



By: Juan Pablo Ruiz Rosero jpabloruiz@unicauca.edu.co

## 1. Installation

1. Download and install the last version of Python 2.7 (for example Python 2.7.14) from:  
<https://www.python.org/downloads/>
2. Install the numpy, scipy, and matplotlib libraries for Python with the automatic installation tool **pip**. From Windows, enter in the command line (Windows + R, cmd, and Enter), and run the installation script:

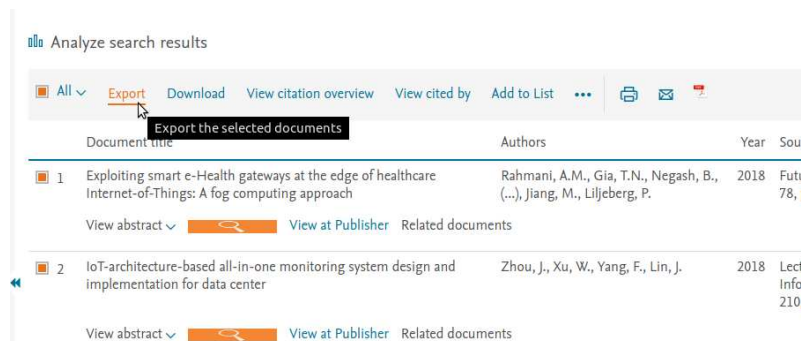
```
python -m pip install --user numpy scipy matplotlib
```

## 2. Download the bibliometric dataset

This section describes how to download the proper dataset from Scopus and WoS. Define a search criteria, it will be used for Scopus and WoS. For this guide 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:

Select your method of export

☐ Mendeley
 ☒ RefWorks
 ☐ RIS Format (EndNote, Reference Manager)
 ☒ CSV (Excel)
 ☐ BibTeX
 ☐ Text (ASCII in HTML)

What information do you want to export?

Customize export

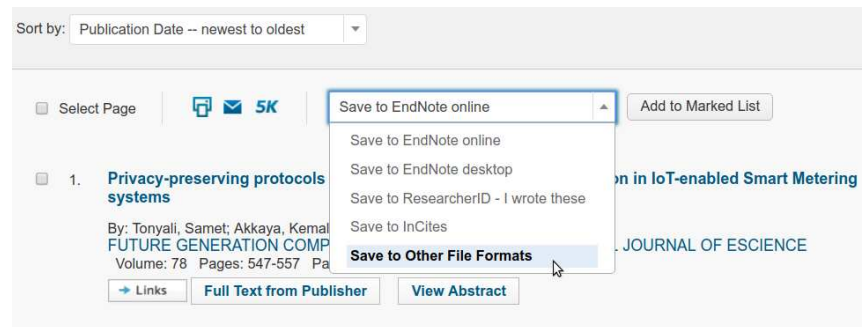
Citation information	Bibliographical information	Abstract and Keywords	Funding Details	Other information
<input checked="" type="checkbox"/> Author(s) <input checked="" type="checkbox"/> Document title <input checked="" type="checkbox"/> Year <input checked="" type="checkbox"/> EID <input checked="" type="checkbox"/> Source title <input checked="" type="checkbox"/> Volume, Issue, Pages <input checked="" type="checkbox"/> Citation count <input checked="" type="checkbox"/> Source and Document Type <input checked="" type="checkbox"/> DOI	<input checked="" type="checkbox"/> Affiliations <input checked="" type="checkbox"/> Serial identifiers (e.g. ISSN) <input checked="" type="checkbox"/> PubMed ID <input checked="" type="checkbox"/> Publisher <input checked="" type="checkbox"/> Editor(s) <input checked="" type="checkbox"/> Language of Original Document <input checked="" type="checkbox"/> Correspondence Address <input checked="" type="checkbox"/> Abbreviated Source Title	<input checked="" type="checkbox"/> Abstract <input checked="" type="checkbox"/> Author Keywords <input checked="" type="checkbox"/> Index Keywords	<input type="checkbox"/> Number <input type="checkbox"/> Acronym <input type="checkbox"/> Sponsor <input type="checkbox"/> Funding text	<input type="checkbox"/> Tradenames and Manufacturers <input type="checkbox"/> Accession numbers and Chemicals <input type="checkbox"/> Conference information <input type="checkbox"/> Include references

Cancel **Export**

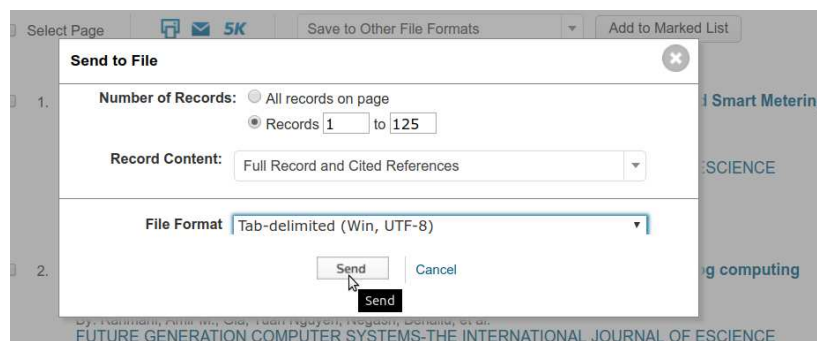
4. Save the file on the folder `/ScientoPy/dataIn`

## 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 Content 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 pre-process and process the bibliometric dataset.

### 3.1. Pre-processing

First we need to pre-process the downloaded dataset. This pre-process joint all the downloaded files from one folder to a single file. Also, this process remove the duplicated files. To pre-process the example dataset (Internet of things papers) run this command inside ScientoPy folder:

```
python preProcess.py dataInIoT
```

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

- **papersPreprocessed.csv:** 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.csv:** this file briefs the pre-process statics results, such as duplicated papers removed, types of documents, and others.

To find more options of the pre-processing script you can run:

```
python preProcess.py -h
```

### 3.2. Extract the top topics

With this script you can extract the top topics of a selected criterion. The ScientoPy script criteria are:

- authors
- source
- subject
- authorKeywords
- indexKeywords
- documentType
- dataBase
- country
- institution

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

```
python scientoPy.py 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, and the h-index associated to each one. Also, this script graphs the evolution of each topic per year, 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:

```
python scientoPy.py -h
```

### 3.3. Analyze pre-defined topics inside a criterion

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

```
python scientoPy.py 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:

```
python scientoPy.py 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 `"` and put the command in one single line.

Also, you can see the results with a parametric graphic (add `--parametric`). This option plot the accumulative documents, average growth rate (AGR), and h-Index of the selected topic, for example:

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

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

```
python 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 (1):

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

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's keywords criterion, you can run the following script:

```
python trendResults.py authorKeywords
```

This script will find the top 200 topics, then it calculates the AGR for the last 3 periods of 2 years. Finally, the 200 top topics are sorted from the highest AGR in the last 2 year period to the lower. The first 10 AGR topics with the corresponding value per period is graphed.

Also, you can find the AGR for selected topics inside a criterion, writing the topic on the argument `-t`, like the following example:

```
python trendResults.py authorKeywords -t "Cloud computing;Security;Big Data"
```

This script have more options like, save the plot on a file, get the list from the lower last period AGR to the higher, or others. For more information you can run:

```
python trendResults.py -h
```

### 3.5. Finding trending documents

The trending documents are the new publications with more citations. To find these publications, this script scale the citations of each publication based on a Year Scale (YS). The YS put more weight to newer publications, by the following Equation (2):

$$YS_i = e^{(i-Y_s)*Y_w} \quad (2)$$

where:

$YS_i$  = Year scale for year  $i$ ;  $i$  = Year;  $Y_s$  = Start year;  $Y_w$  = Year weight, default 1;

The number of citations (citat) of each document is multiplied by the corresponding  $YS_i$ , to get the Scaled citations (S. cita). The final publications list is sorted by this Scaled citations. The following script find the trending documents as described:

```
python topCited.py
```

This script, graphs the all documents citations sum per year, this citations sum scaled (multiplied by the  $YS_i$ ), and the scale per year ( $YS_i$ ). If you want to put more weight to the citation than publication year, you can reduce the Year weight. On the other way, if you want to put more weight to the newer papers you can increase the year weight. The recommended values are between 0 to 5. For example, to get the most cited document of the last years you can run the script with a Year weight of 1.75:

```
python topCited.py --yearWeight 1.75
```

The full document list results is saved in the file [ScientoPy/results/topCitedPapers.csv](#). This script have more options like, save the plot on a file, or others. For more information you can run:

```
python topCited.py -h
```