

به نام خدا

گزارش فاز اول پروژه بازیابی اطلاعات

محمدرضا قادری ۹۶۲۷۰۵۷

در این پروژه از کتابخانه‌های numpy (برای برخی محاسبات احتمالی مورد نیاز ماتریسی) و pandas (برای ارتباط با داده‌ها و خواندن و نوشتن در فایل‌ها) و همچنین از کتابخانه hazm برای برخی از قسمت‌های پیش پردازشی ابتدایی برای اسناد استفاده شده است.

```
from __future__ import unicode_literals
import pandas as pd
import numpy as np
from hazm import *
import collections
import string
```

۱-۱ پیش پردازش اسناد

در این قسمت ابتدا بایستی از فایل دیتاهای مورد نظر را بخونیم. برای این کار روش‌های زیادی مانند openpyxl و pandas و تعدادی دیگر از کتابخانه‌ها موجود هست که برای این پروژه از pandas استفاده کردیم. با توجه به این که از google colab استفاده کردم برای اینکه بتوانم از فایل بخونم از google drive استفاده کردم تا به فایل دسترسی داشته باشم بایستی یک دسترسی به colab با آیدی فایل بدهم تا استفاده کند از آن.

```
[3] !pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)

file_id = '1jcbbbPTNnQ3NKrPJL_oc9vrFvHH4UuR'
downloaded = drive.CreateFile({'id': file_id})
```

سپس با دانلود فایل می‌تونیم ازش استفاده کنیم.

```
[4] downloaded.GetContentFile('/IR1_7k_news.xlsx')
```

با استفاده از کتابخانه pandas شروع به خواندن فایل می‌کنیم چون در ادامه بیشتر با ستون content کار داریم فقط فعلا از این ستون و urls اون‌ها استفاده می‌کنیم.

```
# read data
df = pd.read_excel("/IR1_7k_news.xlsx")
urls = df['url']
content = df['content']
```

در ادامه ابتدا از normalizer کتابخانه hazm استفاده می‌کنیم که در ابتدا کمی content ها رو نرمال کنیم چرا که برخی نگارش‌های اشتباه مانند شماره‌ها که به زبان فارسی نیستند و برخی نیم فاصله و برخی از تنوین‌های زبان عربی و تعدادی دیگر از اشتباهات رایج را با استفاده از این نرمالایزر تغییر می‌دهد.

```
# normalize data by use hazm
normalizer = Normalizer()
print(content[11])
for i in range(len(content)):
    content[i] = normalizer.normalize(content[i])
copy_content = content
print(content[11])
```

نمونه‌ایی از تغییر در content را برای خبر ۹ به صورت زیر است.

به گزارش خبرگزاری فارس از ****اصفهان، **** دیدار تیم‌های ذوب‌آهن و استقلال با ۲ گل
به (قرهاد مجیدی) <https://www.farsnews.ir/special>) به سود شاگردان [قرهاد مجیدی]
پایان رسید.

رشدید <https://search.farsnews.ir/?q=> بعد از این مسابقه [رشدید مظاهری]
در رختکن ذوب‌آهن حاضر شد و با بازیکنان تیم میزبان خوش ویش کرد (o=on&مظاهری
مظاهری بعد از خروج از رختکن به درخواست خبرنگاران برای مصاحبه پاسخ منفی داد و
گفت آفتدر مشکل روحی و روانی دارم که اصلاً نمی‌توانم مصاحبه کنم.

به گزارش فارس، **ظاهراً** طی روزهای گذشته مشکلاتی برای دروازه‌بان استقلال در
به <https://search.farsnews.ir/?q=> تمرینات این تیم به وجود آمده که وی [به صورت]
سریسته مقابل خبرنگاران به این موضوع اشاره کرده است (o=on&صورت

/انتهای پیام

به گزارش خبرگزاری فارس از ****اصفهان، **** دیدار تیم‌های ذوب‌آهن و استقلال با ۲ گل
به (قرهاد مجیدی) <https://www.farsnews.ir/special>) به سود شاگردان [قرهاد مجیدی]
پایان رسید.

رشدید <https://search.farsnews.ir/?q=> بعد از این مسابقه [رشدید مظاهری]
در رختکن ذوب‌آهن حاضر شد و با بازیکنان تیم میزبان خوش ویش کرد (o=on&مظاهری
مظاهری بعد از خروج از رختکن به درخواست خبرنگاران برای مصاحبه پاسخ منفی داد و
گفت آفتدر مشکل روحی و روانی دارم که اصلاً نمی‌توانم مصاحبه کنم.

به گزارش فارس، **ظاهراً** طی روزهای گذشته مشکلاتی برای دروازه‌بان استقلال در
به <https://search.farsnews.ir/?q=> تمرینات این تیم به وجود آمده که وی [به صورت]
سریسته مقابل خبرنگاران به این موضوع اشاره کرده است (o=on&صورت

/انتهای پیام

ولی این نگارش‌ها برای ما کافی نیست هنوز وجود کلمات اینگلیسی و نشانه گذاری‌هایی مانند پرانتز و کروشه و غیره و حتی اعداد و تمامی علائم نگارشی (که با توجه به اینکه از normalizer استفاده کردیم خیلی از علائم به فرسی تبدیل شده اند مانند علامت سوال و ویرگول) پس این‌ها نیز بایستی از متن ما خارج شوند.

```
[6] punctuation = [ '+','=','.', ',', '*', '^','%', '?','[',']','(',')','{','}','<','>', '&c', '\n','\t','\r','\"', '\"\", \"'\"]  
signs = [ '!','@','#','$','%','^','&', '*','-','\'','\\','/', '\\\'','|', '"', '-' , "-" ]  
numbers = [ '0','1','2','3','4','5','6','7','8','9' ]  
signs_ir = [ '~','_','-','.' ]  
img_src = ['UFIINPF']  
  
# some_unknown_char = ['\u200c', '\u200d', '\u200e', '\u200f']  
english = list(string.ascii_lowercase) + list(string.ascii_uppercase)  
not_used = punctuation + signs + numbers + signs_ir + img_src + english
```

برای کلمات و حروف خارجی از کتابخانه string استفاده کردیم و تمامی حروف را به صورت بزرگ و کوچک چک کردیم و تنها حالت متفاوت UFITNPF بود که normalizer در تبدیل آدرس عکس به جا می‌گذاشت.

حالا رو تمامی content ها حرکت می‌کنیم و به جا موارد بالا "" جاگذاری می‌کنیم. نکته‌ای که جا موند انتهای پیام بود که در شکل بالا نیومده است ولی در کد اصلی استفاده شده چرا که بنظر میرسید که فقط برای اطلاع از پایان متن بوده.

```
for i in range(len(content)):
    for l in not_used:
        content[i] = content[i].replace(l, "")
```

حالا با استفاده از نگارش بالا متن قبلی به صورت زیر در آمد.(که تمامی موارد مورد نظر حذف شد)

به گزارش خبرنگاری فارس از اصفهان دیدار تیم‌های نوب‌آهن و استقلال با گل
به سود شاگردان قره‌دا مجیدی قره‌دا مجیدی به
پایان رسید

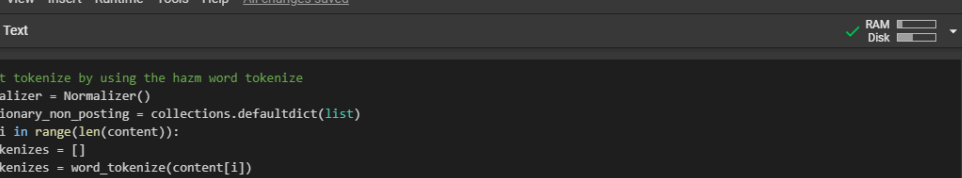
بعد از این مسابقه رشید مظاهری
مظاهری در رختکن نوب‌آهن حاضر شد و با بازیکنان تیم میزبان خوش و بش کرد
مظاهری بعد از خروج از رختکن به درخواست خبرنگاران برای مصاحبه پاسخ منفی داد و
گفت: آنقدر مشکل روحی و روانی دارم که اصلاً نمی‌توانم مصاحبه کنم

به گزارش فارس ظاهراً طی روزهای گذشته مشکلاتی برای دروازه‌بان استقلال در
تمرینات این تیم به وجود آمده که وی به صورت
صورت سر بسته مقابل خبرنگاران به این موضوع اشاره کرده است

انتهای پیام

پس تمامی متن‌ها برای تحلیل بیشتر آمده شده است.

برای جذب کلمات پرتکرار (stop word) ها ابتدا بایستی تو کل متن ها فرکانس هر کلمه را بدست آوریم تا بتوانیم در مورد کلماتی پرتکرار مثل و، در، به، از، این و ... تصمیم بگیریم. برای این منظور از کتابخانه collections برای شمردن تکرار در هر مت به این صورت که وقتی `word_tokenize` انجام می‌دهیم یک لیست از کلمات به ما می‌دهد میتوانیم تکرار را با کمک تابع `counter` از کتابخانه `collections` بدست آوریم سپس در همه متن ها میتوانیم تعداد تکرار را به همین صورت بدست آورد.



```
IR_phase1.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[183] # get tokenize by using the hazm word tokenize
normalizer = Normalizer()
dictionary_non_posting = collections.defaultdict(list)
for i in range(len(content)):
    tokenizes = []
    tokenizes = word_tokenize(content[i])
    tokenizes_count = dict(collections.Counter(tokenizes))
    for key, value in tokenizes_count.items():
        if key in dictionary_non_posting:
            value_old = dictionary_non_posting[key]
            dictionary_non_posting[key] = value + value_old
        else:
            dictionary_non_posting[key] = value

[184] print((dictionary_non_posting['و']))

130261

length_dict_non_posting = {k: v for k, v in sorted(dictionary_non_posting.items(), key=lambda item: item[1], reverse=True)}
with open("mydict.txt", 'w') as f:
    for key, value in length_dict_non_posting.items():
        f.write('%s:%s\n' % (key, value))

[160] # get tokenize by using the hazm word tokenize (the postings list [docid, freq, , [position]])
```

mydict.txt

```
1 130261: و
2 95672: در
3 75209: به
4 56249: از
5 42805: این
6 39419: که
7 38292: با
8 33237: را
9 21717: است
10 15537: برای
11 11548: کرد
12 10460: بر
13 10025: اورا
14 9602: یک
15 9601: کشور
16 9467: آن
17 9412: تا
18 8845: شد
19 8366: هم
20 8223: خود
21 8049: تم
22 8024: دلم
23 7943: گرفتار
24 7563: الفبا
25 7101: ما
```

برای مثال و به تعداد ۱۳۰۲۶۱ بار تکرار شده در این حالت برای اطمینان ۱۰۰ تا کلمه اول را چک می‌کنیم و کلماتی Stop word هستند را حذف می‌کنیم. (فایل mydict تکرارها را دارد) حالا یک آرایه تهیه می‌کنیم از stop word ها تا برای لیست کردن از آن‌ها استفاده نکنیم. برای اینکه کلمات بالا فقط در متون به کثرت استفاده می‌شوند می‌توانیم آن‌ها را از درون dictionary خود کنار بگذاریم (چرا که در کوئری های سرچ خیلی استفاده نمی‌شوند و چون در اکثر متون هستند فرقی برای ما نمی‌کند چرا که مرجع میشوند)

[282] stop_words = ['و', 'در', 'به', 'از', 'این', 'که', 'با', 'را', 'برای', 'بر', 'آن', 'تا', 'هم', 'خود', 'ما', 'وی', 'نیز', 'یا', 'او', 'هر', 'پس', 'پیش', 'انها']

این یک روش برای stop word هاست روش دیگر استفاده از کتابخانه hazm هست که خوش یک لیستی از stop word ها تهیه کرده که ما با توجه با اینها و اینکه ریشه یابی کلمات در کل کانتنت رو تغییر میدهیم و از همین طریق dictionary مورد نظر خود را می‌سازیم . برای ریشه یابی از stem که در کتابخانه hazm موجود است استفاده می‌کنیم.

بعد از در نظر نگرفتن این کلمات می‌توانیم با یافتن مکان هر کلمه در متن که موجود است نسبت به کلمه ابتدایی آنها رو به عنوان لیستی برای هر کلمه ذخیره کرد. از انجایی که برای ترتیب اهمیت اینکه یک کلمه در یک متن چندبار تکرار شده از یک لیست سه تایی برای هر متن استفاده می‌شود، پس هر key که کلمه مورد نظر ماست دارای value لیست سه تایی به ازای هر متنی که در آن موجود باشد دارد که اولین عضو لیست شماره آن متن و دومین تعداد تکرار آن کلمه در متن و سومین عضو لیست position های آن کلمه در آن متن نسبت به ابتدای آن متن هست (کلمه به کلمه).
برای بدست آوردن جایگاه یک کلمه در متن از قطعه کد زیر استفاده شده

```
# to find position of each word in contents
def position_find(word_to_find , contents):
    words = contents.split()
    return [pos for pos, word in enumerate(words, start=0) if word == word_to_find]
```

که متن را می‌گیرد و به نسبت به آن کلمه می‌شمارد در کدام موقعیت‌ها استفاده شده و لیستی برمی‌گرداند. برای مثال کلمه یحیی در متن ۱ (که بهتر هست متن رو به صورت تکه تکه شده کلمات بفرستیم) (که در اصل دومین متن اطلاعات ماهست) در یک جا و position ۱۲۷ از ابتدا قرار گرفته است.

```
[281] print(position_find('یحیی' , content[1]))
[127]
```

در این حالت پس برای ساخت شاخص مکانی اقدام می‌کنیم با این تفاسیر این شاخص را در دیکشنری نگه می‌داریم تا نتوانیم به راحتی و با سرعت به آن دسترسی داشته باشیم در ابتدا یکی یکی متن‌ها رو می‌خوانیم و کلمات آن را بدست می‌آوریم و کلمه به کلمه جلو می‌رویم و با تابع بالا لیست مکان‌های آن را می‌ایم. الان هر سه فاکتور مورد نظر را داریم پس می‌توانیم شاخص مکانی بسازیم.

```

# get tokenize by using the hazm word tokenize (the postings list [docid , freq , [position]])
normalizer = Normalizer()
dictionary = collections.defaultdict(list)
for i in range(len(content)):
    tokenizes = []
    tokenizes = word_tokenize(content[i])
    tokenizes_count = dict(collections.Counter(tokenizes))
    for key , value in tokenizes_count.items():
        if key not in stop_words:
            index = []
            pos_index = position_find(key , content[i])
            index.append(i)
            index.append(value)
            index.append(pos_index)
            dictionary[key].append(index)

284] # length_dict = {key: len(value) for key, value in dictionary.items()}
# length_dict = {k: v for k, v in sorted(dictionary.items(), key=lambda item: item[1] ,reverse=False)}
# print(len(length_dict))
with open("posting-list.txt", 'w') as f:
    for key, value in dictionary.items():
        f.write('%s:%s\n' % (key, value))

```

برای مثال کلمه یحیی رو شاخص مکانی برایش بدست آوردیم.

```

[285] print(dictionary['یحیی'])

[[1, 1, [127]], [47, 2, [160, 163]], [49, 1, [377]], [84, 2, [60, 63]], [128, 1, [50]], [132, 2, [57, 60]], [177, 2, [484, 487]], [223, 1, [6]], [226, 2, [19,

```

که میبینم در متن اول در مکان ۱۲۷ و فقط یک بار و در متن ۴۷ در ۲ جا آمده است.

برای اینکه بتوانیم کمی منطقی‌تر عمل بهتر است نسبت به تعداد تکرار کلمه در متن آن کلمه رو با ترتیب برای کاربر نشان دهد چرا که درصد بیشتری برای اینکه درست باشد دارد. برای این کار یک تابع به صورت زیر داریم که دیکشنری ما رو بر اساس مقدار دو که فرکانس در هر متن هست مرتب می‌کند.

```

[288] # the more priority
def sort_dict(dictionary):
    dicts = dictionary
    sorted_dicts = sorted(dicts, key = operator.itemgetter(1, 2) , reverse=True)
    return sorted_dicts

```

حالا آماده برای بدست آوردن جواب‌های پرسمان هستیم برای پرسمان تک کلمه‌ای تنها کافی هست ما از کاربر یک ورودی میگیریم ابتدا بایستی این کلمه رو نرمال کنیم و سپس در داشته‌های خودمان دنبال کلمه مشابه آن باشیم. سپس در دیکشنری مربوطه به دنبال کلمه نرمال شده می‌گردیم تا اینکه مکان‌های مربوط به آن را بیابیم.

```
# one word query
# this sorted query question by frequency
def query_one_word():
    query = input("enter a word for checking: ")
    time_start = datetime.datetime.now()
    normal_query = lstemmer.stem(query)
    print("Normal word to search {}".format(normal_query))
    # all ,sorted_dict = sort_dict(dictionary[normal_query])
    dicts = dictionary[normal_query]
    sorted_dict = sorted(dicts, key = operator.itemgetter(1, 2) , reverse=True)
    print(sorted_dict)
    time_finish = datetime.datetime.now()
    print("{} results in {} ms".format(len(sorted_dict), ((time_finish - time_start).total_seconds())*1000))
    print("id -> title\n")
    for i in sorted_dict:
        print("{} -> {}".format(i[0] , title[i[0]]))
    print(sorted_dict)
query_one_word()
```

در حالت بالا سه تایی که در دیکشنری داشتیم را بر حسب دومی که تعداد تکرار در اون متن هست مرتب میکنیم برای خروجی بین الملل این کوئری به صورت زیر است

```
enter a word for checking: بین الملل
Normal word to search بین الملل
[[4767, 17, [10, 20, 55, 81, 96, 165, 211, 220, 240, 243, 257, 374, 393, 421, 496, 556, 635]], [2785, 15, [357, 432, 783, 1104, 1112, 1436, 1481, 1606, 1621, 1635, 1640, 1645, 1650, 1655, 1660, 1665, 1670, 1675, 1680, 1685, 1690, 1695, 1700, 1705, 1710, 1715, 1720, 1725, 1730, 1735, 1740, 1745, 1750, 1755, 1760, 1765, 1770, 1775, 1780, 1785, 1790, 1795, 1800, 1805, 1810, 1815, 1820, 1825, 1830, 1835, 1840, 1845, 1850, 1855, 1860, 1865, 1870, 1875, 1880, 1885, 1890, 1895, 1900, 1905, 1910, 1915, 1920, 1925, 1930, 1935, 1940, 1945, 1950, 1955, 1960, 1965, 1970, 1975, 1980, 1985, 1990, 1995, 2000, 2005, 2010, 2015, 2020, 2025, 2030, 2035, 2040, 2045, 2050, 2055, 2060, 2065, 2070, 2075, 2080, 2085, 2090, 2095, 2100, 2105, 2110, 2115, 2120, 2125, 2130, 2135, 2140, 2145, 2150, 2155, 2160, 2165, 2170, 2175, 2180, 2185, 2190, 2195, 2200, 2205, 2210, 2215, 2220, 2225, 2230, 2235, 2240, 2245, 2250, 2255, 2260, 2265, 2270, 2275, 2280, 2285, 2290, 2295, 2300, 2305, 2310, 2315, 2320, 2325, 2330, 2335, 2340, 2345, 2350, 2355, 2360, 2365, 2370, 2375, 2380, 2385, 2390, 2395, 2400, 2405, 2410, 2415, 2420, 2425, 2430, 2435, 2440, 2445, 2450, 2455, 2460, 2465, 2470, 2475, 2480, 2485, 2490, 2495, 2500, 2505, 2510, 2515, 2520, 2525, 2530, 2535, 2540, 2545, 2550, 2555, 2560, 2565, 2570, 2575, 2580, 2585, 2590, 2595, 2600, 2605, 2610, 2615, 2620, 2625, 2630, 2635, 2640, 2645, 2650, 2655, 2660, 2665, 2670, 2675, 2680, 2685, 2690, 2695, 2700, 2705, 2710, 2715, 2720, 2725, 2730, 2735, 2740, 2745, 2750, 2755, 2760, 2765, 2770, 2775, 2780, 2785, 2790, 2795, 2800, 2805, 2810, 2815, 2820, 2825, 2830, 2835, 2840, 2845, 2850, 2855, 2860, 2865, 2870, 2875, 2880, 2885, 2890, 2895, 2900, 2905, 2910, 2915, 2920, 2925, 2930, 2935, 2940, 2945, 2950, 2955, 2960, 2965, 2970, 2975, 2980, 2985, 2990, 2995, 3000, 3005, 3010, 3015, 3020, 3025, 3030, 3035, 3040, 3045, 3050, 3055, 3060, 3065, 3070, 3075, 3080, 3085, 3090, 3095, 3100, 3105, 3110, 3115, 3120, 3125, 3130, 3135, 3140, 3145, 3150, 3155, 3160, 3165, 3170, 3175, 3180, 3185, 3190, 3195, 3200, 3205, 3210, 3215, 3220, 3225, 3230, 3235, 3240, 3245, 3250, 3255, 3260, 3265, 3270, 3275, 3280, 3285, 3290, 3295, 3300, 3305, 3310, 3315, 3320, 3325, 3330, 3335, 3340, 3345, 3350, 3355, 3360, 3365, 3370, 3375, 3380, 3385, 3390, 3395, 3400, 3405, 3410, 3415, 3420, 3425, 3430, 3435, 3440, 3445, 3450, 3455, 3460, 3465, 3470, 3475, 3480, 3485, 3490, 3495, 3500, 3505, 3510, 3515, 3520, 3525, 3530, 3535, 3540, 3545, 3550, 3555, 3560, 3565, 3570, 3575, 3580, 3585, 3590, 3595, 3600, 3605, 3610, 3615, 3620, 3625, 3630, 3635, 3640, 3645, 3650, 3655, 3660, 3665, 3670, 3675, 3680, 3685, 3690, 3695, 3700, 3705, 3710, 3715, 3720, 3725, 3730, 3735, 3740, 3745, 3750, 3755, 3760, 3765, 3770, 3775, 3780, 3785, 3790, 3795, 3800, 3805, 3810, 3815, 3820, 3825, 3830, 3835, 3840, 3845, 3850, 3855, 3860, 3865, 3870, 3875, 3880, 3885, 3890, 3895, 3900, 3905, 3910, 3915, 3920, 3925, 3930, 3935, 3940, 3945, 3950, 3955, 3960, 3965, 3970, 3975, 3980, 3985, 3990, 3995, 4000, 4005, 4010, 4015, 4020, 4025, 4030, 4035, 4040, 4045, 4050, 4055, 4060, 4065, 4070, 4075, 4080, 4085, 4090, 4095, 4100, 4105, 4110, 4115, 4120, 4125, 4130, 4135, 4140, 4145, 4150, 4155, 4160, 4165, 4170, 4175, 4180, 4185, 4190, 4195, 4200, 4205, 4210, 4215, 4220, 4225, 4230, 4235, 4240, 4245, 4250, 4255, 4260, 4265, 4270, 4275, 4280, 4285, 4290, 4295, 4300, 4305, 4310, 4315, 4320, 4325, 4330, 4335, 4340, 4345, 4350, 4355, 4360, 4365, 4370, 4375, 4380, 4385, 4390, 4395, 4400, 4405, 4410, 4415, 4420, 4425, 4430, 4435, 4440, 4445, 4450, 4455, 4460, 4465, 4470, 4475, 4480, 4485, 4490, 4495, 4500, 4505, 4510, 4515, 4520, 4525, 4530, 4535, 4540, 4545, 4550, 4555, 4560, 4565, 4570, 4575, 4580, 4585, 4590, 4595, 4600, 4605, 4610, 4615, 4620, 4625, 4630, 4635, 4640, 4645, 4650, 4655, 4660, 4665, 4670, 4675, 4680, 4685, 4690, 4695, 4700, 4705, 4710, 4715, 4720, 4725, 4730, 4735, 4740, 4745, 4750, 4755, 4760, 4765, 4770, 4775, 4780, 4785, 4790, 4795, 4800, 4805, 4810, 4815, 4820, 4825, 4830, 4835, 4840, 4845, 4850, 4855, 4860, 4865, 4870, 4875, 4880, 4885, 4890, 4895, 4900, 4905, 4910, 4915, 4920, 4925, 4930, 4935, 4940, 4945, 4950, 4955, 4960, 4965, 4970, 4975, 4980, 4985, 4990, 4995, 5000, 5005, 5010, 5015, 5020, 5025, 5030, 5035, 5040, 5045, 5050, 5055, 5060, 5065, 5070, 5075, 5080, 5085, 5090, 5095, 5100, 5105, 5110, 5115, 5120, 5125, 5130, 5135, 5140, 5145, 5150, 5155, 5160, 5165, 5170, 5175, 5180, 5185, 5190, 5195, 5200, 5205, 5210, 5215, 5220, 5225, 5230, 5235, 5240, 5245, 5250, 5255, 5260, 5265, 5270, 5275, 5280, 5285, 5290, 5295, 5300, 5305, 5310, 5315, 5320, 5325, 5330, 5335, 5340, 5345, 5350, 5355, 5360, 5365, 5370, 5375, 5380, 5385, 5390, 5395, 5400, 5405, 5410, 5415, 5420, 5425, 5430, 5435, 5440, 5445, 5450, 5455, 5460, 5465, 5470, 5475, 5480, 5485, 5490, 5495, 5500, 5505, 5510, 5515, 5520, 5525, 5530, 5535, 5540, 5545, 5550, 5555, 5560, 5565, 5570, 5575, 5580, 5585, 5590, 5595, 5600, 5605, 5610, 5615, 5620, 5625, 5630, 5635, 5640, 5645, 5650, 5655, 5660, 5665, 5670, 5675, 5680, 5685, 5690, 5695, 5700, 5705, 5710, 5715, 5720, 5725, 5730, 5735, 5740, 5745, 5750, 5755, 5760, 5765, 5770, 5775, 5780, 5785, 5790, 5795, 5800, 5805, 5810, 5815, 5820, 5825, 5830, 5835, 5840, 5845, 5850, 5855, 5860, 5865, 5870, 5875, 5880, 5885, 5890, 5895, 5900, 5905, 5910, 5915, 5920, 5925, 5930, 5935, 5940, 5945, 5950, 5955, 5960, 5965, 5970, 5975, 5980, 5985, 5990, 5995, 6000, 6005, 6010, 6015, 6020, 6025, 6030, 6035, 6040, 6045, 6050, 6055, 6060, 6065, 6070, 6075, 6080, 6085, 6090, 6095, 6100, 6105, 6110, 6115, 6120, 6125, 6130, 6135, 6140, 6145, 6150, 6155, 6160, 6165, 6170, 6175, 6180, 6185, 6190, 6195, 6200, 6205, 6210, 6215, 6220, 6225, 6230, 6235, 6240, 6245, 6250, 6255, 6260, 6265, 6270, 6275, 6280, 6285, 6290, 6295, 6300, 6305, 6310, 6315, 6320, 6325, 6330, 6335, 6340, 6345, 6350, 6355, 6360, 6365, 6370, 6375, 6380, 6385, 6390, 6395, 6400, 6405, 6410, 6415, 6420, 6425, 6430, 6435, 6440, 6445, 6450, 6455, 6460, 6465, 6470, 6475, 6480, 6485, 6490, 6495, 6500, 6505, 6510, 6515, 6520, 6525, 6530, 6535, 6540, 6545, 6550, 6555, 6560, 6565, 6570, 6575, 6580, 6585, 6590, 6595, 6600, 6605, 6610, 6615, 6620, 6625, 6630, 6635, 6640, 6645, 6650, 6655, 6660, 6665, 6670, 6675, 6680, 6685, 6690, 6695, 6700, 6705, 6710, 6715, 6720, 6725, 6730, 6735, 6740, 6745, 6750, 6755, 6760, 6765, 6770, 6775, 6780, 6785, 6790, 6795, 6800, 6805, 6810, 6815, 6820, 6825, 6830, 6835, 6840, 6845, 6850, 6855, 6860, 6865, 6870, 6875, 6880, 6885, 6890, 6895, 6900, 6905, 6910, 6915, 6920, 6925, 6930, 6935, 6940, 6945, 6950, 6955, 6960, 6965, 6970, 6975, 6980, 6985, 6990, 6995, 7000, 7005, 7010, 7015, 7020, 7025, 7030, 7035, 7040, 7045, 7050, 7055, 7060, 7065, 7070, 7075, 7080, 7085, 7090, 7095, 7100, 7105, 7110, 7115, 7120, 7125, 7130, 7135, 7140, 7145, 7150, 7155, 7160, 7165, 7170, 7175, 7180, 7185, 7190, 7195, 7200, 7205, 7210, 7215, 7220, 7225, 7230, 7235, 7240, 7245, 7250, 7255, 7260, 7265, 7270, 7275, 7280, 7285, 7290, 7295, 7300, 7305, 7310, 7315, 7320, 7325, 7330, 7335, 7340, 7345, 7350, 7355, 7360, 7365, 7370, 7375, 7380, 7385, 7390, 7395, 7400, 7405, 7410, 7415, 7420, 7425, 7430, 7435, 7440, 7445, 7450, 7455, 7460, 7465, 7470, 7475, 7480, 7485, 7490, 7495, 7500, 7505, 7510, 7515, 7520, 7525, 7530, 7535, 7540, 7545, 7550, 7555, 7560, 7565, 7570, 7575, 7580, 7585, 7590, 7595, 7600, 7605, 7610, 7615, 7620, 7625, 7630, 7635, 7640, 7645, 7650, 7655, 7660, 7665, 7670, 7675, 7680, 7685, 7690, 7695, 7700, 7705, 7710, 7715, 7720, 7725, 7730, 7735, 7740, 7745, 7750, 7755, 7760, 7765, 7770, 7775, 7780, 7785, 7790, 7795, 7800, 7805, 7810, 7815, 7820, 7825, 7830, 7835, 7840, 7845, 7850, 7855, 7860, 7865, 7870, 7875, 7880, 7885, 7890, 7895, 7900, 7905, 7910, 7915, 7920, 7925, 7930, 7935, 7940, 7945, 7950, 7955, 7960, 7965, 7970, 7975, 7980, 7985, 7990, 7995, 8000, 8005, 8010, 8015, 8020, 8025, 8030, 8035, 8040, 8045, 8050, 8055, 8060, 8065, 8070, 8075, 8080, 8085, 8090, 8095, 8100, 8105, 8110, 8115, 8120, 8125, 8130, 8135, 8140, 8145, 8150, 8155, 8160, 8165, 8170, 8175, 8180, 8185, 8190, 8195, 8200, 8205, 8210, 8215, 8220, 8225, 8230, 8235, 8240, 8245, 8250, 8255, 8260, 8265, 8270, 8275, 8280, 8285, 8290, 8295, 8300, 8305, 8310, 8315, 8320, 8325, 8330, 8335, 8340, 8345, 8350, 8355, 8360, 8365, 8370, 8375, 8380, 8385, 8390, 8395, 8400, 8405, 8410, 8415, 8420, 8425, 8430, 8435, 8440, 8445, 8450, 8455, 8460, 8465, 8470, 8475, 8480, 8485, 8490, 8495, 8500, 8505, 8510, 8515, 8520, 8525, 8530, 8535, 8540, 8545, 8550, 8555, 8560, 8565, 8570, 8575, 8580, 8585, 8590, 8595, 8600, 8605, 8610, 8615, 8620, 8625, 8630, 8635, 8640, 8645, 8650, 8655, 8660, 8665, 8670, 8675, 8680, 8685, 8690, 8695, 8700, 8705, 8710, 8715, 8720, 8725, 8730, 8735, 8740, 8745, 8750, 8755, 8760, 8765, 8770, 8775, 8780, 8785, 8790, 8795, 8800, 8805, 8810, 8815, 8820, 8825, 8830, 8835, 8840, 8845, 8850, 8855, 8860, 8865, 8870, 8875, 8880, 8885, 8890, 8895, 8900, 8905, 8910, 8915, 8920, 8925, 8930, 8935, 8940, 8945, 8950, 8955, 8960, 8965, 8970, 8975, 8980, 8985, 8990, 8995, 9000, 9005, 9010, 9015, 9020, 9025, 9030, 9035, 9040, 9045, 9050, 9055, 9060, 9065, 9070, 9075, 9080, 9085, 9090, 9095, 9100, 9105, 9110, 9115, 9120, 9125, 9130, 9135, 9140, 9145, 9150, 9155, 9160, 9165, 9170, 9175, 9180, 9185, 9190, 9195, 9200, 9205, 9210, 9215, 9220, 9225, 9230, 9235, 9240, 9245, 9250, 9255, 9260, 9265, 9270, 9275, 9280, 9285, 9290, 9295, 9300, 9305, 9310, 9315, 9320, 9325, 9330, 9335, 9340, 9345, 9350, 9355, 9360, 9365, 9370, 9375, 9380, 9385, 9390, 9395, 9400, 9405, 9410, 9415, 9420, 9425, 9430, 9435, 9440, 9445, 9450, 9455, 9460, 9465, 9470, 9475, 9480, 9485, 9490, 9495, 9500, 9505, 9510, 9515, 9520, 9525, 9530, 9535, 9540, 9545, 9550, 9555, 9560, 9565, 9570, 9575, 9580, 9585, 9590, 9595, 9600, 9605, 9610, 9615, 9620, 9625, 9630, 9635, 9640, 9645, 9650, 9655, 9660, 9665, 9670, 9675, 9680, 9685, 9690, 9695, 9700, 9705, 9710, 9715, 9720, 9725, 9730, 9735, 9740, 9745, 9750, 9755, 9760, 9765, 9770, 9775, 9780, 9785, 9790, 9795, 9800, 9805, 9810, 9815, 9820, 9825, 9830, 9835, 9840, 9845, 9850, 9855, 9860, 9865, 9870, 9875, 9880, 9885, 9890, 9895, 9900, 9905, 9910, 9915, 9920, 9925, 9930, 9935, 9940, 9945, 9950, 9955, 9960, 9965, 9970, 9975, 9980, 9985, 9990, 9995, 10000, 10005, 10010, 10015, 10020, 10025, 10030, 10035, 10040, 10045, 10050, 10055, 10060, 10065, 10070, 10075, 10080, 10085, 10090, 10095, 10100, 10105, 10110, 10115, 10120, 10125, 10130, 10135, 10140, 10145, 10150, 10155, 10160, 10165, 10170, 10175, 10180, 10185, 10190, 10195, 10200, 10205, 10210, 10215, 10220, 10225, 10230, 10235, 10240, 10245, 10250, 10255, 10260, 10265, 10270, 10275, 10280, 10285, 10290, 10295, 10300, 10305, 10310, 10315, 10320, 10325, 10330, 10335, 10340, 10345, 10350, 10355, 10360, 10365, 10370, 10375, 10380, 10385, 10390, 10395, 10400, 10405, 10410, 10415, 10420, 10425, 10430, 10435, 10440, 10445, 10450, 10455, 10460, 10465, 10470, 10475, 10480, 10485, 10490, 10495, 10500, 10505, 10510, 10515, 10520, 10525, 10530, 10535, 10540, 10545, 10550, 10555, 10560, 10565, 10570, 10575, 10580, 10585, 10590, 10595, 10600, 10605, 10610, 10615, 10620, 10625, 10630, 10635, 10640, 10645, 10650, 10655, 10660, 10665, 10670, 10675, 10680, 10685, 10690, 10695, 10700, 10705, 10710, 10715, 10720, 10725, 10730, 10735, 10740, 10745, 10750, 10755, 10760, 10765, 10770, 10775, 10780, 10785, 10790, 10795, 10800, 10805, 10810, 10815, 10820, 10825, 10830, 10835, 10840, 10845, 10850, 10855, 10860, 10865, 10870, 10875, 10880, 10885, 10890, 10895, 10900, 10905, 10910, 10915, 10920, 10925, 10930, 10935, 10940, 10945, 10950, 10955, 10960, 10965, 10970, 10975, 10980, 10985, 10990, 10995, 11000, 11005, 11010, 11015, 11020, 11025, 11030, 11035, 11040, 11045, 11050, 11055, 11060, 11065, 11070, 11075, 11080, 11085, 11090, 11095, 11100, 11105, 11110, 11115, 11120, 11125, 11130, 11135, 11140, 11145, 11150, 11155, 11160, 11165, 11170, 11175, 11180, 11185, 11190, 11195, 11200, 11205, 11210, 11215, 11220, 11225, 11230, 11235, 11240, 11245, 11250, 11255, 11260, 11265, 11270, 11275, 11280, 11285, 11290, 11295, 11
```


برای ژیمناستیک تنها ۸ پاسخ پیدا شد که همانطور که گفته شد مرتب شده است.

```
enter a word for checking: ژیمناستیک
Normal word to search ژیمناستیک
[[1367, 12, [15, 48, 126, 132, 145, 185, 231, 245, 281, 287, 300, 306]], [632, 12, [14, 35, 70, 137, 154, 164, 223, 237, 261, 324, 339, 380]], [4188, 3, [2
8 results in 0.361 ms
id -> title

1367 -> هشدار هیت ژیمناستیک تهران در خصوص سالن‌های مختلط و اقدامات غیرواخلاقی ->
632 -> خیرخواه: برخی به دنبال طعنه زدن ژیمناستیک هستند/ با باکوت فدراسیون موفقیت‌ها بیشتر شد ->
4188 -> جزئیات تملیلی‌های ورزش ایران تا ۹ مهر ۱۴۰۰/ کدام فعالیت‌های ورزشی در تهران ممنوع است؟ ->
3664 -> ثبت نام ۱۳ نفره برای تست و تست فدراسیون ژیمناستیک + اسامی ->
3878 -> جزئیات تملیلی ورزش ایران تا پایان نیمه‌امه تصویر ->
1456 -> سوت‌زنی | کنکور و برخورد با سالن‌های ورزشی مختلط توسط وزارت ورزش ->
4056 -> دبیر: اگر من در مباحث فنی ۱۰ باتم، درست‌کار ۱۰۰ است/ بنا کاملاً بر اساس چرخه انتخابی عمل کرد ->
3615 -> دبیر مجمع فدراسیون ژیمناستیک مشخص شد ->
[[1367, 12, [15, 48, 126, 132, 145, 185, 231, 245, 281, 287, 300, 306]], [632, 12, [14, 35, 70, 137, 154, 164, 223, 237, 261, 324, 339, 380]], [4188, 3, [2
```

در ادامه میبایستی برای کوئری های چندتایی بایستی از position intersection استفاده کنیم که به صورت زیر است و موقعیت مکانی رو مرج میکند.

```
# get the to word position list and start find k position for p1 , p2
def position_intersect(p1,p2,k):
    answer = []
    len1 = len(p1)
    len2 = len(p2)
    i = j = 0
    while i != len1 and j != len2:
        if docID(p1[i]) == docID(p2[j]):
            l = []
            pp1 = position(p1[i])
            pp2 = position(p2[j])

            plen1 = len(pp1)
            plen2 = len(pp2)
            ii = jj = 0
            while ii != plen1:
                while jj != plen2:
                    if abs(pp1[ii] - pp2[jj]) <= k:
                        l.append(pp2[jj])
                    elif pp2[jj] > pp1[ii]:
                        break
                    jj+=1
                while l != [] and abs(l[0] - pp1[ii]) > k :
                    l.remove(l[0])
                for ps in l:
                    answer.append([ docID(p1[i]), pp1[ii], ps ])
                ii+=1
            i+=1
            j+=1
        elif docID(p1[i]) < docID(p2[j]):
            i+=1
        else:
            j+=1
    return answer
```

و برای گرفتن doc id و لیست موقعیت‌ها از ای دو استفاده شده

```
[47] # the dictionary values is the posting lists
# post_list = dictionary[key][i]
# the i is iteated
# return docID
def docID(post_list):
    return post_list[0]
```

```
# the dictionary values is the posting lists
# post_list = dictionary[key][i]
# the i is iteated |
# return list of position
def position(plist):
    return plist[2]
```

در کوئری‌های دوتایی کنار هم بودن دو کلمه برای ما ارزش دارد پس از position intersection استفاده می‌کنیم تا با $k = 1$ وجود دوتایی آن‌ها را بیابیم برای دانشگاه امیرکبیر پاسخ به صورت زیر بوده است که با توجه به اینکه نیاز به مرج و مقایسه بین دو position intersection زمان بر تر خواهد بود و تعداد کمتری جواب خواهد داشت اینجا هم تعداد پاسخ‌ها کمتر از قبل و تعداد تکی این کلمات هست.

```
enter words for checking: دانشگاه امیرکبیر
Normal word to search دانشگاه
2625
[[362, 1, [195]], [390, 10, [18, 21, 76, 206, 209, 238, 263, 342, 362, 374]], [393, 1, [18]], [552, 2, [108, 167]], [560, 1, [80]], [724, 3, [89, 133, 136]]]
Normal word to search امیرکبیر
23
[[1753, 1, [4038]], [1959, 1, [574]], [2131, 1, [511]], [2709, 1, [436]], [2792, 4, [18, 182, 223, 302]], [2793, 4, [12, 15, 231, 316]], [4580, 1, [95]], [2625, [[362, 1, [195]], [390, 10, [18, 21, 76, 206, 209, 238, 263, 342, 362, 374]], [393, 1, [18]], [552, 2, [108, 167]], [560, 1, [80]], [724, 3, [89, 2]]]
[[2793, 13, 12], [2792, 17, 18], [7236, 33, 34], [2792, 181, 182], [2792, 222, 223], [2793, 230, 231], [2792, 301, 302], [2793, 315, 316], [2709, 437, 436]]
defaultdict(<class 'list'>, {2793: [[13, 12], [230, 231], [315, 316]], 2792: [[17, 18], [181, 182], [222, 223], [301, 302]], 7236: [[33, 34]], 2709: [[437]]})
[2792, 2793, 7236, 2709, 2131, 1959, 1753]
7 results in 10.651000000000002 ms
id -> title

2792 -> امروز محیط دانشگاه‌های ما عرصه نفاق مقدس است
2793 -> باید برای ثبت نقش دانشگاهیان در دوران نفاق مقدس کار تحقیقی صورت گیرد
7236 -> برگزاری یادبودی برای محسن‌سور رجایی
2709 -> پیشرفت صنایع هوایی ما قائل مقایسه با قتل و انقلاب نیست/ شب و روز سپهرنویس‌ها در وحشت است
2131 -> بزرگداشت شهدای مسجد قدس در مقابل کمولگری افغانستان/ آمریکا و آل سعود مفسران اصلی جنایت در افغانستان
1959 -> نامه ۸ بسج دانشجویی دانشگاه‌های تهران به معنرین اول رئیس جمهور
1753 -> نامه جمعی از اساتید و محققان/ آقای رئیس‌جمهور در گام دوم انقلاب به داد «هنریت» در کشور برسید
```

برای کلمه واکسن آسترزنکا هم پاسخ به شکل زیر بوده

```

واکسن آسترازنکا
enter words for checking:
Normal word to search
2583
[[327, 1, [343]], [374, 1, [319]], [640, 1, [1194]], [782, 1, [86]], [1021, 1, [16]], [1032, 2, [403, 508]], [1034, 5, [12, 80],
Normal word to search آسترازنکا
47
[[4931, 1, [30]], [5569, 1, [30]], [5685, 3, [205, 401, 499]], [5823, 2, [1487, 1741]], [5825, 6, [387, 415, 456, 499, 1101, 23],
[(2583, [[327, 1, [343]], [374, 1, [319]], [640, 1, [1194]], [782, 1, [86]], [1021, 1, [16]], [1032, 2, [403, 508]], [1034, 5,
2
[[4931, 29, 30], [5569, 29, 30], [5685, 204, 205], [5857, 213, 214], [5845, 254, 255], [6336, 377, 378], [5825, 386, 387], [583
defaultdict(<class 'list'>, {4931: [[29, 30]], 5569: [[29, 30]], 5685: [[204, 205], [400, 401], [498, 499]], 5857: [[213, 214]]
5825, 5833, 5685, 4931, 5569, 5857, 5845, 6336]
8 results in 10.436 ms
id -> title

5825 -> نکاتی که باید در مورد واکسن‌های کرونا بدانیم
5833 -> نکاتی که باید در مورد واکسن‌های کرونا بدانیم
5685 -> بهترین سلاح مبارزه با کرونا
4931 -> محموله ۱.۴ میلیون دوزی واکسن کرونا وارد کشور شد
5569 -> محموله ۱.۴ میلیون دوزی واکسن کرونا وارد کشور شد
5857 -> واکسن‌های کرونا با چه داروهایی تکامل دارند؟
5845 -> مقررات تازه برای سفر زمینی ایران و ارمنستان
6336 -> امکان ایجاد لخته خون در واکسن آسترازنکا چقدر است؟

```

در ابتدا با توجه به لینک ها به صورت سورت شده که در این لینک ها به صورت سورت شده بنابر تعداد آمدن این دو کنار هم در لینک هاست. به مانند قبلی برای مرج کردن زمان بیشتری برده است.

برای سه تایی ها و بیشتر دوتایی دوتایی کنار هم رو می توانیم بگیریم و با position intersection موقعیت مکانی های اون دوتا را با هم بدست بیاریم و سپس باید در متن های یکی دنبال اون مکان های بگردیم که کلمه مشترکشون یک جا باشد تا پاسخ دهد (این کار رو میتوان به صورت بازگشتی انجام داد)

برای مثال جمهوری اسلامی ایران

enter words for checking: جمهوری اسلامی ایران
 Normal word to search جمهوری
 3746
 [[88, 1, [1911]], [109, 1, [53]], [189, 2, [452, 900]], [314, 1, [298]], [417, 2, [385, 388]], [446, 2, [225, 245]], [493, 2, [1449, 1493]], [510, 2, [383, 386]], [510, 2, [383, 386]], [510, 2, [383, 386]]]
 Normal word to search اسام
 0
 []
 Normal word to search ایر
 11430
 [[2, 1, [96]], [5, 7, [1232, 1354, 1596, 1614, 1865, 3239, 3281]], [6, 1, [92]], [12, 1, [396]], [16, 1, [221]], [39, 1, [91]], [50, 1, [81]], [52, 1, [11]], [65, [3746, [80, 1, [1911]], [109, 1, [53]], [189, 2, [452, 900]], [314, 1, [298]], [417, 2, [385, 388]], [446, 2, [225, 245]], [493, 2, [1449, 1493]], [510, 2, [383, 386]], [510, 2, [383, 386]]]
 2
 [[4506, 12, 11], [2310, 13, 12], [6702, 14, 13], [1777, 15, 14], [2926, 16, 15], [2928, 17, 16], [2925, 21, 20], [3018, 23, 22], [2716, 24, 23], [6739, 72, 71], [defaultdict({<class 'list'>, [4506: [[12, 11]], 2310: [[13, 12]], 6702: [[14, 13]], 1777: [[15, 14]], 2926: [[16, 15]], 2928: [[17, 16]], [90, 89]], 2925: [[21, 20]]
 2928, 2908, 2785, 4506, 2310, 6702, 1777, 2926, 2925, 3018, 2716, 6739, 4057, 3035, 4066, 2741, 3262, 2492, 3111, 3362, 2724, 4531, 6514, 2217]
 24 results in 25.386 ms
 id -> title

در اینجا جمهوری به جمهور و اسلامی به اسلام و ایران به ایر نرمال شده حالا در کل متون اسلام وجود ندارد به صورت خالی پس فقط جمهور و ایر رو مرج می‌کنیم و در نهایت لیست پایانی را سورت می‌کنیم.

در ادامه دانشگاه صنعتی امیرکبیر هم به مانند بالا اتفاق می افتد با توجه به متن هایی که بازگردانده شده مشخص شده که اطلاعات بازگردانده شده درست است.

```
❏ enter words for checking: دانشگاه صنعتی امیرکبیر
Normal word to search دانشگاه
2625
[[362, 1, [195]], [390, 10, [18, 21, 76, 206, 209, 238, 263, 342, 362, 374]], [393, 1, [18]], [552, 2, [108, 167]], [560, 1, [80]], [724, 3, [89, 133, 136]], [
Normal word to search صنعت
0
[]
Normal word to search امیرکبیر
23
[[1753, 1, [4038]], [1959, 1, [574]], [2131, 1, [511]], [2709, 1, [436]], [2792, 4, [18, 182, 223, 302]], [2793, 4, [12, 15, 231, 316]], [4580, 1, [95]], [5021
[2625, [[362, 1, [195]], [390, 10, [18, 21, 76, 206, 209, 238, 263, 342, 362, 374]], [393, 1, [18]], [552, 2, [108, 167]], [560, 1, [80]], [724, 3, [89, 133,
2
[[2793, 13, 12], [2792, 17, 18], [7236, 33, 34], [2792, 181, 182], [2792, 222, 223], [2793, 230, 231], [2792, 301, 302], [2793, 315, 316], [2709, 437, 436], [2
defaultdict(<class 'list'>, {2793: [[13, 12], [230, 231], [315, 316]], 2792: [[17, 18], [181, 182], [222, 223], [301, 302]], 7236: [[33, 34]], 2709: [[437, 436
[2792, 2793, 7236, 2709, 2131, 1959, 1753]
7 results in 9.078 ms
id -> title

2792 -> امروز محیط دانشگاهی ما عرصه دفاع مقدس است
2793 -> یاد برای ثبت نقش دانشگاهیان در دوران دفاع مقدس کار تحقیقی صورت گیرد
7236 -> برگزاری یادبودی برای محمدرضا رجایی
2709 -> بهشت صنعت هوایی ما قفل مفیسه با قفل آن انقلاب نیست/ شب و روز سپهرنیست‌ها در ریخت است
2131 -> بزرگداشت شهدای مسجد فتول در مقابل کنسولگری افغانستان/ آمریکا و آل سعود مفسران اصلی جنایت در افغانستان
1753 -> نامه ۸ بسیج دانشجویی دانشگاهی تهران به معاون اول رئیس جمهور
1959 -> نامه جمعی از اساتید و متخصصان/ آقای رئیس‌جمهور در گام دوم انقلاب به داد «سپهریت» در کشور برسید
1753
```

در آخرین کوئری سازمان به ساز نرمال شد و بقیه به همان فرم ماندند

```
enter words for checking: سازمان ملل متحد
Normal word to search ساز
0
[]
Normal word to search ملل
204
[[102, 2, [370, 434]], [106, 2, [22, 85]], [202, 2, [355, 444]], [210, 3, [94, 1106, 1426]], [1146, 1, [525]], [1733, 1, [332]], [1749, 1, [1
Normal word to search متحد
103
[[403, 1, [448]], [560, 1, [940]], [1743, 1, [761]], [1748, 3, [258, 680, 1140]], [1782, 1, [820]], [1794, 1, [117]], [1910, 1, [210]], [1913
[(0, []), (204, [102, 2, [370, 434]], [106, 2, [22, 85]], [202, 2, [355, 444]], [210, 3, [94, 1106, 1426]], [1146, 1, [525]], [1733, 1, [332
2
[[2785, 9, 10], [6873, 19, 20], [6799, 20, 21], [6873, 27, 28], [2741, 34, 35], [2743, 36, 37], [2770, 37, 38], [2781, 41, 42], [1972, 46, 47
defaultdict(<class 'list'>, {2785: [[9, 10], [423, 424], [1099, 1100], [1572, 1573], [2064, 2065], [2728, 2729]], 6873: [[19, 20], [27, 28],
[2785, 6873, 2781, 6799, 2743, 2301, 6911, 6567, 2217, 2298, 2741, 2770, 1972, 4781, 4772, 2038, 3374, 1973, 3195, 2861, 7543, 5086, 2025, 27
32 results in 2.227 ms
id -> title

نگاهی به ۸ سخنرانی روحانی در سازمان ملل متحد/ از ابراز علاقمندی به سخنان اوپاما تا نصیحت‌های برجایی -> 2785
پیشنهادی برای کم کردن تنش بین کشورهای حوزه نوروز -> 6873
رئیس: گفت و گویی که به لغو تحریم‌ها منجر نشود را انجام نمی دهیم/ تحریم دارو در دوران کرونا جنایت علیه بشریت است -> 2781
شهرهای میوبای -> 6799
راستینه: اظهارات آیت الله رئیسی مطالبه‌گری انقلابی در سازمان ملل بود/ تاکید رئیس جمهور بر منافع ملت -> 2743
هیئت پارلمانی ایران به رم سفر کرد -> 2301
انتشار «کودکان و حقوق بشر» برای بجما -> 6911
استدلال‌ها برای ادامه جنگ ید از فتح خرمشهر چه بود؟ -> 6567
نامه بسیج دانشگاه امام صادق به وزیر خارجه سلیق/ آقای ظریف! شما بیشتر وزیر امور برجام بودید تا وزیر خارجه -> 2217
اعتراض هیات پارلمانی ایران به نماینده منصور هادی در اجلاس تئیزرات آب و هوایی/ همکاری دولت مستفی یمن با ائتلاف سعودی علیه یمنی‌ها -> 2298
اسفندی: اظهارات رئیسی در سازمان ملل اقدامی برای رسوا کردن غربی‌ها و احقاق حقوق مردم بود -> 2741
اظهارات رئیسی در سازمان ملل مطالبات به حق ملت ایران بود -> 2770
بررسی طرح تکمیل از دانشگاه تربیت دبیر شهید رجایی در کمیسیون آموزش -> 1972
```

با توجه به پاسخ‌ها وجود سازمان ملل متحد در تیترز اولین لینک (و صد البته حضور نام آقای روحانی ^_^) نشان دهند درست بودن جوابها دارد.

سوال ۲) با توجه به اینکه ما رابطه log از Cfi ها رو می‌خواهیم و این رابطه به صورتی هست که بایستی به صورت خطی باشد با توجه به این که قبل از اینکه بخواهیم stop word را حذف کنیم حدودا شبیه و متمایل به یک رابطه خطی هست میتوان گفت صدق میکند(با توجه به اینکه فقط چندین stop word کمی وجود داشت که فاصله زیادی داشت انتظار همین چیزی می‌رفت)

سوال ۳) در قانون heap رابطه خطی با \log آن دارد مقادیر k, b دارد که حدودا هر دو ثابت هستند البته که امکان تغییر برای k رو داریم که می‌توانیم فاصله برای کلمات در متن از هم استفاده کرد و تعداد کلمات می‌شود.

سوال ۴) توی ریشه‌یابی یکی از مسائل مهم فعل بودن یا کلمه بودن بود که مثال بارز آن است بود که اگر ابتدا به عنوان کلمه در نظر گرفته می‌شد به اس تبدیل می‌شد و فعل (Lemmatizer) تاثیری بر آن نداشت به طور کلی ت در انتهای کلمات که حذف می‌شد می‌توانست برای فعل ها مشکل ساز باشد البته ها و ... هم حذف می‌شد. کلمه اعلام که به وفور در خبرها استفاده می‌شود به اعلا با نرمال کردن تغییر میکند که کلمه‌ای با معنی جداست. یکی از مشکلات دیگه کلمه سازمان بود که در نرمال کردن آن به ساز تبدیل می‌شد و باعث می‌شد به فعل تبدیل شود. فعل‌های که به ان ختم میشوند مثل نمی‌توان هم به نمی‌تو تبدیل میشد که با stop ها به مشکل می‌خورد.

سوال ۵) و سوال ۱) هم در متنهای بالا در موردشان صحبت شد.

با تشکر از توجه شما * *