به نام خدا

گزارش فاز یک پروژه بازیابی امیرحسن سعادتمند
۹۷۳۱۰۲۹

۱)
طبق مراحل گفته شده در دستور کار و مطالب تدریس شده در کلاس چهار گام اصلی انجام میشود

۱) نرمالایز کردن هر داکیومنت
۲) توکنایز کردن (استخراج توکن از هر سند)
۳) ریشه یابی (steming)
۴) حذف stopwords ها (کلمات پرتکرار .)

ابتدا data structure ها و کلاس های ساخته شده را بررسی میکنیم

```python
import os

import pandas as pd


class Adapter:
    base_dir = os.path.dirname(__file__)
    file_name = "IR1_7k_news.xlsx"
    data_path = os.path.join(base_dir, file_name)

    def read_excel_file(self):
        df = pd.read_excel(self.data_path)
        return df

    def get_doc_id_and_documents(self):
        df = self.read_excel_file()
        for doc_id, document in enumerate(df.values):
            yield doc_id, document[0], document[2]


get_doc_id_and_document = Adapter().get_doc_id_and_documents()
```

کلاس Adapter وظیفه خواندن از فایل اکسل به صورت داک به داک (تکی تکی) و به صورت جنریک نوشته شده تا قابل iterate داشته باشد و نیاز نباشد در وهله اول همه چیز خوانده شود

کلاس MyTokenizer است که در آن Stopword ها را از فایلی به نام stopwords.bat در پروژه هضم است دربردارد به علاوه آبجکت iterable از داکیومنت هایی که از adapter فرستاده میشوند

متد اصلی این کلاس get_tokenized_stream است .. به طوری نوشته شده که هرداک یکی یکی از Adapter گرفته شده و بعد از نرمالسازی و حذف stopword ها و ریشه یابی استریمی از توکن برمیکرداند
return Tupe(tokens , doc_id)

```python
from parsivar import Normalizer, Tokenizer, FindStems


class MyTokenizer:
    def __init__(self):
        self.stop_words = self.get_stop_words()
        self.docs = get_doc_id_and_document
        self.stemer = FindStems().convert_to_stem

    def get_tokenized_stream(self) -> Tuple:
        for doc_id, document, doc_title in self.docs:
            normalized_document: str = Normalizer().normalize(document)
            token_words: List = Tokenizer().tokenize_words(normalized_document)
            tokens = list()
            for word in token_words:
                stem = self.stemer(word)
                # if stem not in self.stop_words:
                tokens.append(stem)
            yield doc_id, tokens, doc_title

    @staticmethod
    def get_stop_words() -> List:
        base_dir = os.path.dirname(__file__)
        file_name = "stopwords.dat"
        with open(os.path.join(base_dir, file_name), 'r') as file_p:
            stop_word_list = file_p.readlines()
            stop_word_list = [stopword.replace("\n", "") for stopword in stop_word_list]
        return stop_word_list


get_token_stream = MyTokenizer().get_tokenized_stream()
```

دیتا اسراکچر اصلی برای شاخص از نمونه تمرین hw1 موجود در صفحه درس نمونه برداری شده

```python
from typing import Dict


class DocTerm:
    def __init__(self, doc_id, term_id):
        self.count = 0
        self.docId = doc_id
        self.termId = term_id
        self.positions = list()

    def insert(self, position):
        self.count += 1
        self.positions.append(position)
        self.positions = sorted(self.positions)


class Term:
    def __init__(self, id):
        self.count = 0
        self.id = id
        self.docTerms: Dict[DocTerm] = dict()

    def insert(self, doc_id, position):
        self.count += 1
        if doc_id not in self.docTerms.keys():
            self.docTerms[doc_id] = DocTerm(doc_id, self.id)
            sorted(self.docTerms.items())
        self.docTerms[doc_id].insert(position=position)
```

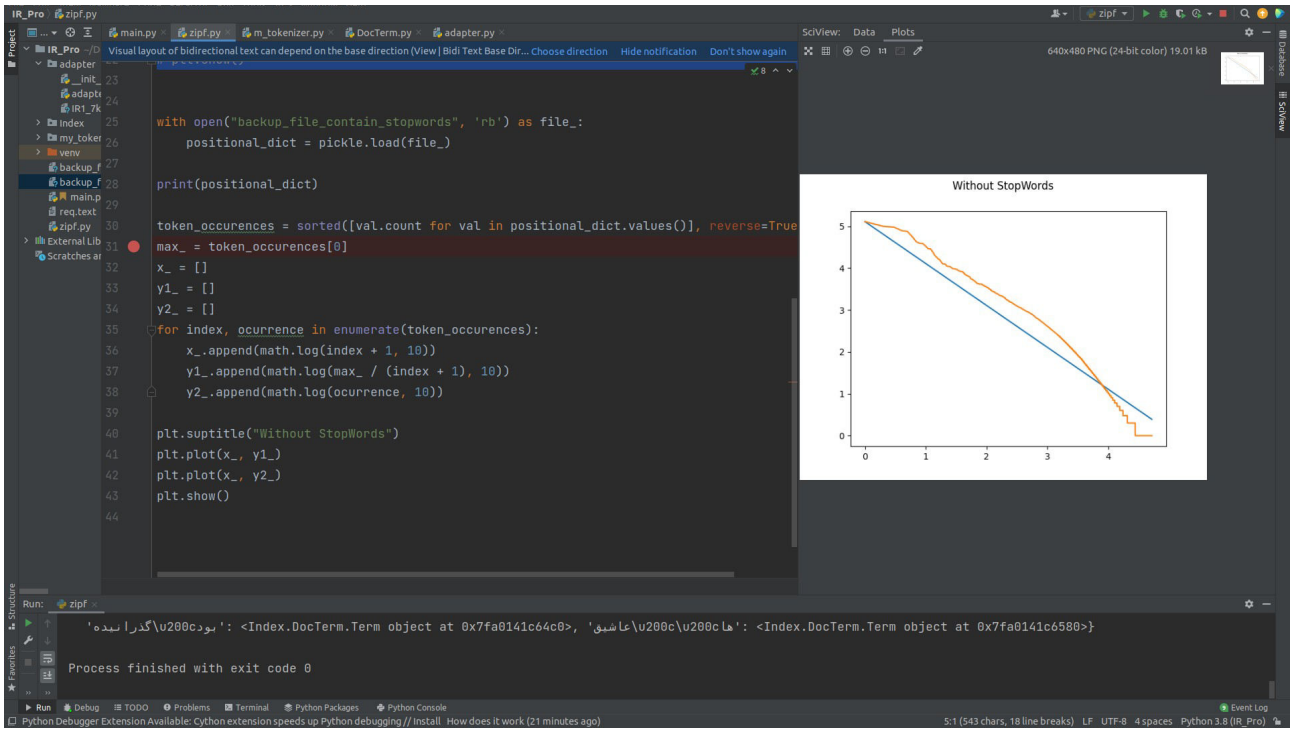
علت اصلی نرمالایز کردن یکسان سازی نوسته هاست به عنوان مثال
همه اعداد به انگلیسی باشند یا چندین فرم از یک کلمه به یک صورت نوشته شوند مثل U_S_A , U.S.A

با tokenize کردن تک تک کلمات استخراج میشوند و آماده ساخت دیکشنری کلمات میشوند

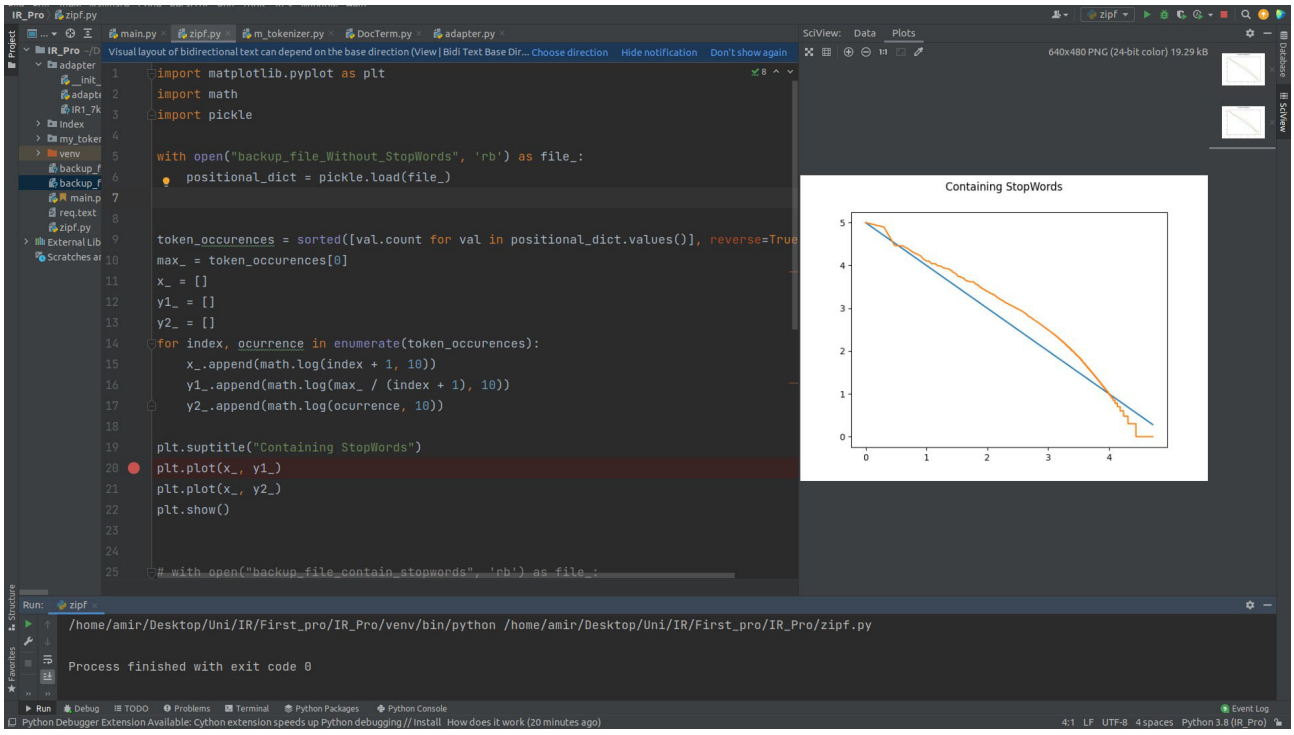
و با ریشه یابی حالت های مختلف یک مفهوم و یا یک فعل را به یک صورت مینویسیم
مثلا همه افعال میگویم بگویم و ... از گفتن هستند

و درنهایت عمل حذف stopword هاست که کلماتی همچون حروف اضافه و ... که خاصیت مفهومی برای بازیابی ندارند را شاخص نمیکنیم

۲)
نمودار پس از حذف stopword ها :



و با در نظر گفتن آنها (شامل stopword ها) :



3)
قبل ریشه یابی :
تعداد توکن ها ۳۲۶۳۹۲
تعداد کلمه ها ۱۷۹۰۴
تعداد همه توکن ها ۱۹۵۹۵۵۲
تعدااد کل کلکمات ۷۲۸۹۰

این در ازای b = 0.67 به دست میاد

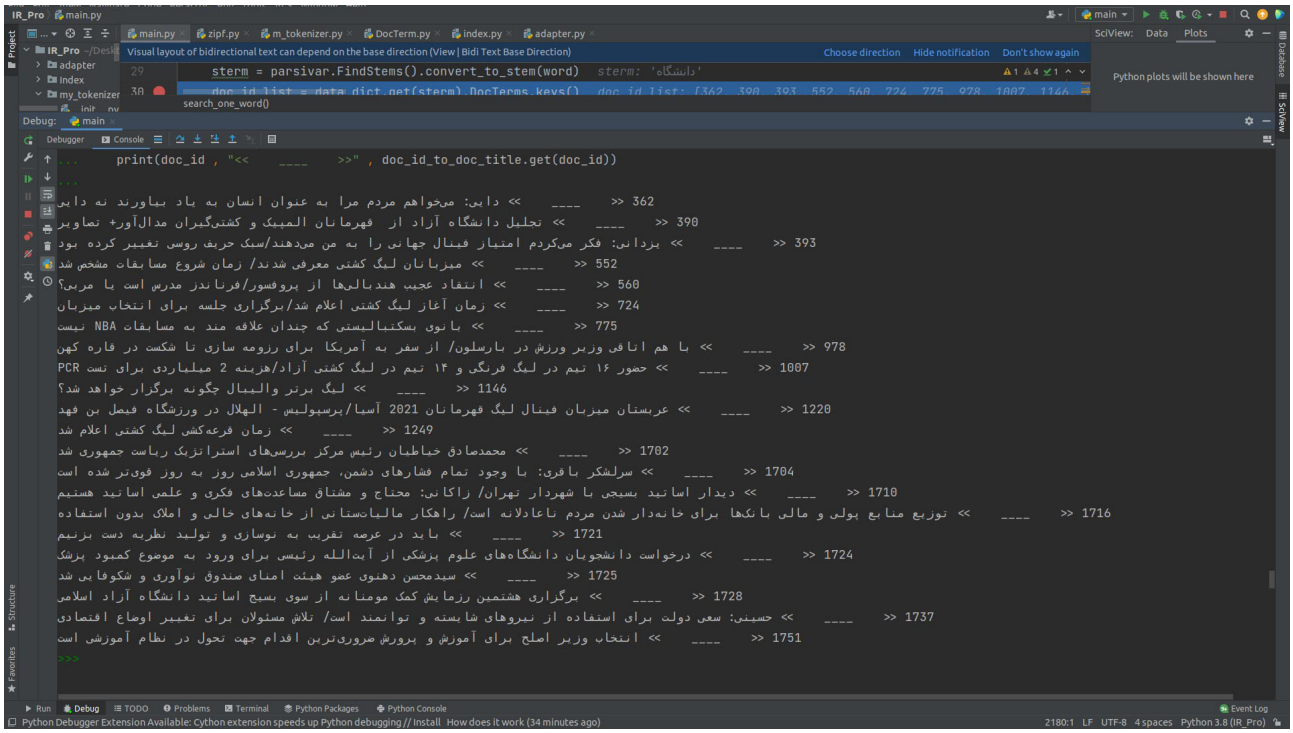
بعد ریشه یابی :
تعداد توکن ها ۳۳۹۸۸۳
تعداد کلمه ها ۱۳۸۱۲
تعداد کل توکن ۲۰۷۸۶۸۴
تعداد کل کلم ها ۵۵۸۹۰

و در این حالت هم b = 0.66 به دست میاد و رابطه برقرا هست

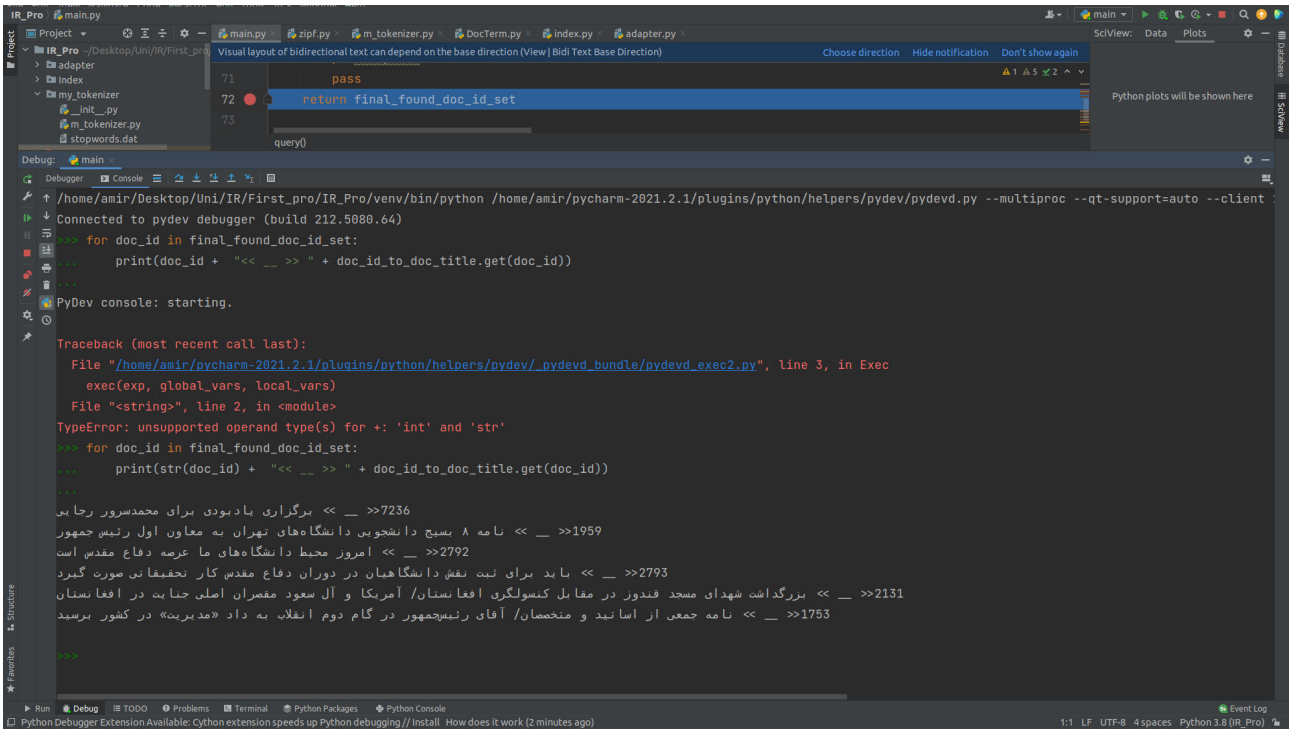
4)
کلماتی که در ریشه خودشان پسوند هایی دارند که بعد از ریشه یابی ب مشکل میخورند و شکل درست آنها ذخیره نمیشود

کلمه ای مانند درخت که به ت ختم شده اما بعد ریشه یابی به صورت درخ ذخیره شده
یا فعلی مثل نهادن یه هر وجه و زمانی از این مصدر که به نه ختم میشود و با ۹ اشتباه گرفته میشود

5)
الف) پرسمان = دانشگاه

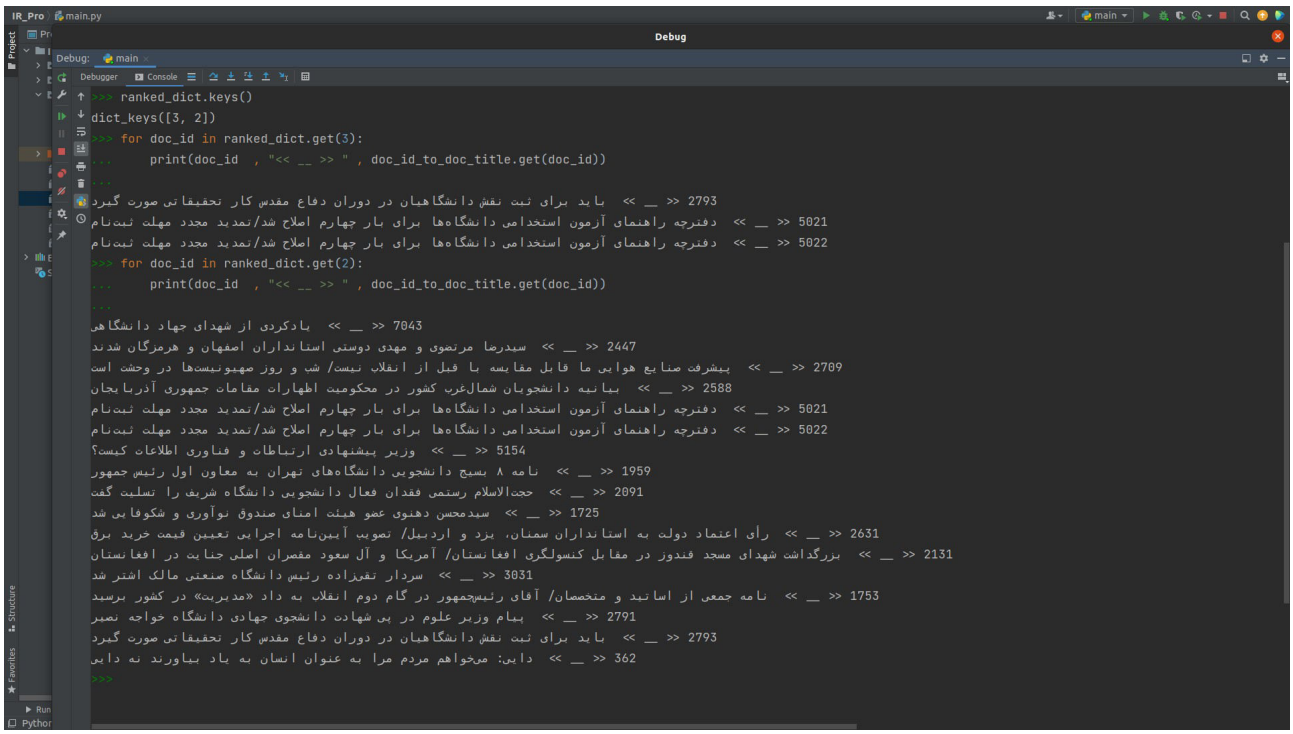


ب) دانشگاه امیرکبیر

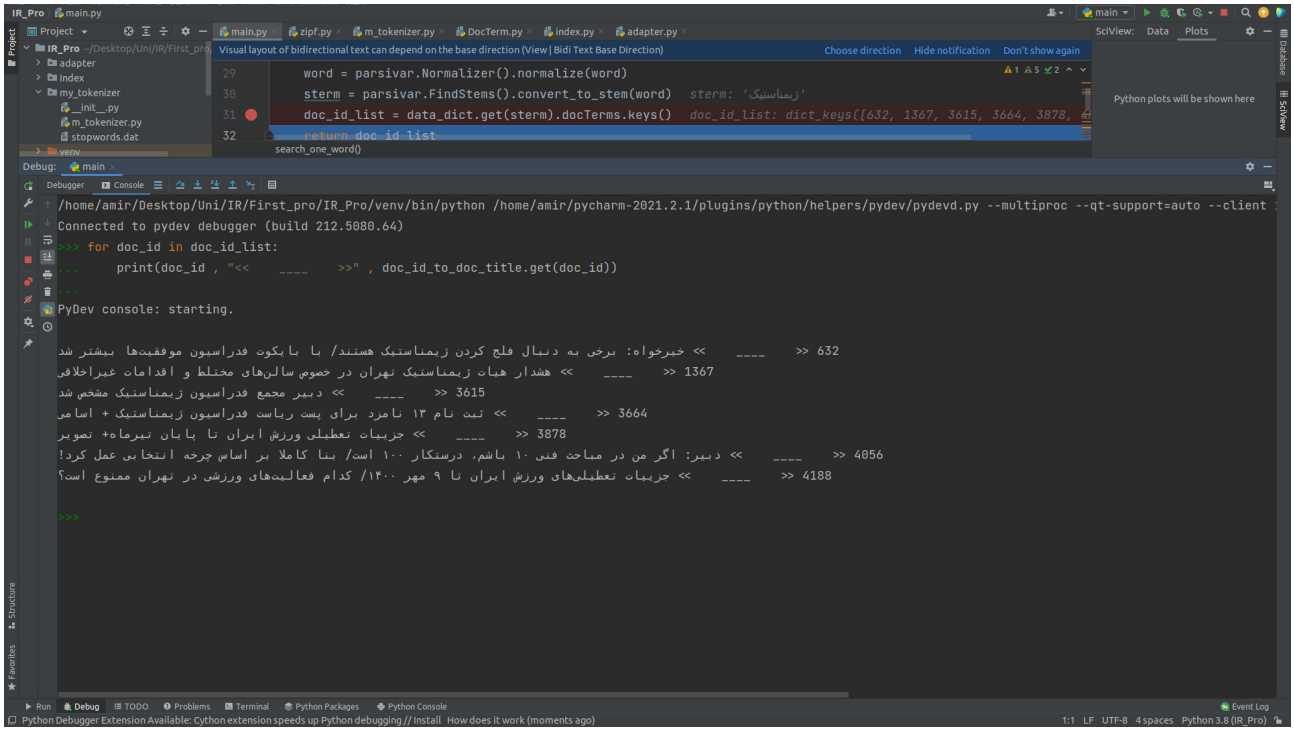


پ) دانشگاه صنعتی امیرکبیر

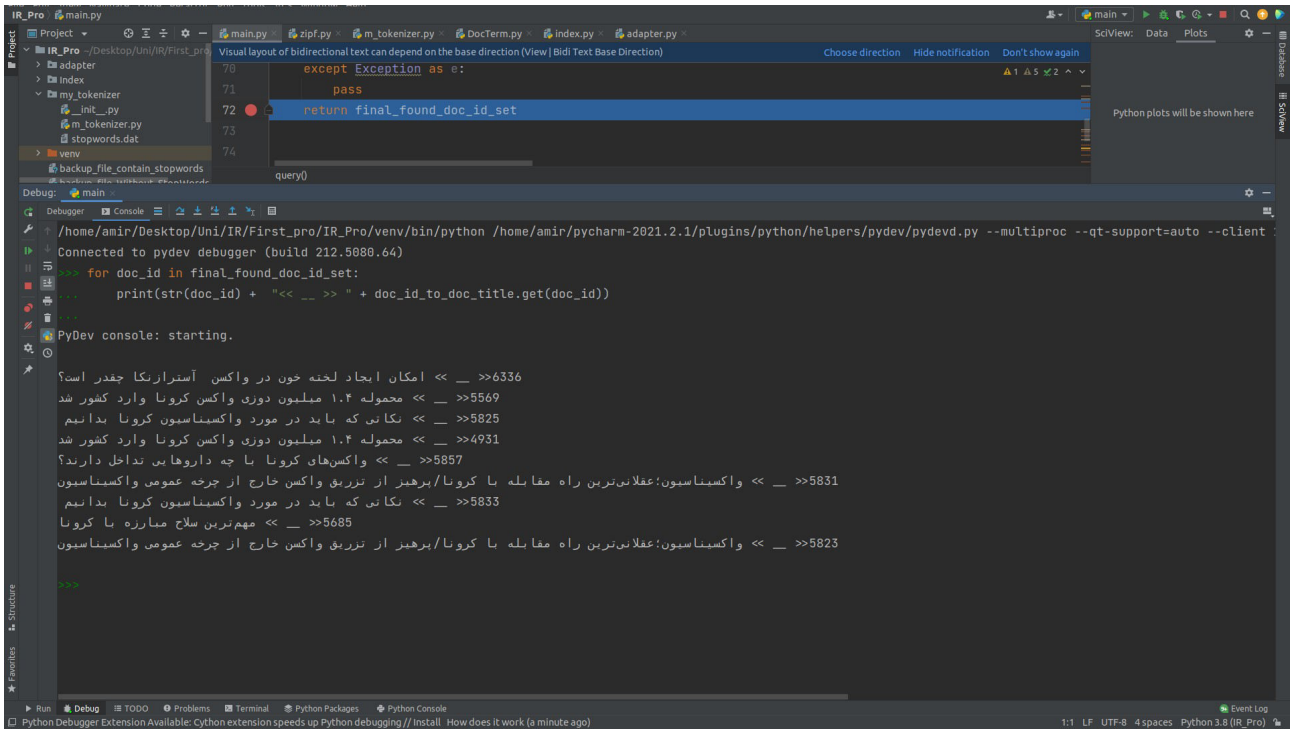
در اینجا داده ها رنکنینگ شدند و به ترتیب بالاترین رنک را دارند چاپ شده اند



ت) پرسمان زیمناستیک



ث) واکسن استرازنکا


except Exception as e:
pass
return final_found_doc_id_set
Python plots will be shown here
/home/amir/Desktop/Uni/IR/First_pro/IR_Pro/venv/bin/python /home/amir/pycharm-2021.2.1/plugins/python/helpers/pydev/pydevd.py --multiproc --qt-support=auto --client
Connected to pydev debugger (build 212.5080.64)
>>> for doc_id in final_found_doc_id_set:
... print(str(doc_id) + "<< __ >> " + doc_id_to_doc_title.get(doc_id))
...
PyDev console: starting.
امکان ایجاد لخته خون در واکسن آسترازنکا چقدر است؟ << __ >>6336
محموله ۱.۴ میلیون دوزی واکسن کرونا وارد کشور شد << __ >>5569
نکاتی که باید در مورد واکسیناسیون کرونا بدانیم << __ >>5825
محموله ۱.۴ میلیون دوزی واکسن کرونا وارد کشور شد << __ >>4931
واکسن‌های کرونا با چه داروهایی تداخل دارند؟ << __ >>5857
واکسیناسیون؛عقلانی‌ترین راه مقابله با کرونا/پرهیز از تزریق واکسن خارج از چرخه عمومی واکسیناسیون << __ >>5831
نکاتی که باید در مورد واکسیناسیون کرونا بدانیم << __ >>5833
مهم‌ترین سلاح مبارزه با کرونا << __ >>5685
واکسیناسیون؛عقلانی‌ترین راه مقابله با کرونا/پرهیز از تزریق واکسن خارج از چرخه عمومی واکسیناسیون << __ >>5823
>>>