امیرعلی بلباسی ۹۸۳۱۱۰۹

In [5]:

```
%pip install hazm

from __future__ import unicode_literals
from hazm import *
import codecs
import json
import collections

print('#part 1')
f = open('foo.json')
#f = open('IR_data_news_12k.json')
data = json.load(f)
f.close()

contents = []
for i in data:
    tmp = data[i]['content']
    contents.append(tmp[0:len(tmp)-16])
print('#1')
for i in contents:
    print(i)
print()

normalizer = Normalizer()
for i in range(len(contents)):
    contents[i] = normalizer.normalize(contents[i])
#print('#2')
#for i in contents:
#    print(i)

tokens = []
for i in range(len(contents)):
    tokens.append(word_tokenize(contents[i]))
#print('#3')
#for i in tokens:
#    print(i)
#    print()

lemmatizer = Lemmatizer()
for i in range(len(tokens)):
    for j in range(len(tokens[i])):
```

```
        tokens[i][j] = lemmatizer.lemmatize(tokens[i][j])
#print('#4')
#for i in tokens:
#    print(i)
#    print()

"""
stemmer = Stemmer()
for i in range(len(tokens)):
    for j in range(len(tokens[i])):
        tokens[i][j] = stemmer.stem(tokens[i][j])
print('#5')
for i in tokens:
    print(i)
    print()
"""

f = codecs.open('stopwords.dat', encoding='utf-8')
stopWords = []
for l in f.readlines():
    stopWords.append(l.strip('\n'))
f.close()
#print('A#')
#print(stopWords)
#print()

for i in range(len(stopWords)):
    stopWords[i] = normalizer.normalize(stopWords[i])
#print('B#')
#print(stopWords)
#print()

for i in range(len(stopWords)):
    stopWords[i] = lemmatizer.lemmatize(stopWords[i])
#print('C#')
#print(stopWords)
#print()

tmpTokens = []
for i in range(len(tokens)):
    tmpTokensDoc = []
    for j in range(len(tokens[i])):
        if(tokens[i][j] not in stopWords): tmpTokensDoc.append(tokens[i][j])
    tmpTokens.append(tmpTokensDoc)
tokens = tmpTokens.copy()
```

```
print('#5')
for i in tokens:
    for j in i:
        print(j)
    print('\n+++++\n')
print()

print('#part 2')
#uniqueTokens = set()
#for i in tokens:
#    for j in i:
#        uniqueTokens.add(j)
#sortedTokens = []
#sortedTokens = sorted(list(uniqueTokens))
#print('#6')
#for i in sortedTokens:
#    print(i)
#print()

positionalIndex = {}
for j in range(len(tokens)):
    for k in range(len(tokens[j])):
        token = tokens[j][k]
        if(token in positionalIndex):
            oldDocFrequency = positionalIndex[token][0]
            oldFrequencyInDoc = positionalIndex[token][1].copy()
            oldPositionsInDoc = positionalIndex[token][2].copy()
            if(j in oldFrequencyInDoc):
                oldFrequencyInDoc[j] += 1
                oldPositionsInDoc[j].append(k)
                positionalIndex[token] = (oldDocFrequency,
oldFrequencyInDoc.copy(), oldPositionsInDoc.copy())
            else:
                a = oldFrequencyInDoc.copy()
                b = oldPositionsInDoc.copy()
                a[j] = 1
                b[j] = [k]
                positionalIndex[token] = (oldDocFrequency+1, a.copy(),
b.copy())
        else:
                a = dict()
                b = dict()
                a[j] = 1
                b[j] = [k]
                positionalIndex[token] = (1, a.copy(), b.copy())
```

```
positionalIndex =
collections.OrderedDict(sorted(positionalIndex.items())).copy()
print('#7')
for i in positionalIndex:
    print(f'{i}:')
    print(positionalIndex[i])
    print()
Requirement already satisfied: hazm in d:\applications\anaconda\lib\site-
packages (0.7.0)
Requirement already satisfied: nltk==3.3 in
d:\applications\anaconda\lib\site-packages (from hazm) (3.3)
Requirement already satisfied: six in d:\applications\anaconda\lib\site-
packages (from nltk==3.3->hazm) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
#part 1
#1
```

در نامه ای رسمی به (AFC) به گزارش خبرگزاری فارس، کنفدراسیون فوتبال آسیا فدراسیون فوتبال ایران و باشگاه گیتی پسند زمان قرعه کشی جام باشگاه های فوتسال آسیا را رسماً اعلام کرد. بر این اساس ۲۵ فروردین ماه ۱۴۰۱ مراسم قرعه کشی جام باشگاه های فوتسال آسیا در مالزی برگزار می شود. باشگاه گیتی پسند بعنوان قهرمان فوتسال ایران در سال ۱۴۰۰ به این مسابقات راه پیدا کرده است. پیش از این گیتی پسند تجربه ۳ دوره حضور در جام باشگاه های فوتسال آسیا را داشته که هر سه دوره به فینال مسابقات راه پیدا کرده و یک عنوان قهرمانی و دو مقام دومی بدست آورده است.

به گزارش خبرگزاری فارس، سید حمید سجادی در حاشیه مراسم گرامیداشت روز جوان در جمع خبرنگاران در رابطه با عرضه سهام سرخابی‌ها در بورس اظهار داشت: منتظر طی روند هستیم و بعدا اطلاع رسانی خواهیم کرد. وی در مورد حضور تماشاگران در مسابقات فوتبال اظهار داشت: حضور تماشاگران در لیگ برتر فوتبال تابع نظر فدراسیون فوتبال، سازمان لیگ و ستاد ملی مبارزه با کرونا است.

```
#5
گزارش
خبرگزاری
فارس
کنفدراسیون
فوتبال
آسیا
AFC
نامه
رسم
فدراسیون
فوتبال
ایران
```

باشگاه
گیتی
پسند
زمان
قرعه
کشید#کش
جام
باشگاه
فوتسال
آسیا
رسما
اعلام
اساس
۲۵
فروردین
ماه
۱۴۰۱
مراسم
قرعه
کشید#کش
جام
باشگاه
فوتسال
آسیا
مالزی
برگزار
باشگاه
گیتی
پسند
بعنوان
قهرمان
فوتسال
ایران
۱۴۰۰
مسابقات
گیتی
پسند
تجربه
۳
دوره
حضور
جام
باشگاه
فوتسال
آسیا

دوره
فینال
مسابقات
عنوان
قهرمان
مقام
بدست

+++++

گزارش
خبرگزاری
فارس
سید
حمید
سجاد
حاشیه
مراسم
گرامیداشت
روز
جوان
خبرنگار
رابطه
عرضه
سهام
سرخابی‌ها
بورس
اظهار
منتظر
بعدا
اطلاع
رساند#رسان
حضور
تماشاگر
مسابقات
فوتبال
اظهار
حضور
تماشاگر
لیگ
برتر
فوتبال
تابع
فدراسیون
سازمان

لیگ
ستاد
ملی
مبارزه
کرونا

+++++

#part 2
#7
AFC:
(1, {0: 1}, {0: [6]})

آسیا:
(1, {0: 4}, {0: [5, 21, 35, 56]})

اساس:
(1, {0: 1}, {0: [24]})

اطلاع:
(1, {1: 1}, {1: [20]})

اظهار:
(1, {1: 2}, {1: [17, 26]})

اعلام:
(1, {0: 1}, {0: [23]})

ایران:
(1, {0: 2}, {0: [11, 44]})

باشگاه:
(1, {0: 5}, {0: [12, 19, 33, 38, 54]})

بدست:
(1, {0: 1}, {0: [63]})

برتر:
(1, {1: 1}, {1: [30]})

برگزار:
(1, {0: 1}, {0: [37]})

بعدا:

(1, {1: 1}, {1: [19]})

بعنوان:
(1, {0: 1}, {0: [41]})

بورس:
(1, {1: 1}, {1: [16]})

تابع:
(1, {1: 1}, {1: [32]})

تجربه:
(1, {0: 1}, {0: [49]})

تماشاگر:
(1, {1: 2}, {1: [23, 28]})

جام:
(1, {0: 3}, {0: [18, 32, 53]})

جوان:
(1, {1: 1}, {1: [10]})

حاشیه:
(1, {1: 1}, {1: [6]})

حضور:
(2, {0: 1, 1: 2}, {0: [52], 1: [22, 27]})

حمید:
(1, {1: 1}, {1: [4]})

خبرنگار:
(1, {1: 1}, {1: [11]})

خبرگزاری:
(2, {0: 1, 1: 1}, {0: [1], 1: [1]})

دوره:
(1, {0: 2}, {0: [51, 57]})

رابطه:
(1, {1: 1}, {1: [12]})

رساند#رسان:

(1, {1: 1}, {1: [21]})

رسم:
(1, {0: 1}, {0: [8]})

رسما:
(1, {0: 1}, {0: [22]})

روز:
(1, {1: 1}, {1: [9]})

زمان:
(1, {0: 1}, {0: [15]})

سازمان:
(1, {1: 1}, {1: [34]})

ستاد:
(1, {1: 1}, {1: [36]})

سجاد:
(1, {1: 1}, {1: [5]})

سرخابی‌ها:
(1, {1: 1}, {1: [15]})

سهام:
(1, {1: 1}, {1: [14]})

سید:
(1, {1: 1}, {1: [3]})

عرضه:
(1, {1: 1}, {1: [13]})

عنوان:
(1, {0: 1}, {0: [60]})

فارس:
(2, {0: 1, 1: 1}, {0: [2], 1: [2]})

فدراسیون:
(2, {0: 1, 1: 1}, {0: [9], 1: [33]})

فروردین:

(1, {0: 1}, {0: [26]})

فوتبال:
(2, {0: 2, 1: 2}, {0: [4, 10], 1: [25, 31]})

فوتسال:
(1, {0: 4}, {0: [20, 34, 43, 55]})

فینال:
(1, {0: 1}, {0: [58]})

قرعه:
(1, {0: 2}, {0: [16, 30]})

قهرمان:
(1, {0: 2}, {0: [42, 61]})

لیگ:
(1, {1: 2}, {1: [29, 35]})

مالزی:
(1, {0: 1}, {0: [36]})

ماه:
(1, {0: 1}, {0: [27]})

مبارزه:
(1, {1: 1}, {1: [38]})

مراسم:
(2, {0: 1, 1: 1}, {0: [29], 1: [7]})

مسابقات:
(2, {0: 2, 1: 1}, {0: [46, 59], 1: [24]})

مقام:
(1, {0: 1}, {0: [62]})

ملی:
(1, {1: 1}, {1: [37]})

منتظر:
(1, {1: 1}, {1: [18]})

نامه:

(1, {0: 1}, {0: [7]})

پسند:
(1, {0: 3}, {0: [14, 40, 48]})

کرونا:
(1, {1: 1}, {1: [39]})

کشید#کش:
(1, {0: 2}, {0: [17, 31]})

کنفدراسیون:
(1, {0: 1}, {0: [3]})

گرامیداشت:
(1, {1: 1}, {1: [8]})

گزارش:
(2, {0: 1, 1: 1}, {0: [0], 1: [0]})

گیتی:
(1, {0: 3}, {0: [13, 39, 47]})

۱۴۰۰:
(1, {0: 1}, {0: [45]})

۱۴۰۱:
(1, {0: 1}, {0: [28]})

۲۵:
(1, {0: 1}, {0: [25]})

۳:
(1, {0: 1}, {0: [50]})

ساختمان داده استفاده شده بدین صورت است:

```
{
      string: (int, {string: int}, {string: []})
}
```

شاخص مکانی یک دیکشنری است که هر term در آن key و value آن یک tuple سه‌تایی است. عضو اول tuple تعداد مستنداتی است که term ما حداقل یکبار در آنها آمده (doc frequency). عضو دوم آن یک دیکشنری است بدین صورت که key آن id مستند و value آن تعداد تکرار term ما در آن مستند است. سومین عضو tuple یک دیکشنری دیگر است بدین صورت که key آن id مستند و value آن یک لیست است که در آن به ترتیب از اول به آخر position های حضور term ما در آن مستند ذخیره شده است.

طی دو حلقه (یکی روی مستندات و دیگری درون هر مستند) کلمه به کلمه پیش می‌رویم و با توجه به اینکه کلمه اولین بار است که به شاخص مکانی اضافه می‌شود یا اولین بار است که درون یک مستند ظاهر می‌شود یا هیچ‌کدام، با توجه به ساختمان داده ذکر شده، به شاخص مکانی کلمات جدید اضافه و یا جزئیات کلمات قدیمی را آپدیت می‌کنیم. سر آخر نیز دیکشنری را به ترتیب الفبا sort می‌کنیم. بدین ترتیب شاخص مکانی ما ساخته می‌شود. در مثال بالا شاخص مکانی ایجاد شده برای دو خبر اول دیتاست (موجود در foo.json) را مشاهده می‌کنید.

در ادامه نیز خروجی شاخص مکانی (شاخص مکانی همه دوازده هزار خبر) را به ازای کلمه ‘گیتی’ مشاهده می‌کنید:

```
print('#7')
print(positionalIndex['گیتی'])
```

```
#part 1
#part 2
#7
(48, {0: 3, 258: 1, 261: 2, 364: 2, 669: 2, 817: 3, 981: 2, 1486: 1, 1787: 1, 1800: 1, 1885: 2, 1889: 2, 1891: 1, 2033: 3, 2320: 2, 2398: 1, 2401: 1
, 2478: 1, 2545: 3, 2583: 5, 2607: 1, 2621: 1, 2641: 1, 2932: 2, 3099: 1, 3442: 3, 3585: 4, 4050: 4, 4476: 3, 4567: 1, 4855: 2, 4905: 3, 5033: 2, 53
12: 3, 5347: 2, 5480: 3, 5707: 3, 5708: 2, 5711: 2, 5786: 4, 6082: 2, 6149: 1, 6182: 3, 6650: 1, 6723: 4, 6874: 1, 6938: 1, 7791: 1}, {0: [13, 36, 4
4], 258: [11], 261: [18, 98], 364: [80, 134], 669: [22, 72], 817: [125, 127, 163], 981: [17, 47], 1486: [19], 1787: [39], 1800: [158], 1885: [7, 32]
, 1889: [16, 24], 1891: [18], 2033: [10, 45, 79], 2320: [6, 37], 2398: [77], 2401: [8], 2478: [9], 2545: [28, 63, 77], 2583: [13, 33, 55, 64, 110],
2607: [24], 2621: [27], 2641: [252], 2932: [21, 53], 3099: [108], 3442: [20, 27, 38], 3585: [26, 32, 57, 132], 4050: [21, 28, 46, 94], 4476: [21, 41
, 67], 4567: [20], 4855: [248, 327], 4905: [20, 32, 108], 5033: [23, 52], 5312: [123, 138, 159], 5347: [24, 85], 5480: [22, 27, 103], 5707: [14, 59,
 80], 5708: [11, 71], 5711: [43, 64], 5786: [86, 103, 112, 141], 6082: [30, 37], 6149: [130], 6182: [25, 38, 79], 6650: [40], 6723: [15, 39, 56, 88]
, 6874: [368], 6938: [53], 7791: [221]})
```