SCIENTOPY库

汇报人：李星琛

2020年5月12日

目录

1/ScientoPy库背景及简介

2/ScientoPy库的文件目录及架构

3/ScientoPy库的参数和功能

4/ScientoPy库的核心代码分析

5/ScientoPy库的运行截图（演示）

1/ScientoPy库背景及简介

ScientoPy简介

文献计量分析是由不同工具支持的新兴研究领域，其中一些工具基于网络表示或主题分析。

尽管工具开发了多年，但仍需要支持合并来自不同来源的信息，并加强纵向时态分析，这是主题发展趋势的一部分。

ScientoPy是用于科学出版物中主题趋势分析的科学计量工具，该工具有助于合并Scopus和Clarivate Web of Science的问题，提取并表示分析主题的h-index，并使用四种不同的可视化选项为作者，机构，通配符和趋势主题提供时间分析的可能性。

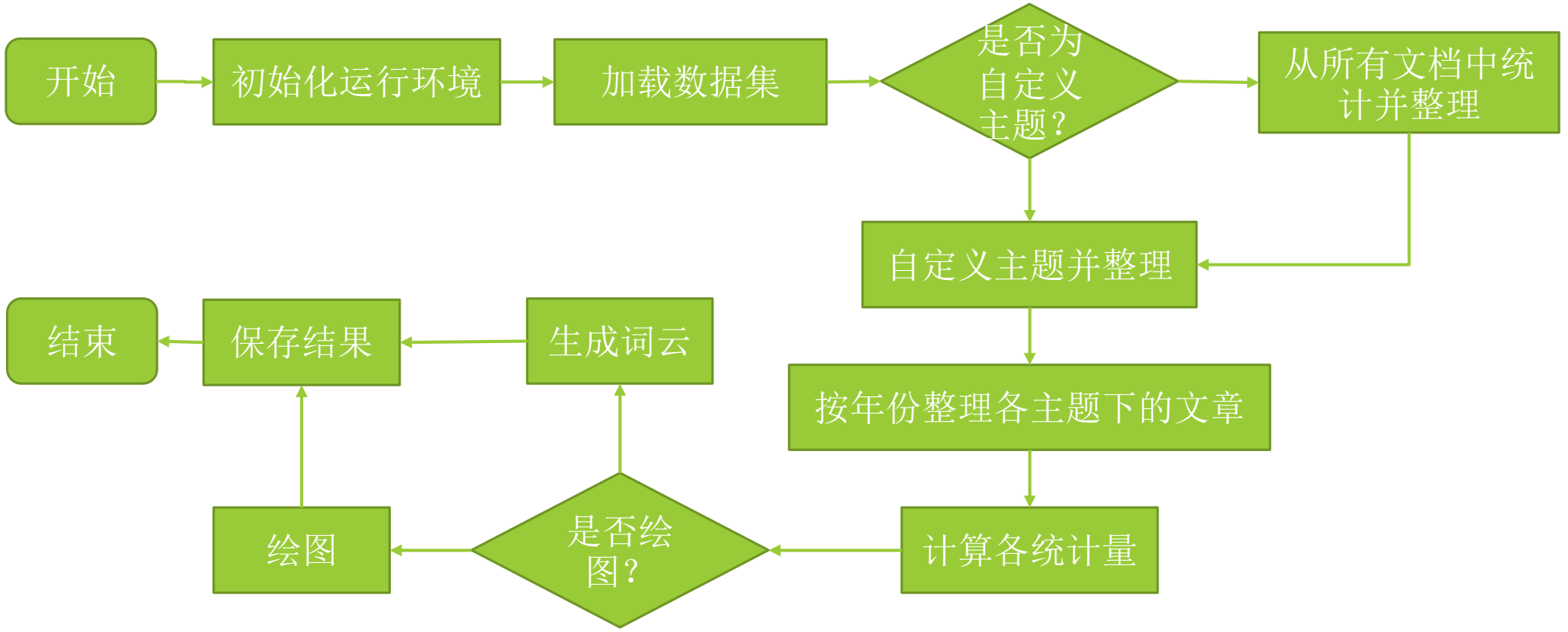
该工具可以在不同的新兴领域进行未来的文献计量分析。

2/ScientoPy库的文件目录及架构

ScientoPy文件目录

dataInExample	在Scopus和WoS上搜索Internet of things和Gateway的示例数据集		
dataPre	预处理结果的输出文件夹以及scientoPy的输入文件夹	papersPreprocessed.tsv	预处理的论文数据
		PreprocessedBrief.tsv	预处理结果的简要表
graphs	预处理和scientoPy的图形输出文件夹		
Manual	PDF手册		
results	scientoPy的输出文件夹	AuthorKeywords.tsv	scientoPy的输出文件，包括文档，总数平均增长率，平均论文数，h指数等
		AuthorKeywords_extended.tsv	所选字段的scientoPy的输出文件，显示与每个主题相关的文档的热门或自定义主题
		papersPreprocessed.tsv	包含上次使用的scientoPy的输出文件，使用选项-r或--previousResults时用作scientoPy的输入

ScientoPy流程图



字段描述规范

Criteron	Description
author	Authors last name and first name initial
sourceTitle	Publication or journal name
subject	Research areas, only from WoS documents
authorKeywords	Author keywords
indexKeywords	Keywords generated by the index, from WoS {Keyword Plus}, and from Scopus {Indexed keywords}
bothKeywords	AuthorKeywords and indexKeywords are used for this search
abstract	Document abstract, for use with pre-defined topics and asterisk wild-card
documentType	Type of document
dataBase	Database where the document was extracted (WoS or Scopus)
country	Country extracted from authors affiliations
institution	Institution extracted from authors affiliations
institutionWithCountry	Institution with country extracted from authors affiliations

图表描述规范

Graph type	Argument	Description
Time line	`-g time_line`	Graphs the number of documents of each topic vs the publication year
Horizontal bars	`-g bar`	Graphs the total number of documents of each topic in horizontal bars
Horizontal bars trends	`-g bar`	Graphs the total number of documents of each topic in horizontal bars, with the percentage of document published in the last years
Evolution	`-g evolution`	Graphs two plots, one with the accumulative number of documents vs the publication year, and other with the average papers per year vs the percentage of documents in the last years
Word cloud	`-g word_cloud`	Generate a word cloud based on the topic total number of publications

3/ScientoPy库的参数和功能

参数及环境变量概览

usage: python [option] ... [-c cmd | -m mod | file | -] [arg] ...

Options and arguments (and corresponding environment variables):

-b : issue warnings about str(bytes_instance), str(bytearray_instance)
and comparing bytes/bytearray with str. (-bb: issue errors)

-B : don't write .pyc files on import; also PYTHONDONTWRITEBYTECODE=x

-c cmd : program passed in as string (terminates option list)

-d : debug output from parser; also PYTHONDEBUG=x

-E : ignore PYTHON* environment variables (such as PYTHONPATH)

-h : print this help message and exit (also --help)

参数及环境变量概览

-i : inspect interactively after running script; forces a prompt even

if stdin does not appear to be a terminal; also PYTHONINSPECT=x

-I : isolate Python from the user's environment (implies -E and -s)

-m mod : run library module as a script (terminates option list)

-O : remove assert and __debug__-dependent statements; add .opt-1 before.pyc extension; also PYTHONOPTIMIZE=x

-OO : do -O changes and also discard docstrings; add .opt-2 before.pyc extension

-q : don't print version and copyright messages on interactive startup

-s : don't add user site directory to sys.path; also PYTHONNOUSERSITE

参数及环境变量概览

-S : don't imply 'import site' on initialization

-u : force the stdout and stderr streams to be unbuffered;

this option has no effect on stdin; also PYTHONUNBUFFERED=x

-v : verbose (trace import statements); also PYTHONVERBOSE=x

can be supplied multiple times to increase verbosity

-V : print the Python version number and exit (also --version)

when given twice, print more information about the build

-W arg : warning control; arg is action:message:category:module:lineno

also PYTHONWARNINGS=arg

参数及环境变量概览

-x : skip first line of source, allowing use of non-Unix forms of #!cmd

-X opt : set implementation-specific option

--check-hash-based-pycs always|default|never:

control how Python invalidates hash-based .pyc files

file : program read from script file

- : program read from stdin (default; interactive mode if a tty)

arg ...: arguments passed to program in sys.argv[1:]

主要功能概览

-h, --help show this help message and exit

-c {author,sourceTitle,subject,authorKeywords,indexKeywords,abstract,bothKeywords,documentType,dataBase,country,institution,institutionWithCountry}

Select the criterion to analyze the topics

-g {bar_trends,bar,time_line,evolution,word_cloud}, --graphType {bar_trends,bar,time_line,evolution,word_cloud}

Select the graph type to plot

-l LENGTH, --length LENGTH

Length of the top topics to analyze, default 10

主要功能概览

-s SKIPFIRST, --skipFirst SKIPFIRST

To filter the first top elements. Ex: to filter the first 2 elements on the list use -s 2

-t TOPICS, --topics TOPICS

Specific topics to analyze according to critera, group topics with "," and divide the topics with ";" Ex: authorKeywords -t "internet of things, iot; bluetooth" asterisk wildcard ex: authorKeywords -t "device*"

主要功能概览

--startYear STARTYEAR

Start year to limit the search, default: 1990

--endYear ENDYEAR End year year to limit the search, default: 2019

--savePlot SAVEPLOT Save plot to a file. Ex: --savePlot "topKeywords.eps"

--pYear To present the results in percentage per year instead of documents per year

--yLog Plot Y axes in log scale

--noPlot Do not plot the results, use for large amount of topics

主要功能概览

--agrForGraph To use average growth rate (AGR) instead average documents per year (ADY) in parametric and parametric 2 graphs

--wordCloudMask WORDCLOUDMASK

PNG mask image to use for wordCloud

--windowWidth WINDOWWIDTH

Window width in years for average growth rate and average documents per year, minimum 1

-r, --previousResults

Analyze based on the previous results

主要功能概览

--onlyFirst Only look in the first elemet of the topic, for example to analyze only the first author name, country or institution

--graphTitle GRAPHTITLE

To put a title in the output graph

--plotWidth PLOTWIDTH

Set the plot width size in inches, default: 6.4

--plotHeight PLOTHEIGHT

Set the plot heigth size in inches, default: 4.8

主要功能概览

--trend Get and graph the top trending topics, with the highest average growth rate

-f FILTER, --filter FILTER

Filter to be applied on a sub topic.Example to extract instituions from United States: scientoPy.py institutionWithCountry -f "United States"

4/ScientoPy库的核心代码分析

预处理

```
# Read files from the dataInFolder
for file in os.listdir(os.path.join(args.dataInFolder, '')):
    if file.endswith(".csv") or file.endswith(".txt"):
        print("Reading file: %s" % (os.path.join(args.dataInFolder, '') + file))
        ifile = open(os.path.join(args.dataInFolder, '') + file, "r", encoding='utf-8')
        paperUtils.openFileToDict(ifile, paperDict)

# If not documents found
if (globalVar.loadedPapers == 0):
    print("ERROR: 0 documents found from " + os.path.join(args.dataInFolder, ''))
    print("")
    return 0

# Removing duplicates
if not args.noRemDupl:
    paperDict = paperUtils.removeDuplicates(paperDict, logWriter, preProcessBrief)

# if not remove duplicates
else:
    preProcessBrief["totalAfterRemDupl"] = preProcessBrief["papersAfterRemOmitted"]
    preProcessBrief["removedPapersScopus"] = 0
    preProcessBrief["removedPapersWoS"] = 0
    preProcessBrief["papersScopus"] = preProcessBrief["loadedPapersScopus"]
    preProcessBrief["papersWoS"] = preProcessBrief["loadedPapersWoS"]
```

预处理

```
# To avoid by zero division
if preProcessBrief["totalAfterRemDupl"] > 0:
    percentagePapersWos = 100.0 * preProcessBrief["papersWoS"] / preProcessBrief["totalAfterRemDupl"]
    percentagePapersScopus = 100.0 * preProcessBrief["papersScopus"] / preProcessBrief["totalAfterRemDupl"]
else:
    percentagePapersWos = 0
    percentagePapersScopus = 0

# Saving graph
plt.tight_layout()

if args.savePlot == "":
    if self.fromGui:
        plt.show(block=False)
    else:
        plt.show(block=True)
else:
    plt.savefig(os.path.join(globalVar.GRAPHS_OUT_FOLDER, args.savePlot),
                bbox_inches='tight', pad_inches=0.01)
    print("Plot saved on: " + os.path.join(globalVar.GRAPHS_OUT_FOLDER, args.savePlot))

if args.savePlot == "":
    if self.fromGui:
        plt.show()
```

载入数据集

```
# Open the dataset only if not loaded in papersDict
if loadDataSet:
    self.papersDict = []
    self.lastPreviousResults = args.previousResults
    # Open the storage database and add to sel.fpapersDict
    if not os.path.isfile(INPUT_FILE):
        print("ERROR: %s file not found" % INPUT_FILE)
        print("Make sure that you have run the preprocess step before run scientoPy")
        exit()

    ifile = open(INPUT_FILE, "r", encoding='utf-8')
    print("Reading file: %s" % (INPUT_FILE))
    paperUtils.openFileToDict(ifile, self.papersDict)
    ifile.close()

    print("Scopus papers: %s" % globalVar.papersScopus)
    print("WoS papers: %s" % globalVar.papersWoS)
    print("Omited papers: %s" % globalVar.omitedPapers)
    print("Total papers: %s" % len(self.papersDict))
```

寻找核心主题

```
# Find the top topics
else:
    print("Finding the top topics...")

    topicDic = {}

    # For each paper, get the full topicDic
    for paper in papersDictInside:

        # For each item in paper criteria
        for item in paper[args.criterion].split(";"):
            # Strip paper item and upper case
            item = item.strip()
            item = item.upper()

            # If paper item empty continue
            if item == "":
                continue
```

寻找核心主题

```
# If filter sub topic, omit items outside that do not match with the subtopic
if filterSubTopic != "" and len(item.split(",")) >= 2:
    if (item.split(",")[1].strip().upper() != filterSubTopic.upper()):
        continue

# If topic already in topicDic
if item in topicDic:
    topicDic[item] += 1
# If topic is not in topicDic, create this in topicDic
else:
    topicDic[item] = 1

# If onlyFirst, only keep the firt processesing
if args.onlyFirst:
    break
```

寻找核心主题

```
    # If trending analysis, the top topic list to analyse is bigger
    if args.trend:
        topicListLength = globalVar.TOP_TREND_SIZE
        startList = 0
    else:
        topicListLength = args.length
        startList = args.skipFirst

    # Get the top topics by the topDic count
    topTopcis = sorted(topicDic.items(),
                       key=lambda x: -x[1])[startList:(startList + topicListLength)]

    # Put the topTopics in topic List
    for topic in topTopcis:
        topicList.append([topic[0]])

    if len(topicList) == 0:
        print("\nFINISHED : There is not results with your inputs criteria or filter")
        del papersDictInside
        return

# print("Topic list:")
# print(topicList)
```

提取总数

```
# Extract accumulative
for topicItem in topicResults:
    citedAccumValue = 0
    papersAccumValue = 0
    for i in range(0, len(topicItem["CitedByCountAccum"])):
        citedAccumValue += topicItem["CitedByCount"][i]
        topicItem["CitedByCountAccum"][i] = citedAccumValue

        papersAccumValue += topicItem["PapersCount"][i]
        topicItem["PapersCountAccum"][i] = papersAccumValue
```

提取AGR

```
# Extract the Average Growth Rate (AGR)
for topicItem in topicResults:
    # Calculate rates
    pastCount = 0
    # Per year with papers count data
    for i in range(0, len(topicItem["PapersCount"])):
        topicItem["PapersCountRate"][i] = topicItem["PapersCount"][i] - pastCount
        pastCount = topicItem["PapersCount"][i]

    # Calculate AGR from rates
    endYearIndex = len(topicItem["year"]) - 1
    startYearIndex = endYearIndex - (args.windowWidth - 1)

    topicItem["agr"] = \
        round(np.mean(topicItem["PapersCountRate"][startYearIndex: endYearIndex + 1]), 1)
```

提取ADY

```
# Extract the Average Documents per Year (ADY)
for topicItem in topicResults:

    # Calculate ADY from rates
    endYearIndex = len(topicItem["year"]) - 1
    startYearIndex = endYearIndex - (args.windowWidth - 1)

    topicItem["AverageDocPerYear"] = \
        round(np.mean(topicItem["PapersCount"][startYearIndex: endYearIndex + 1]), 1)

    topicItem["PapersInLastYears"] = \
        np.sum(topicItem["PapersCount"][startYearIndex: endYearIndex + 1])

    if topicItem["PapersTotal"] > 0:
        topicItem["PerInLastYears"] = \
            round(100 * topicItem["PapersInLastYears"] / topicItem["PapersTotal"], 1)
```

计算h指数

```
# Calculate h index per topic
for topicItem in topicResults:

    # print("\n" + topicName)

    # Sort papers by cited by count
    papersIn = topicItem["papers"]
    papersIn = sorted(papersIn, key=lambda x: int(x["citedBy"]), reverse=True)

    count = 1
    hIndex = 0
    for paper in papersIn:
        # print(str(count) + ". " + paper["citedBy"])
        if int(paper["citedBy"]) >= count:
            hIndex = count
        count += 1
        # print("hIndex: " + str(hIndex))
        topicItem["hIndex"] = hIndex
```

5/ScientoPy库的运行截图（演示）

预处理

python preProcess.py dataInExample

-h, --help show this help message an

--noRemDupl To do not remove the d

--savePlot SAVEPLOT Save the pre process

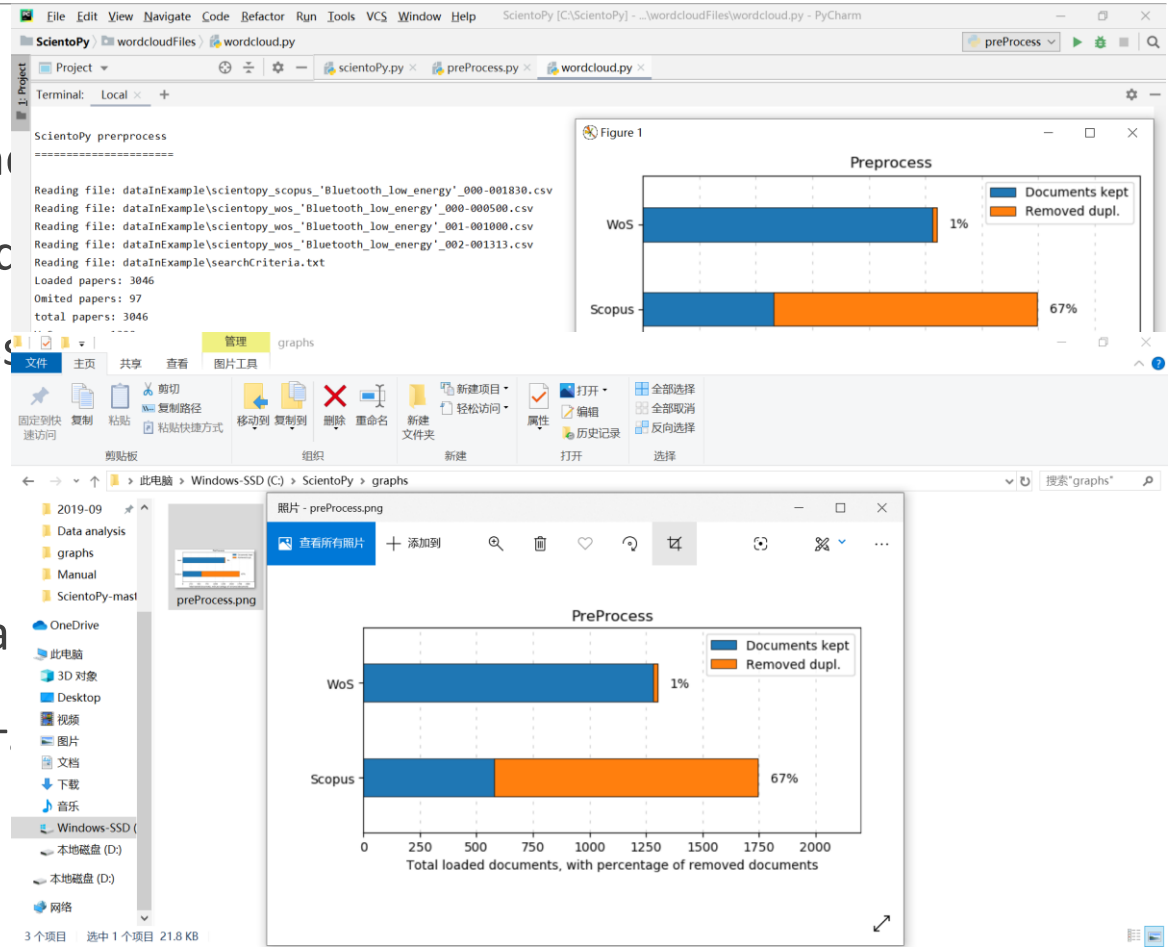
"preProcessed.eps"

--graphTitle GRAPHTITLE

To put a title in the output gra

E.g. python preProcess.py dataInExample --

<C:\ScientoPy\graphs>

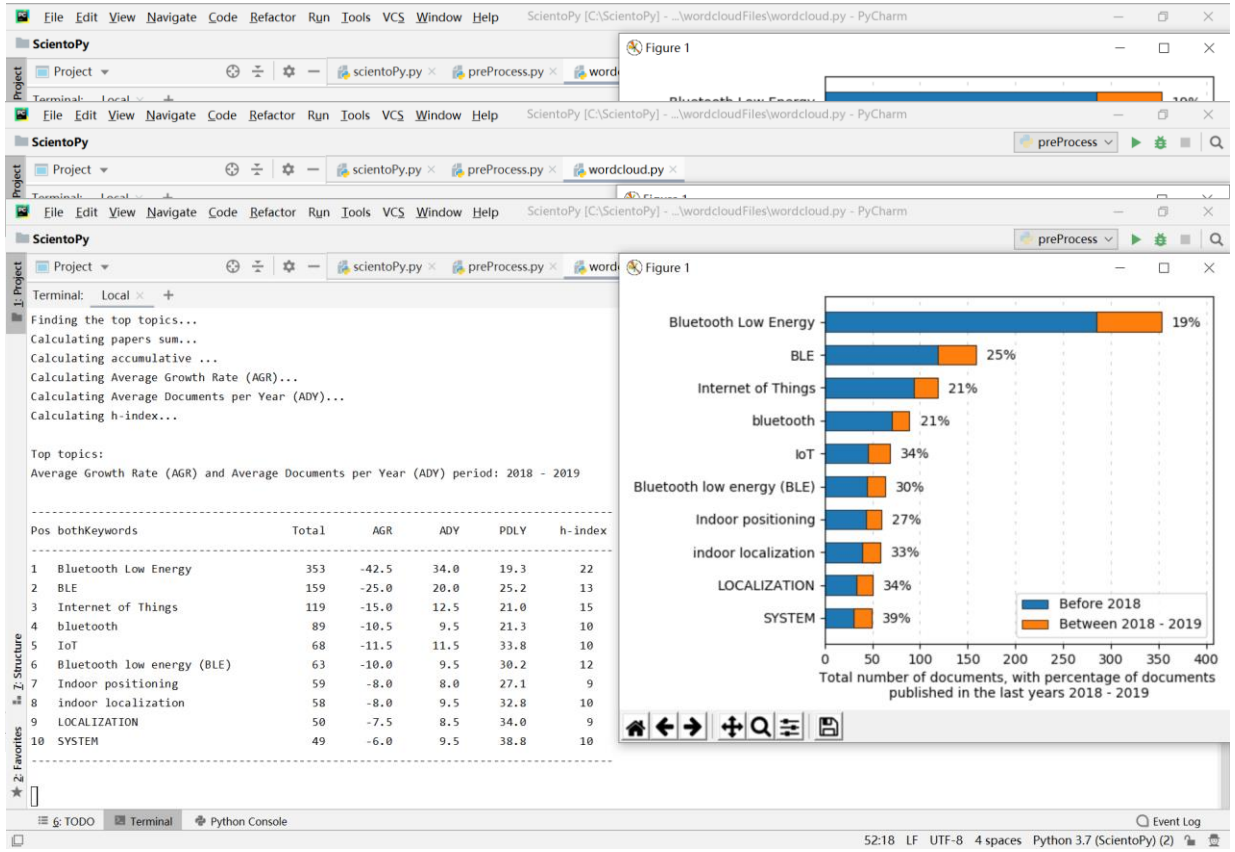


抽取关键词

python scientoPy.py -c authorKeywords

python scientoPy.py -c indexKeywords

python scientoPy.py -c bothKeywords

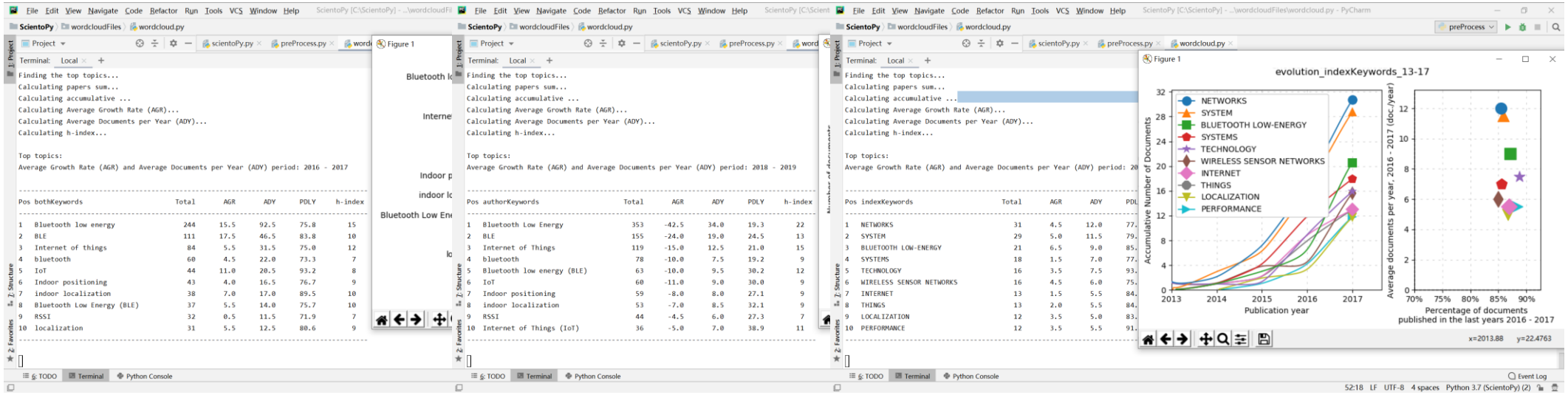


抽取关键词

E.g. python scientoPy.py -c bothKeywords --startYear 2016 --endYear 2017

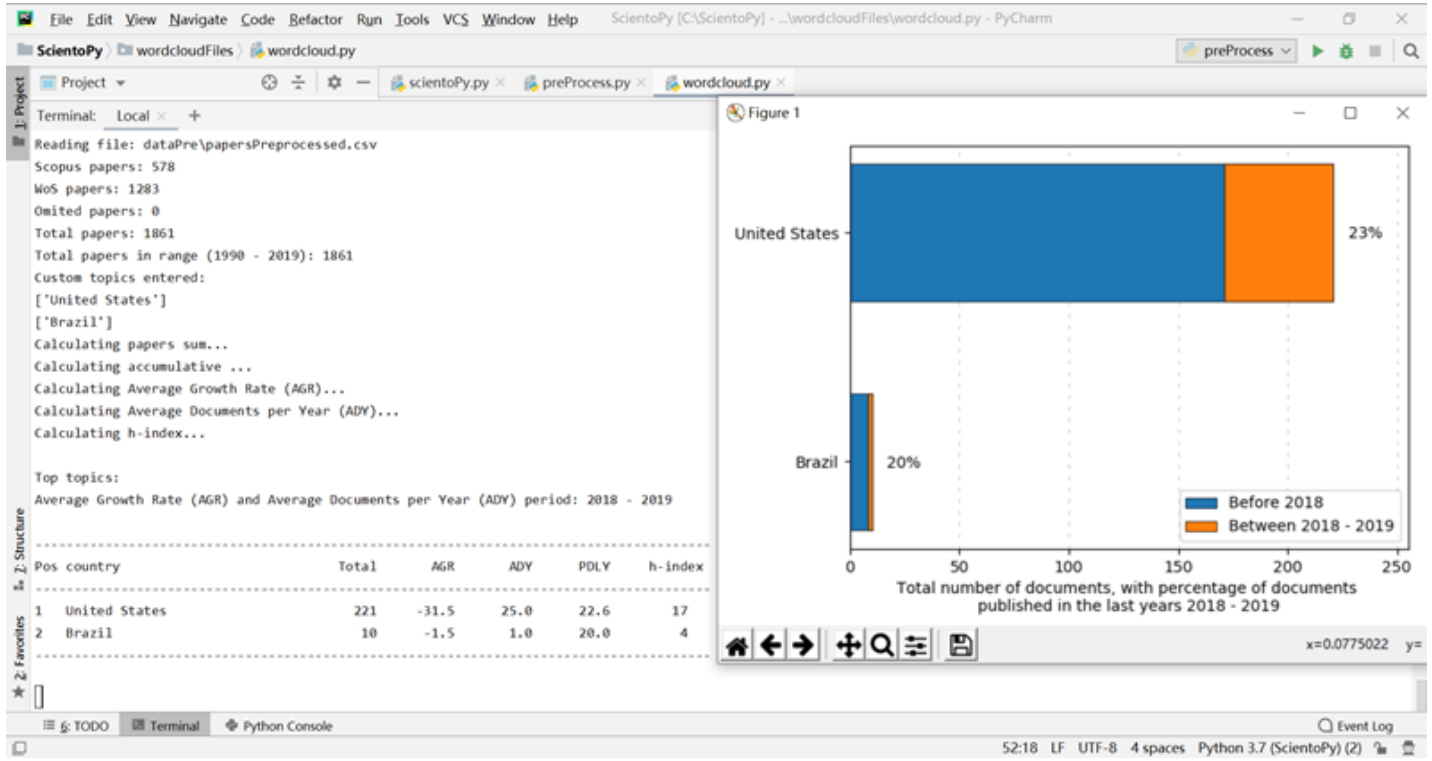
python scientoPy.py -c authorKeywords -g time_line

python scientoPy.py -c indexKeywords --startYear 2013 --endYear 2017 -g evolution --graphTitle evolution_indexKeywords_13-17



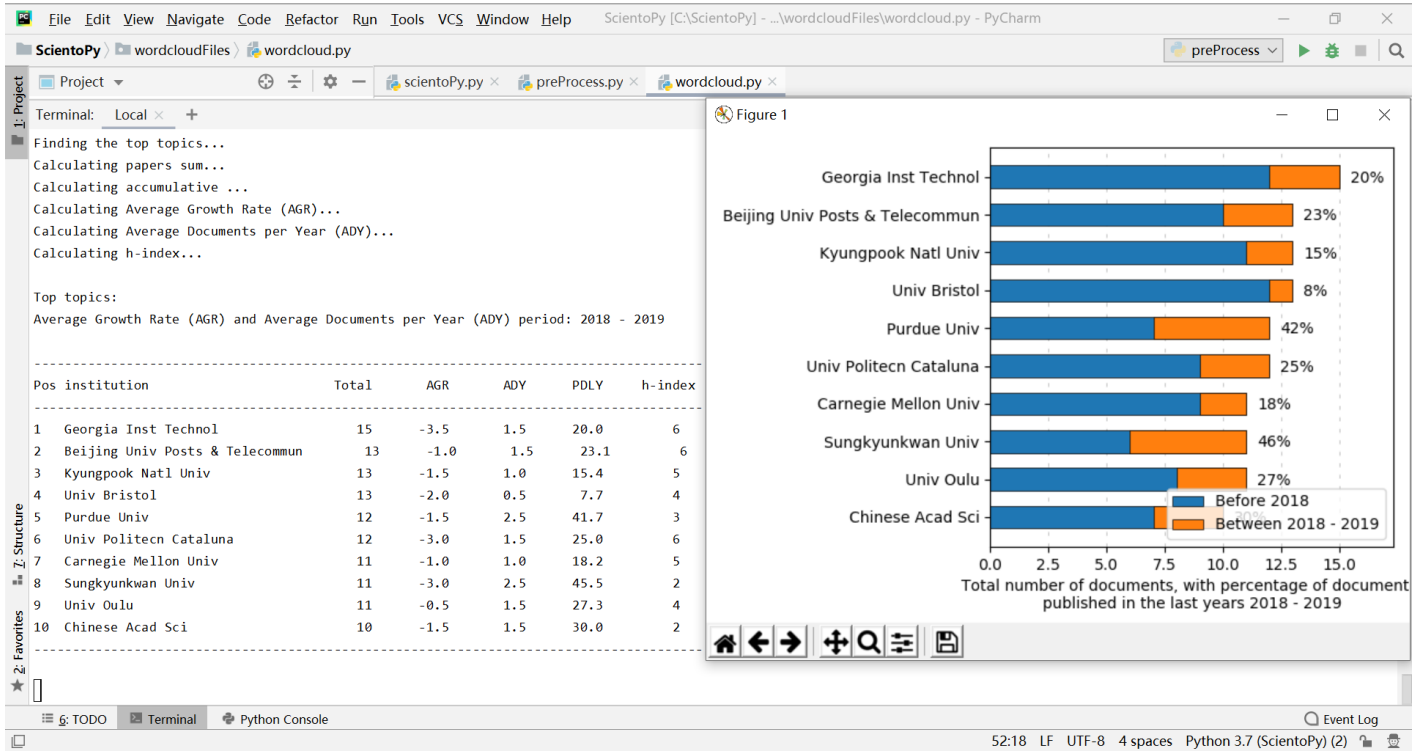
分析自定义主题-国家

python scientoPy.py -c country -t "United States; Brazil"



分析自定义主题-机构

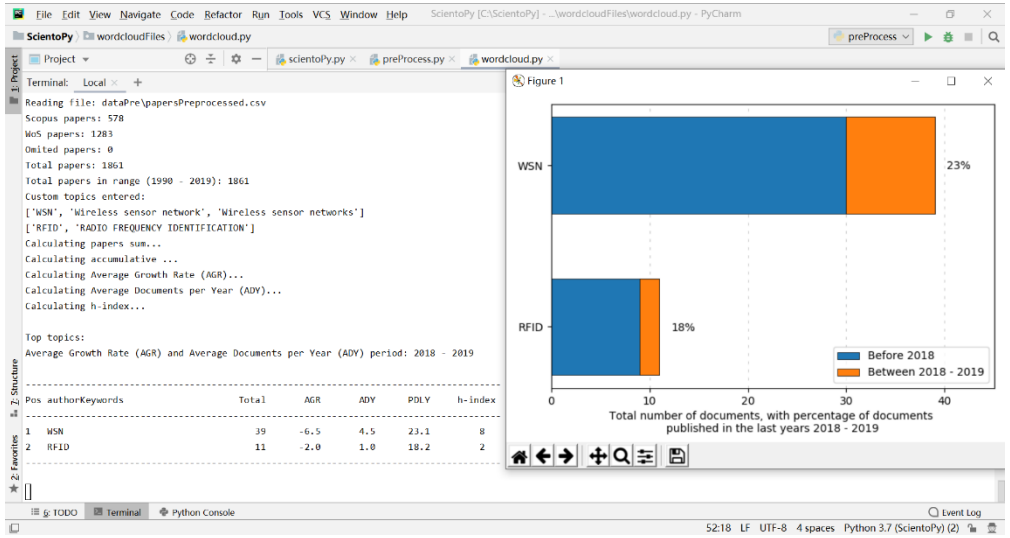
python scientoPy.py -c institution

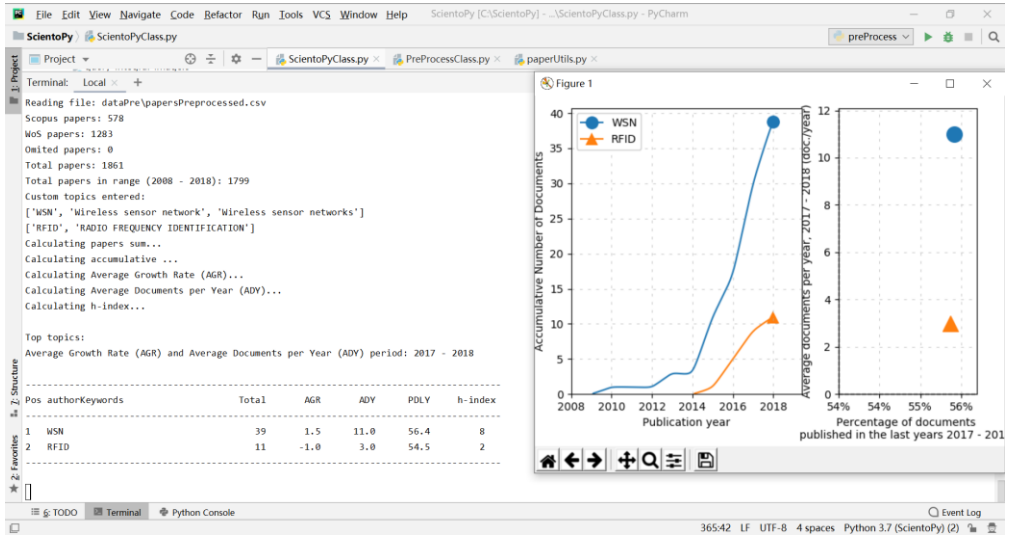


处理同义词

python scientoPy.py -c authorKeywords –t”WSN, Wireless sensor network, Wireless sensor networks; RFID, RADIO FREQUENCY IDENTIFICATION”

python scientoPy.py -c authorKeywords -t "WSN, Wireless sensor network, Wireless sensor networks; RFID, RADIO FREQUENCY IDENTIFICATION" -g evolution --startYear 2008 --endYear 2018

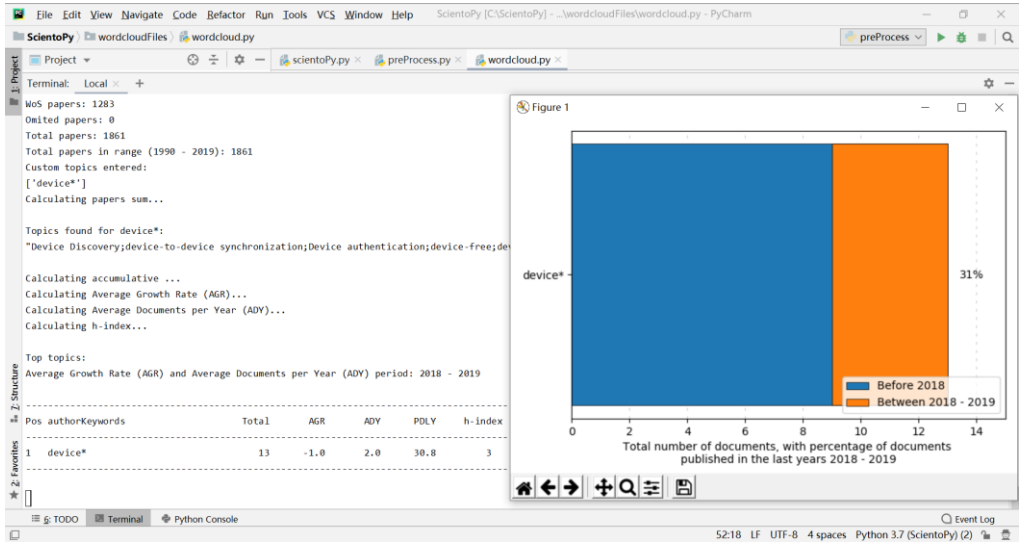


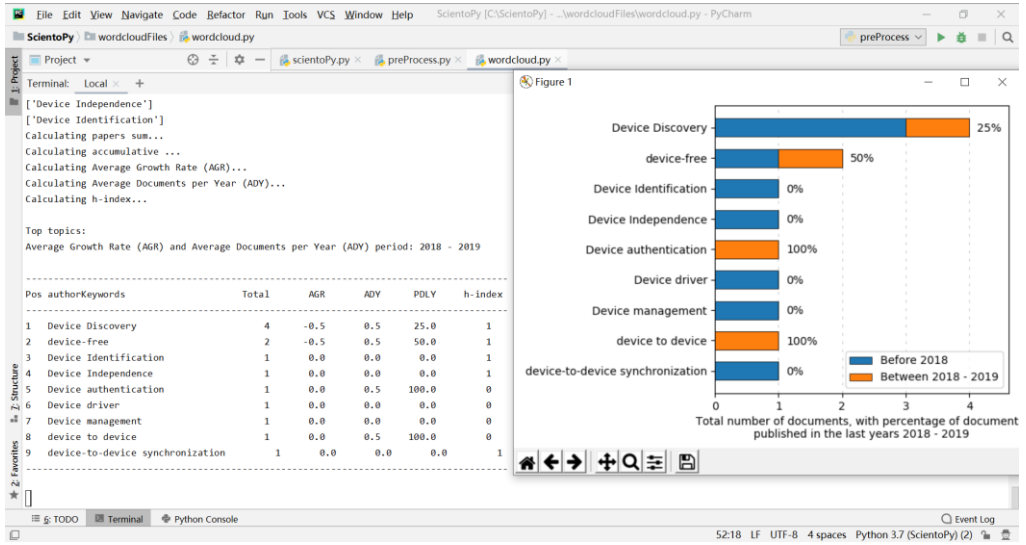


利用*通配符

Python scientoPy.py -c authorKeywords -t “device*”

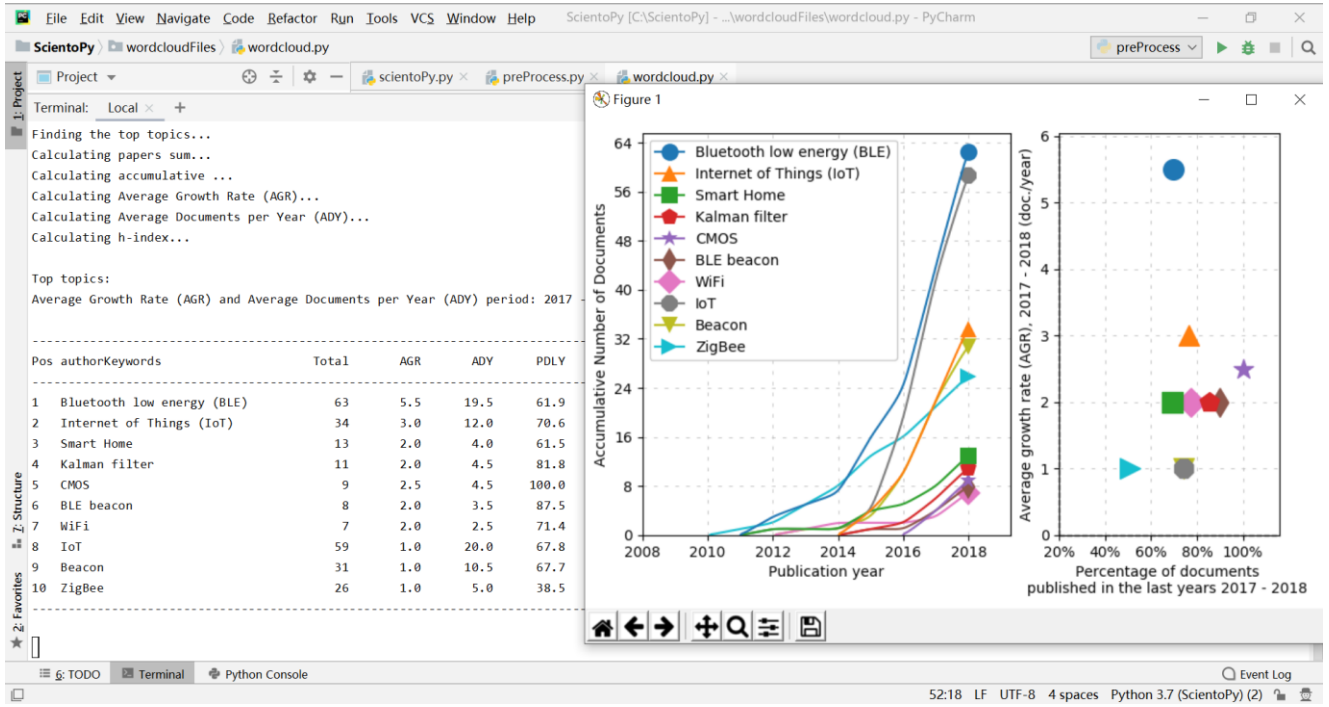
python scientoPy.py -c authorKeywords -t “Device Discovery;device-to-device synchronization;Device authentication;device-free; device to device;Device driver;Device management;Device Independence;Device Identification”





关键词演变趋势

python scientoPy.py -c authorKeywords --trend --startYear 2008 --endYear 2018 --windowWidth 2 --agrForGraph -g evolution

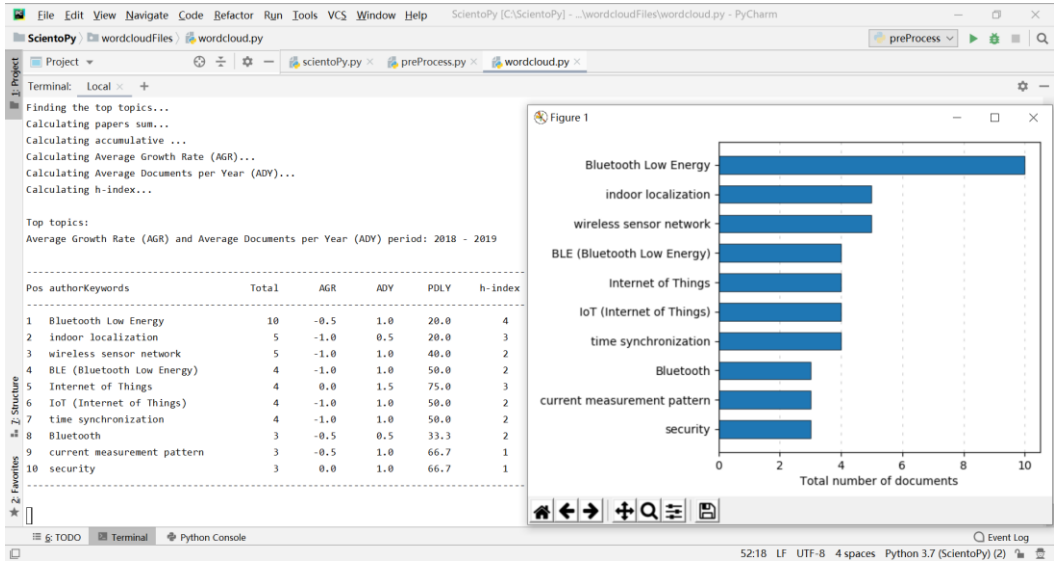


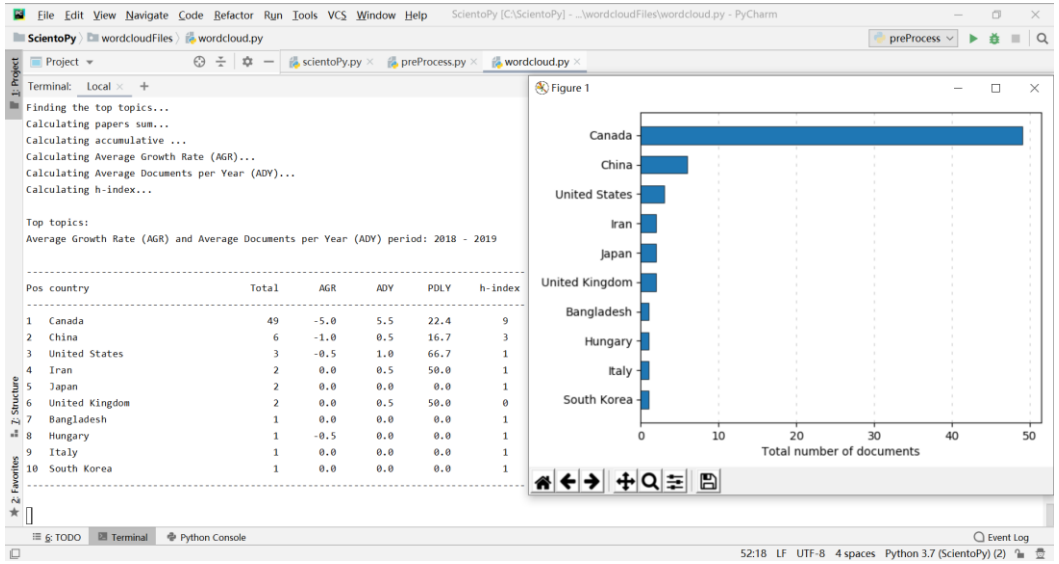
基于某个结果的分析

python scientoPy.py -c country -t "Canada" --noPlot

python scientoPy.py -c authorKeywords -r -g bar

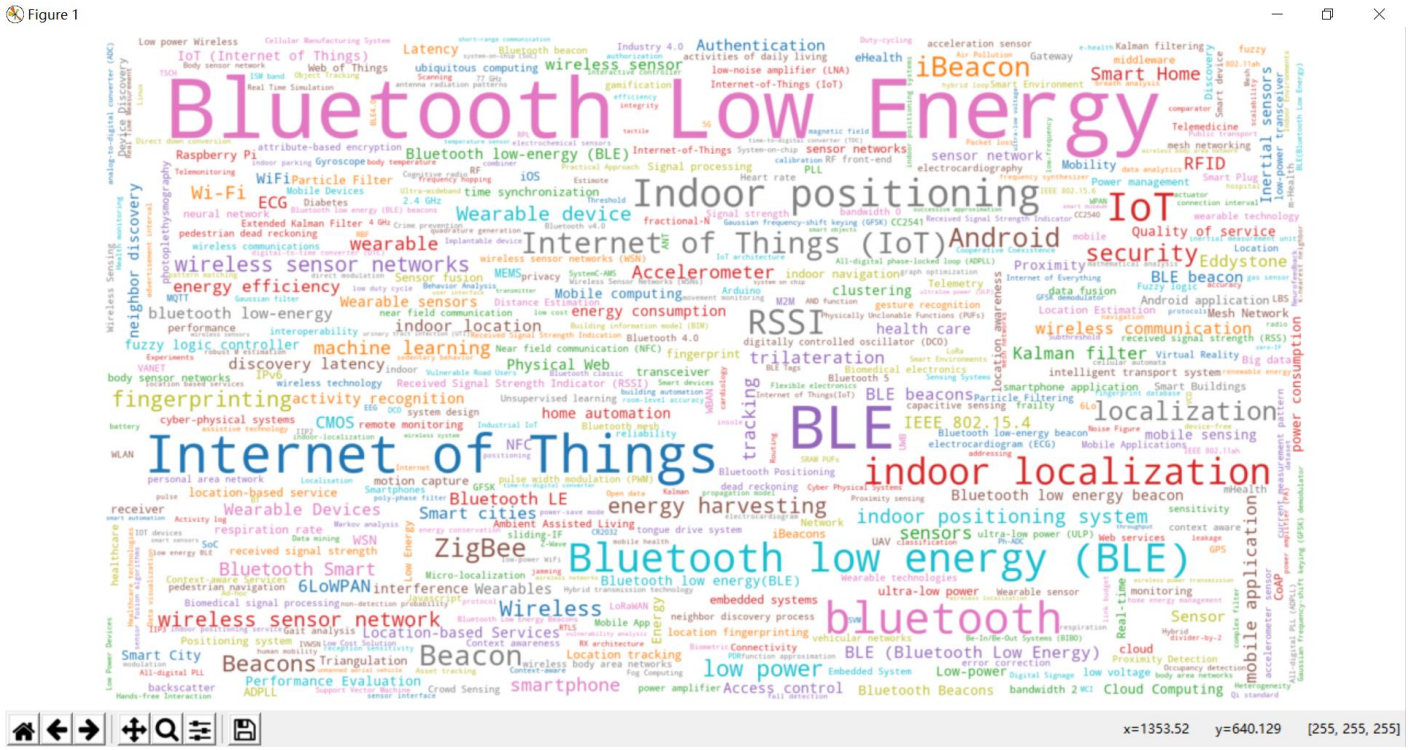
python scientoPy.py -c country -r -g bar





生成词云

python scientoPy.py -c authorKeywords --startYear 2008 --endYear 2018 -l 500 -g word_cloud



Thanks for watching!

汇报人：李星琛

2020年5月12日