ScientoPy v 1.4.0, Installation and User Manual

Juan Pablo Ruiz Rosero
jpabloruiz@unicauca.edu.co

Contents

1 Installation

1. For Windows download and install the Python 3 latest version (for example Python 3.6.5) from: `https://www.python.org/downloads/`.
2. For Debian or Ubuntu run these commands to install Python3:

   ```
   sudo apt-get install python3 python3-tk python3-pip
   ```

3. To use wordCloud in Windows, install Microsoft Visual C++ Redistributable para Visual Studio 2017 according to these instructions: `https://www.scivision.co/python-windows-visual-c++-14-required/`
4. Install the unidecode, numpy, scipy, matplotlib, and wordcloud Python libraries. For Windows, enter in the command line (Windows + R, cmd, and Enter), and run the installation script:

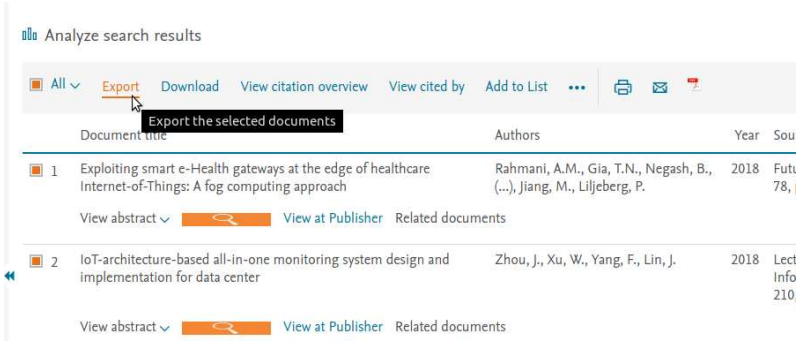
   ```
   python3 -m pip install --user unidecode numpy scipy matplotlib wordcloud
   ```

2 Download the bibliometric dataset

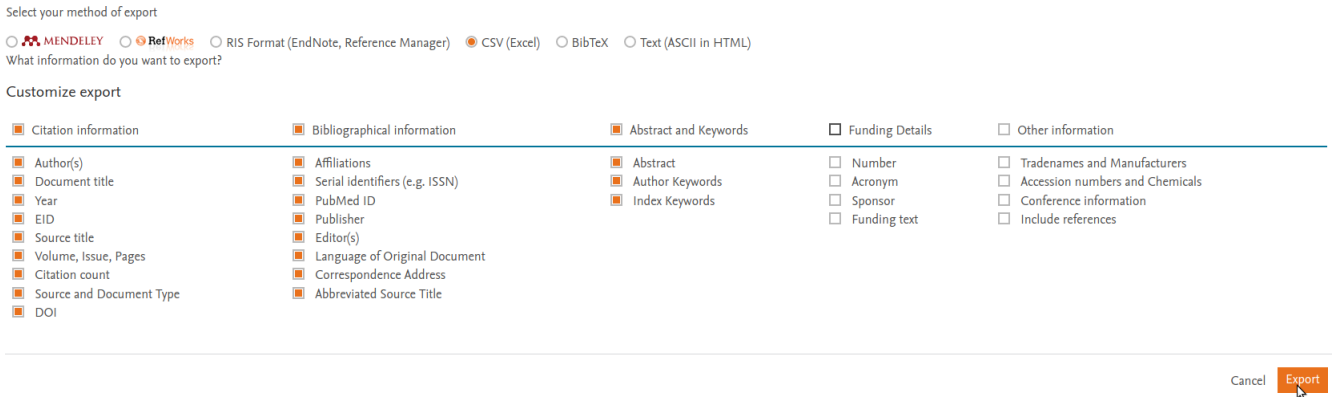
This section describes how to download the proper dataset from Scopus and WoS. Define a search criteria that will be used for Scopus and WoS. For this guide and for the example dataset we are using: "Internet of thing" AND "Gateway"

2.1 Download the dataset from Scopus

1. Make your search with the defined search criteria for Article title, Abstract, Keywords.
2. Select all the results and click on Export:



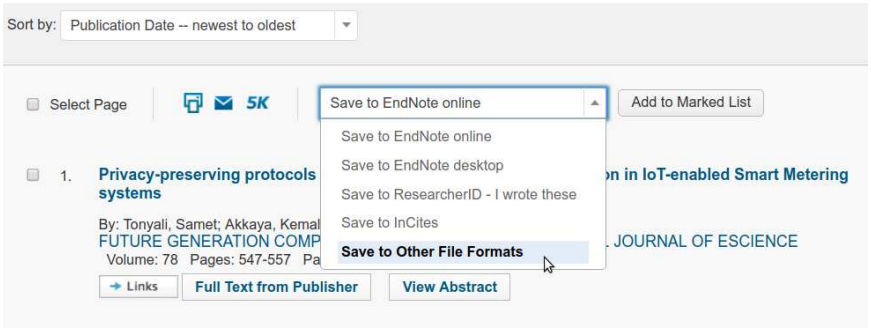
3. Select as method of export **CSV (Excel)**, and select the Customize export **Citation information, Bibliographical information, Abstract and Keywords**, then click on Export:



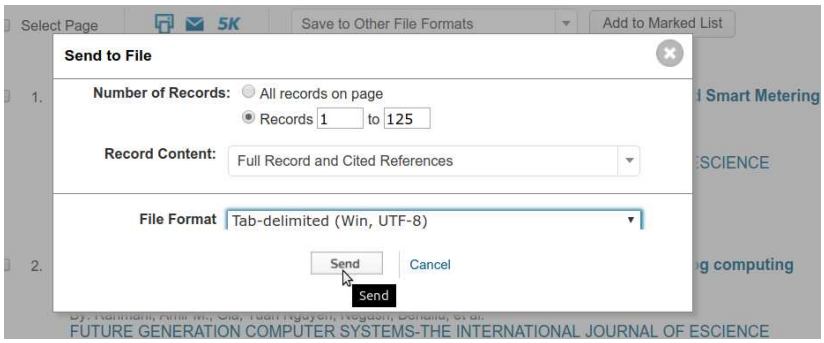
4. Save the file on the folder `/ScientoPy/dataIn`

2.2 Download the dataset from WoS

1. Make your search with the defined search criteria for Topic.
2. Select **Save in Other File Formats**



3. Select the number of records to download, on Record Contented select **Full Record and Cited References**, on File Format select **Tab-delimited (Win, UTF-8)**, and click on Send.



4. Save the file on the folder `/ScientoPy/dataIn`

3 Running the ScientoPy scripts

This section describes the ScientoPy scripts to preprocess and analyze the bibliometric dataset.

3.1 Preprocessing

First we need to preprocess the downloaded dataset. This preprocess merge all the downloaded files from one folder to a single file. Also, this process remove the duplicated files. To preprocess the example dataset ("Bluetooth low energy" downloaded in March 21, 2019, located in dataInExample) run this command inside ScientoPy folder:

```
python3 preProcess.py dataInExample
```

Then, inside the folder `ScientoPy/dataPre` you will find the following files:

- **papersPreprocessed.tsv:** this file contains the information of all papers after the pre-process. This file will be used by the others scripts as the input data.
- **PreprocessedBrief.tsv:** this file briefs the pre-process statics results, such as duplicated papers removed, types of documents, and others.

To find more options of the preprocessing script you can run:

```
python3 preProcess.py -h
```

3.2 Extract the top topics

With this script you can extract the top topics of a selected criterion. The ScientoPy criterion are described on Table 1:

Table 1: ScientoPy criterion description

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

For example, to find the top author keywords you can run this script:

```
python3 scientoPy.py -c authorKeywords
```

This will generate a list with the top 10 topics on the selected criterion (in this case authorKeywords), with the number of documents per topic. Also, this script show the bar graph with the percentage of documents in the last years, and saves the quantitative results on the folder `ScientoPy/results`.

This script have more options like, save the plot on a file, or increase the number of topic results. For more information you can run:

```
python3 scientoPy.py -h
```

3.3 Analyze custom topics inside a criterion

If you want to make an analysis of custom topics, such as the two selected countries papers evolution, you can use the `scientoPy.py` script, with the option `-t`, to specify the topics:

```
python3 scientoPy.py -c country -t "United States; Brazil"
```

You can analyze any topic in any criterion. Put the topics on the `-t` argument. Divide the topics with the `;`. Also, you can integrate two or more topics in one, by dividing it with `,`. This is very useful for abbreviations and plural singulars, for example:

```
python3 scientoPy.py -c authorKeywords -t \
"WSN, Wireless sensor network, Wireless sensor networks; RFID, RADIO FREQUENCY IDENTIFICATION"
```

Note: The command is very long, for that reason the command was divided by `\`. If you have problems in Windows, remove the "änd put the command in one single line.

3.3.1 Asterisk (*) wildcard

You can use the asterisk wildcard to find phrases or words which starts or ends with the letters that you have inserted. For example, if you want to find "device", "devices", and "device integration", enter the following command:

```
python3 scientoPy.py -c authorKeywords -t "device*"
```

ScientoPy will print the topics found for the previous search:

```
Topics found for device*:
"Device Discovery;device-to-device synchronization;Device authentication;device-free;
device to device;Device driver;Device management;Device Independence;Device Identification"
```

You can use this information, to analyze each specific topic found, like this:

```
python3 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"
```

3.3.2 Evolution plot

Also, you can see the results with an evolution plot (add `-g evolution`). This option plots the accumulative documents, average documents per year (ADY), and percentage of documents in the last years, for example:

```
python3 scientoPy.py -c authorKeywords -t \
"WSN, Wireless sensor network, Wireless sensor networks; RFID, RADIO FREQUENCY IDENTIFICATION" \
-g evolution
```

This script have more options like, save the plot on a file, or others. For more information you can run:

```
python3 scientoPy.py -h
```

3.4 Finding trending topics

This script finds the top trending topics based on the higher average growth rate (AGR) over the others. The AGR is calculated on two years periods, using the following Equation (3.4):

$$AGR = \frac{\sum_{i=Y_s}^{Y_e} P_i - P_{i-1}}{(Y_e - Y_s) + 1}, \tag{3.4}$$

where:

AGR = Average growth rate;
Y_s = Start year;
Y_e = End year;
P_i = Number of publications on year i.

To find the top trending topics on author keywords criterion, you can run the following script:

```
python3 scientoPy.py -c authorKeywords --trend --startYear 2008 --endYear 2018 \
--windowWidth 2  --agrForGraph -g evolution
```

This script will find the top 200 topics, then it calculates the AGR for the last 2 years (`--windowWidth 2`). Finally, the 200 top topics are sorted from the highest AGR in the last 2 year period to the lower. The first 3 AGR topics are filtered (they correspond to the keyword Internet of things), and the next 10 topics are garph in a parametric plot.

3.5 Analysis based on the previous results

ScientoPy generates an output file with all the output documents from the last run script. For example if we run the command:

```
python3 scientoPy.py -c country -t "Canada" --noPlot
```

ScientoPy will create a documents output file (`results/papersPreprocessed.tsv`) with all documents that have authors with affiliation in Canada. This output file can be used by ScientoPy to perform an analysis based on this, in that way if we run the following command with the option `-r` or `--previousResults` after the previous one to analyze based on the previous results:

```
python3 scientoPy.py -c authorKeywords -r -g bar
```

we will obtain the top author keywords from papers where the author affiliation correspond to Canada. Also, we can run the following command to know which are the countries that have more common documents with Canada:

```
python3 scientoPy.py -c country -r -g bar
```

Note: the ScientoPy documents output file is only generated when the `-r` or `--previousResults` is not used. In that way, if we run many times a ScientoPy command with this option, the documents output file will not overwritten.

3.6 Output files and directories

After run some ScientoPy commands or after run all the example commands by executing the script `exampleGenerateGraphs.sh` you will find the following folder and files structure:

```
ScientoPy
├── dataInExample
├── dataPre
│   ├── papersPreprocessed.tsv
│   └── PreprocessedBrief.tsv
├── graphs
├── Manual
└── results
    ├── AuthorKeywords.tsv
    ├── AuthorKeywords_extended.tsv
    └── papersPreprocessed.tsv
```

These folders and output files are described bellow:

- **dataInExample:** contains Scopus and WoS example data set for the search criteria "Internet of things" AND "Gateway" downloaded in 27 November 2017. This is the input example for preprocess script.
- **dataPre:** output folder for the preprocess results, and input folder for scientoPy script.
- **papersPreprocessed.tsv:** preprocesed papers data with all input documents merged, filtered, and duplication removed. This is the input file that scientoPy script uses.
- **PreprocessedBrief.tsv:** preproceses brief table that shows the preprocess results related to total papers found per data base, the omitted papers, the duplicated papers count per data base, and the total number of papers per paper type (Conference paper, article, review...)
- **graphs:** graphs output folder for preprocess and scientoPy scripts
- **Manual:** folder with the pdf manual and example paper with scientoPy commands highlighted used for graph and tables generation.
- **results:** output folder for scientoPy result output files
- **AuthorKeywords.tsv:** scientoPy output file for the selected criterion (in this case authorKeywords) that shows the top topics or the custom topics with the total number of documents, the Average Growth Rate (AGR), the Average Documents per Year (ADY), the h-index, and the documents per each year.
- **AuthorKeywords_extended.tsv:** scientoPy output file for the selected criterion (in this case authorKeywords) that show the top or custom topics with the documents related to each one.
- **papersPreprocessed.tsv:** inside the results folder, this file contains the output papers from the last scientoPy used script. This is used as an input for scientoPy script when it use the option `-r` or `--previousResults`

4 ScientoPy graph types

ScientoPy has 5 different ways to graph the results described on Table 2.

Table 2: ScientoPy output graphs types

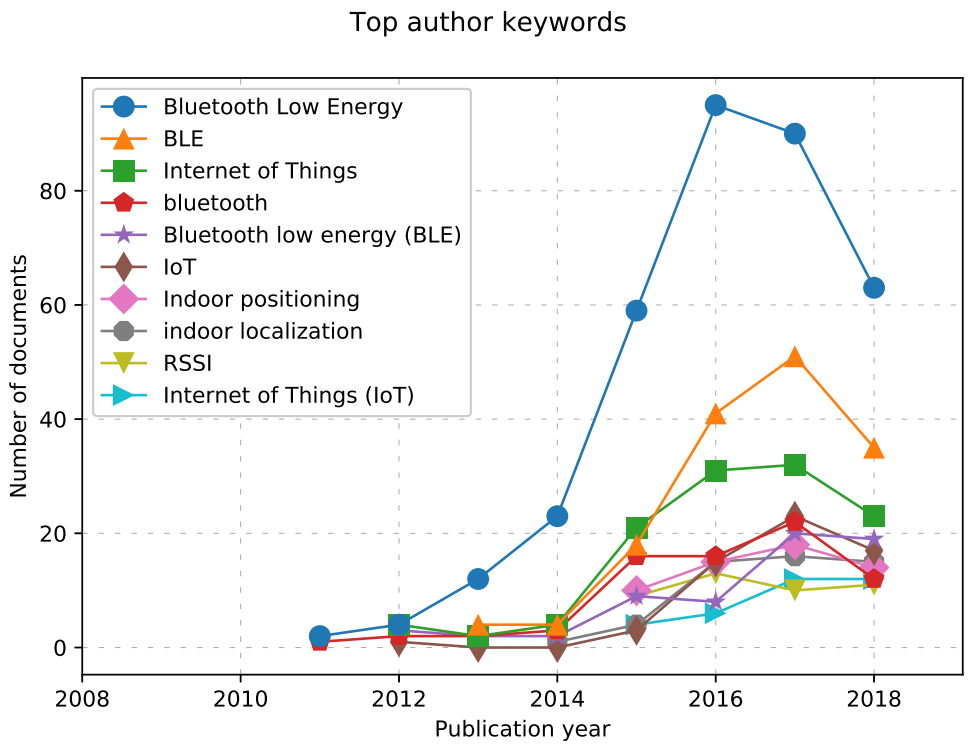
Graph type	Argument	Description
Time line	`-g time_line`	Graphs the number of documents of each topic vs the publication year
Horizontal bars	`-g bar`	Graphs the total number of documents of each topic in horizontal bars
Horizontal bars trends	`-g bar`	Graphs the total number of documents of each topic 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

Below are showed some examples of these graphs types, with the command used.

4.1 Time line graph

Command:

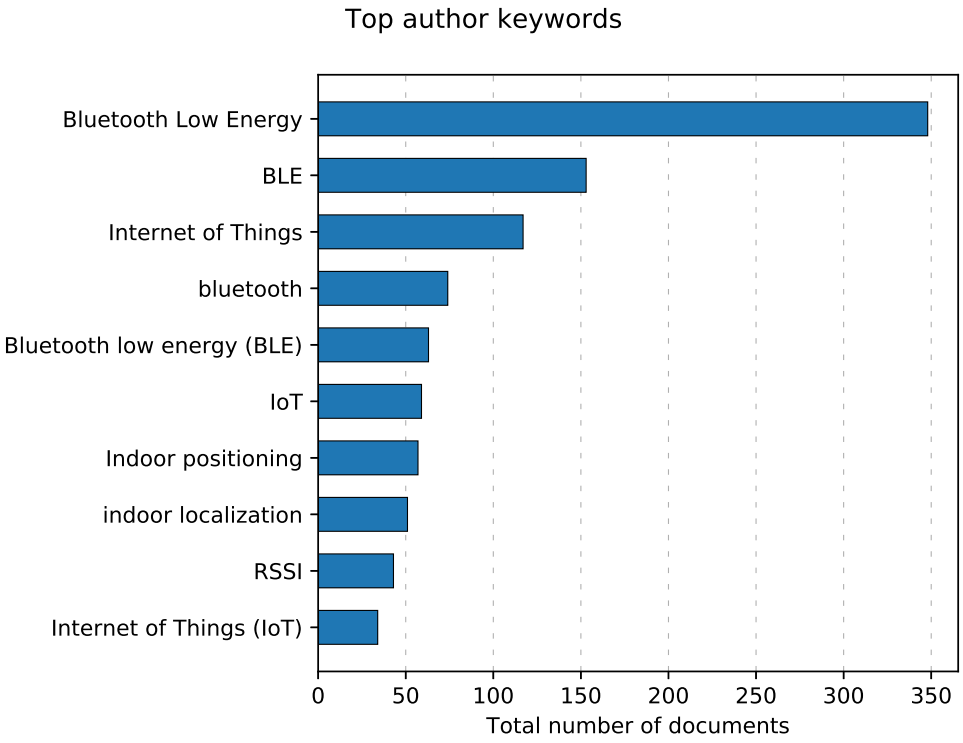
```
python3 scientoPy.py -c authorKeywords --startYear 2008 --endYear 2018 -g time_line
```



4.2 Horizontal bars graph

Command:

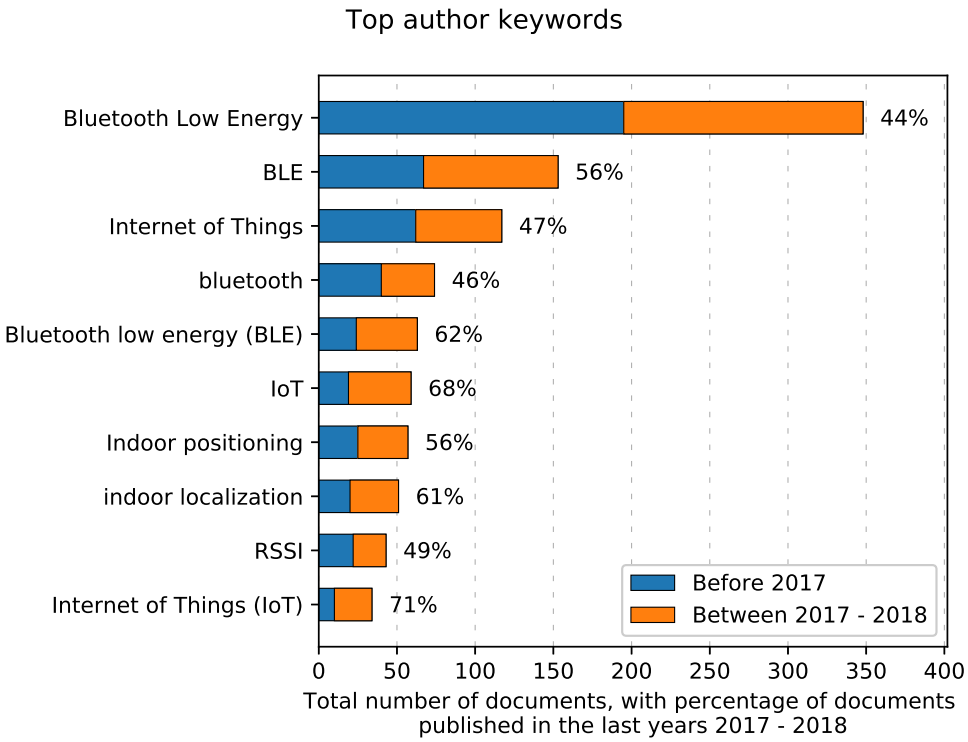
```
python3 scientoPy.py -c authorKeywords --startYear 2008 --endYear 2018 -g bar
```



4.3 Horizontal bars trends

Command:

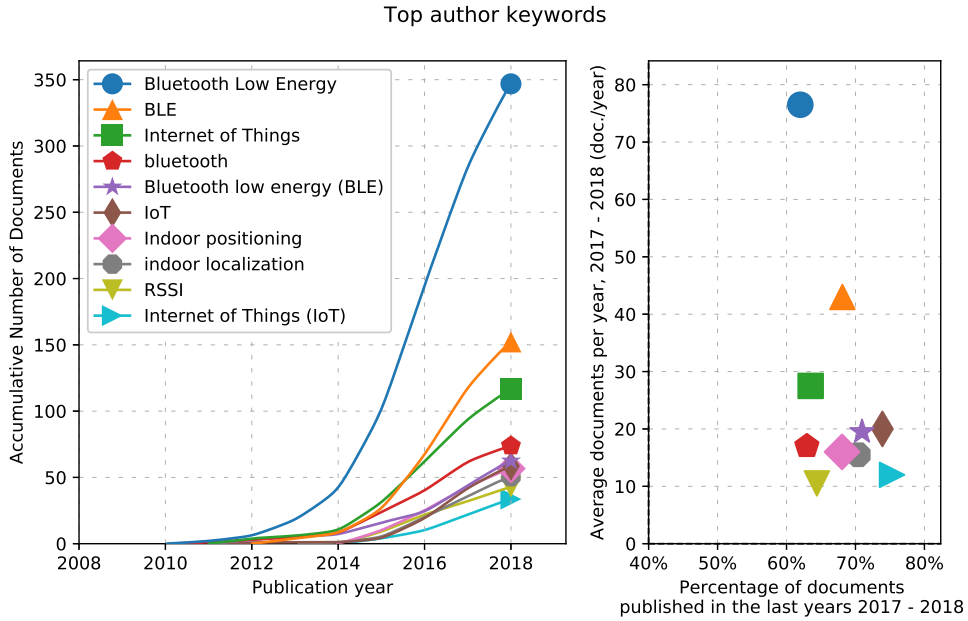
```
python3 scientoPy.py -c authorKeywords --startYear 2008 --endYear 2018 -g bar_trends
```



4.4 Evolution graph

Command:

```
python3 scientoPy.py -c authorKeywords --startYear 2008 --endYear 2018 -g evolution
```



4.5 Word cloud graph

Command:

```
python3 scientoPy.py -c authorKeywords --startYear 2008 --endYear 2018 -l 500 -g word_cloud
```