام MyFile.txt در پوشه Q1 ذخیره شده اند. پاسخ ها که همگی مثبت اند نمایش دهنده پذیرفته شدن عبارات توسط برنامه است.

سوال ۲

فایل Lorem Ipsum.txt فایلی متشکل از کلمات و جملات بی معنی و تصادفی است که قالب پاراگرافی دارد و در پوشه Q2 قرار دارد. استفاده از word_tokenize() رشته را بر اساس کلمات(نقاط پایانی و فاصله ها و..) تقسیم می‌کند اما استفاده از sent_tokenize() موجب می‌شود تا جملات متن جدا شوند. با اجرای فایل خروجی برنامه قابل مشاهده است.

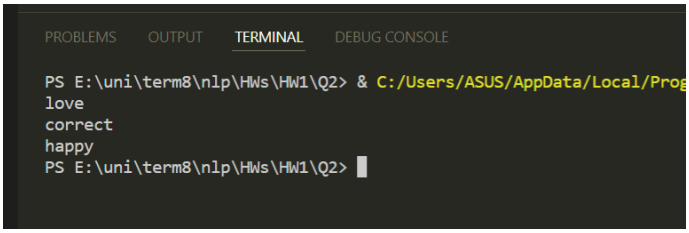


سوال ۳

کد مربوط به تمرین در پوشه Q3 قرار دارد.

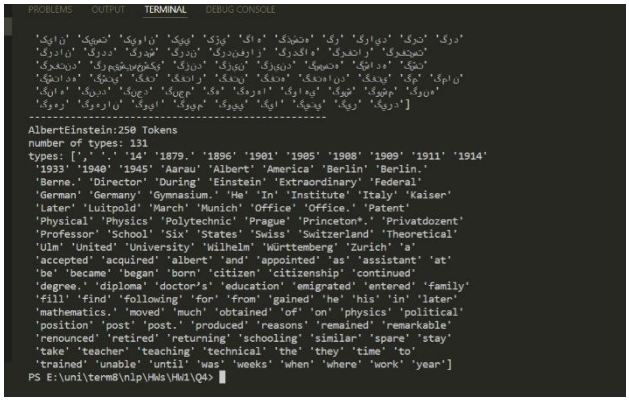
در این سوال ما لیستی از incorrect_words را تعریف می کنیم که برای آنها به املای صحیح نیاز داریم.
سپس یک حلقه برای هر کلمه در لیست کلمات نادرست اجرا می کنیم که در آن فاصله ژاکارد کلمه نادرست را محاسبه می کنیم که هر کلمه املای صحیح همان حرف اولیه را به شکلbi-gram کاراکترها داشته باشد.
سپس آنها را به ترتیب صعودی مرتب می کنیم تا کمترین فاصله در بالا باشد و کلمه مربوط به آن را استخراج کرده و چاپ می کنیم.

```
incorrect_words=['looove', 'correct', 'happpy']
```



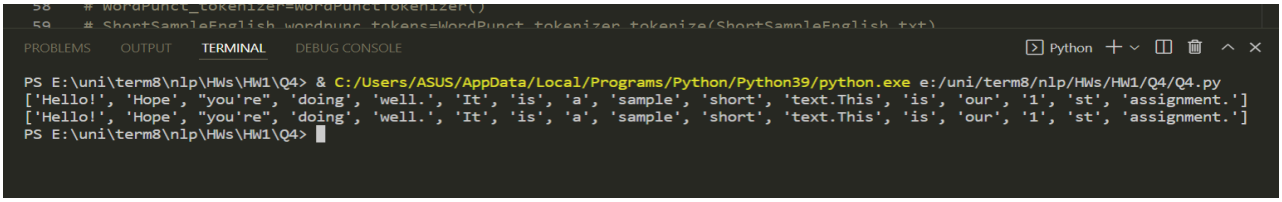
سوال ۴

نتایج موارد پ و ت در دوعکس زیر آمده است.




```
PS E:\uni\term8\nlp\HWs\HW1\Q4> & C:/Users/ASUS/AppData/Local/Programs/Python/Python39/python.exe e:/uni/term8/nlp/HWs/HW1/Q4/Q4.py
['14', '1879', '1896', '1901', '1905', '1908', '1909', '1911', '1914', '1914', '1933', '1940', '1945']
['Hello', '!', 'Hope', 'you', "''re", 'doing', 'well', '.', 'It', 'is', 'a', 'sample', 'short', 'text', '.This', 'is', 'our', '1', 'st'
, 'assignment', '.']
['الس', 'م', '!', 'امیدوارم', 'خوب', 'باشید', '.', 'این', 'یک', 'متن', 'کوتاه', 'نمونه', 'است', '.این', 'تمرین', '۱', 'ام', 'است', '.']
PS E:\uni\term8\nlp\HWs\HW1\Q4>
```

پیاده سازی WhitespaceTokenizer با استفاده از RegexpTokenizer در کد پیاده سازی شده است.و تصویر زیر نتیجه حاصل است.



این تابع برحسب علائم نگارشی رشته را توکنایز میکند و عبارت منظم معادل آن \w+|[^\w\s]+ می‌باشد،نتیجه در تصویر زیر قابل مشاهده است

```
PS E:\uni\term8\nlp\HWs\HW1\Q4> & C:/Users/ASUS/AppData/Local/Programs/Python/Python39/python.exe e:/uni/term8/nlp/HWs/HW1/Q4/Q4.py
['Hello', '!', 'Hope', 'you', "''", 're', 'doing', 'well', '.', 'It', 'is', 'a', 'sample', 'short', 'text', '.', 'This', 'is', 'our', '
1', 'st', 'assignment', '.']
PS E:\uni\term8\nlp\HWs\HW1\Q4>
```

سوال ۵

Porter stemmer

پنج مرحله کاهش کلمات در این روش استفاده می شود که هر کدام مجموعه ای از قوانین نقشه برداری خاص خود را دارد. غالباً ریشه حاصل کلمه کوتاه تری با همان ریشه است.

LancasterStemmer

لنکستر ساده است، اگرچه اغلب با ریشه گیری بیش از حد ریشه های غیر زبانی یا بی معنی را ارائه می دهد.

نتیجه اجرای کد تمرین ۵ در شکل زیر آمده است.

```
PS E:\uni\term8\nlp\HWs\HW1> cd Q5
PS E:\uni\term8\nlp\HWs\HW1\Q5> & C:/Users/ASU
Stemming using PorterStemmer
was==> wa
Germany==> germani
weeks==> week
later==> later
family==> famili
moved==> move
Italy==> itali
----------------------------
Stemming using LancasterStemmer
was==> was
Germany==> germany
weeks==> week
later==> lat
family==> famy
moved==> mov
Italy==> ita
PS E:\uni\term8\nlp\HWs\HW1\Q5>
```

لمتایزر با آرگومان‌های پیش فرض به خوبی پاسخ گو نیست چرا که تمامی واژگان را مانند یک اسم در نظر می‌گیرد و از این رو در ارائه پاسخ مناسب برای فعل‌ها یا سایر انواع کلمات بازمی‌ماند. اما با استفاده از تشخیص pos کلمه و استفاده از آن می توانیم قدرت آن را بیشتر کنیم. برای مثال در تصویر زیر پیش و پس از استفاده از pos را مشاهده می‌کنیم.

```
lemmatize with deafult values
Waves==>Waves
fishing==>fishing
rocks==>rock
was==>wa
corpora==>corpus
better==>better
ate==>ate
broken==>broken
---------------------------
lemmatize with giving apropriate POS
Waves==>Waves
fishing==>fish
rocks==>rock
was==>be
corpora==>corpora
better==>good
ate==>ate
broken==>broken
```

البته با وجود استفاده از pos کلمات باز هم می بینیم بعضی از کلمات به خوبی نتیجه نداده اند چرا که pos آن ها به گونه ای متفاوت از انتظار ما تعیین شده است.

('Waves', 'NNS'), ('fishing', 'VBG'), ('rocks', 'NNS'), ('was', 'VBD'), ('corpora', 'RB'), ('better', 'JJR'), ('ate', 'NN'), ('broken', 'NN')

برای مثال broken یک اسم تلقی می شود.

سوال ۶

به دلیل داده‌های زیاد و زمان بر بودن نمونه های موجود در پوشه و عکس ها مربوط به ۲۰۰۰۰ داده است که البته به راحتی می توان با برداشتن شرط تعداد و یا افزایش تعداد مجاز کل داده ها را بررسی کرد.

به ترتیب مراحل گفته شده در داک تمرین و به کمک عبارات منظم بخشی از پیش پردازش را مانند حذف کاراکتر های اضافی و.. صورت می پذیرد.

سپس با توکنایز کردن هر توییت اعمال سری دیگری از شرط‌های مربوط به کلمات و ریشه گیری ازآن ها توییت پاکسازی شده را از جوین کردن کلمات به‌دست آورده و در فایل final به همراه توییت اولیه می‌نویسیم.

word_tokenize رشته را بر اساس علائم نگارشی جمله تقسیم می کند.

stop words کلماتی در هر زبانی هستند که معنای زیادی به جمله اضافه نمی کنند. آنها را می توان با خیال راحت بدون قربانی کردن معنای جمله نادیده گرفت. معمولا به رایج ترین کلمات در یک زبان اشاره دارد. هیچ لیست جهانی از "کلمات توقف" وجود ندارد که توسط همه ابزارهای NLP مشترک استفاده شود.کلمات توقف اغلب قبل از آموزش مدل‌های یادگیری عمیق و یادگیری ماشین از متن حذف می‌شوند،این کلمات به تعداد زیاد در متون استفاده میشود بنابراین اطلاعات منحصربه‌فرد کمی ارائه می‌کنند که می‌تواند برای طبقه‌بندی یا خوشه‌بندی استفاده شود.

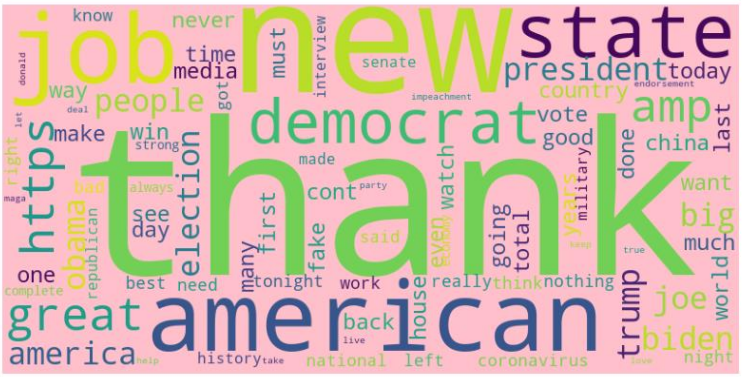
همان طور که پیشتر با روش‌های پورتر و لنکستر و تفاوت‌های آنها درسوال های قبل آشنا شدیم از آنجا که پورتر ملایم تر عمل می‌کند و محبوب تر ورایج تر است در این تمرین از پورتربرای یافتن ریشه کلمات استفاده می‌کنیم.

نمونه توییت ها در حین پردازش:

1. The threshold identification of Ballots is turning out to be even bigger than originally anticipated. A very large number of Ballots are impacted. Stay tuned!
2. The threshold identification of Ballots is turning out to be even bigger than originally anticipated. A very large number of Ballots are impacted. Stay tuned!
3. the threshold identification of ballots is turning out to be even bigger than originally anticipated. a very large number of ballots are impacted. stay tuned!

1. RT @RealRLimbaugh: Winning, winning, winning! @realDonaldTrump
2. RT : Winning, winning, winning!
3. rt : winning, winning, winning!
4. win win win

بیشترین کلمات تکرار شده در تصویر cloud به نمایش گذاشته شده اند.نتایج زیر تگ های ترند و تعداد آن هم چنین بیشترین کلمات تکرار شده به همراه تعداد می باشد.



Comon_tags:

[('maga', 241), ('timetogettough', 95), ('covid19', 67), ('trumpvlog', 63), ('kag2020', 41), ('2a', 41), ('coronavirus', 38), ('1', 31), ('trump', 24), ('paycheckprotectionprogram', 24)]

Comon_words:

[('https', 5433), ('great', 2108), ('http', 1901), ('amp', 1834), ('president', 1631), ('trump', 1555), ('people', 1172), ('thank', 1036), ('biden', 876), ('new', 857), ('obama', 810), ('get', 744), ('news', 725), ('would', 703), ('america', 695), ('big', 681), ('like', 670), ('joe', 662), ('election', 650), ('country', 647), ('democrats', 621), ('today', 619), ('never', 614), ('vote', 600), ('one', 591), ('american', 575), ('china', 556), ('time', 543),

('many', 535), ('going', 520), ('good', 516), ('fake', 494), ('must', 483), ('want', 480), ('make', 477), ('even', 469), ('years', 454), ('job', 452), ('thanks', 452), ('see', 444), ('state', 444), ('back', 438), ('cont', 432), ('states', 416), ('win', 414), ('house', 409), ('media', 399), ('last', 397), ('jobs', 390), ('way', 389), ('total', 388), ('much', 383), ('first', 355), ('watch', 353), ('world', 346), ('done', 343), ('day', 340), ('ever', 331), ('nothing', 329), ('national', 329), ('know', 328), ('best', 328), ('need', 327), ('coronavirus', 326), ('really', 325), ('night', 323), ('work', 322), ('said', 318), ('tonight', 310), ('got', 310), ('right', 308), ('bad', 308), ('think', 307), ('left', 306), ('history', 297), ('republican', 295), ('made', 292), ('interview', 284), ('senate', 283), ('military', 283), ('complete', 283), ('strong', 282), ('always', 281), ('economy', 280), ('also', 279), ('impeachment', 273), ('donald', 271), ('endorsement', 271), ('americans', 269), ('let', 266), ('love', 263), ('maga', 263), ('democrat', 262), ('take', 262), ('deal', 262), ('keep', 259), ('live', 257), ('help', 257), ('true', 254), ('party', 253)]