Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Programa de Ingeniería Electrónica y Telecomunicaciones
ScientoPy v 1.3.3, Installation and User Manual

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

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
   ```

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
   ```

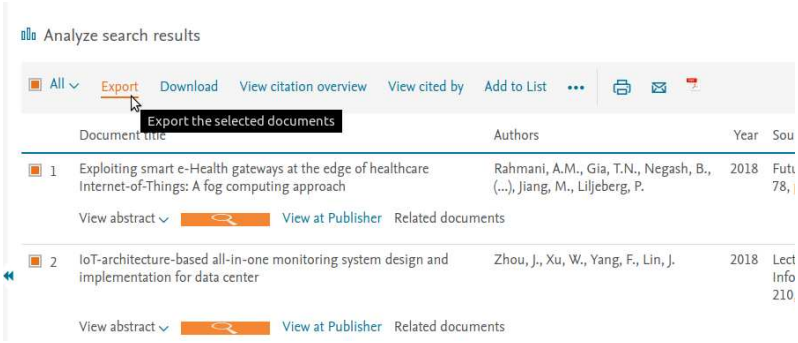
   IMPORTANT NOTE: If you have installed Python 2 and Python 3, run the previous and the following commands with `python3` instead `python`

2 Download the bibliometric dataset

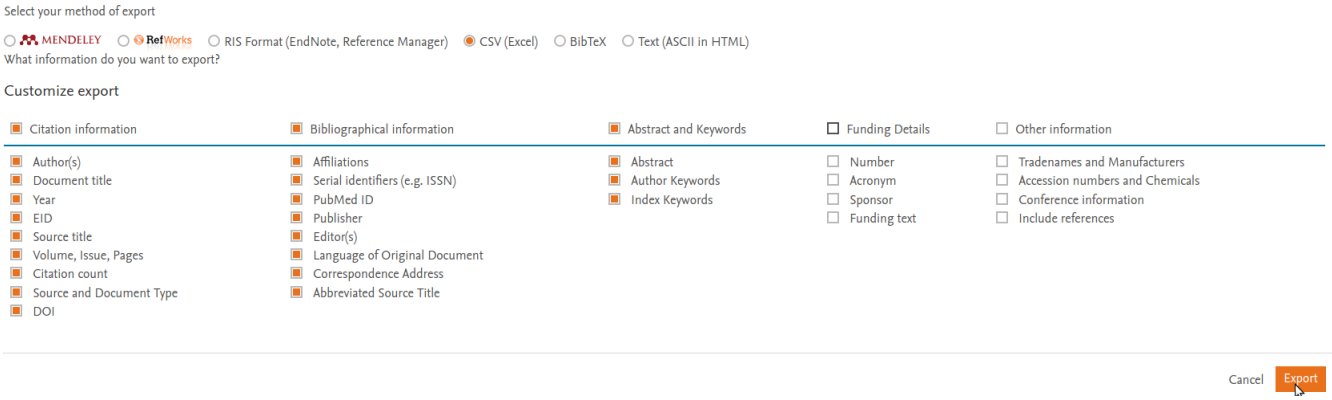
This section describes how to download the proper dataset from Scopus and WoS. Define a search criteria, it will be used for Scopus and WoS. For this guide we are using: "Internet of thing" AND "Gateway"

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:



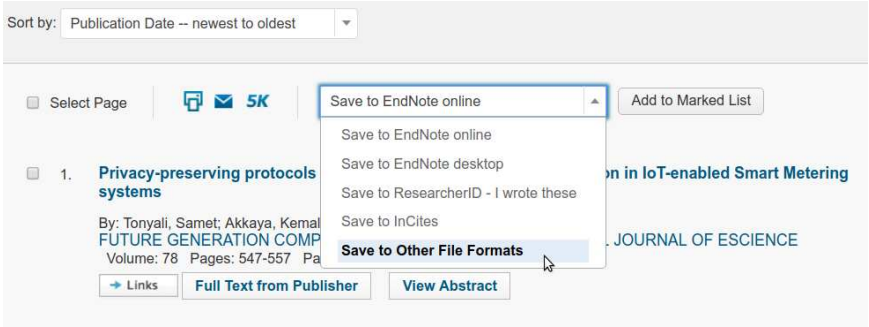
3. Select as method of export **CSV (Excel)**, and select the Customize export **Citation information, Bibliographical information, Abstract and Keywords**, then click on Export:



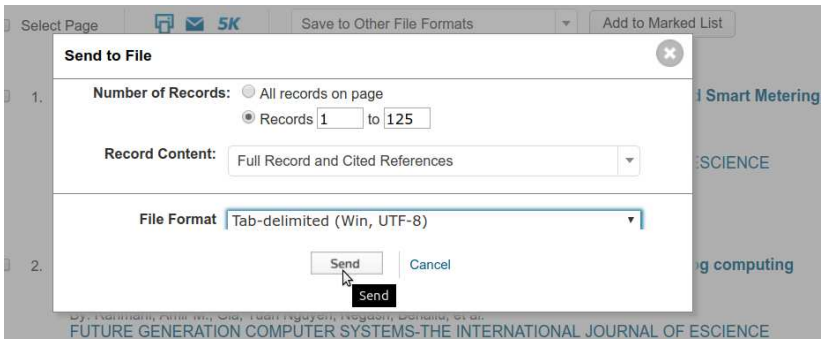
4. Save the file on the folder `/ScientoPy/dataIn`

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 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:

```
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 pre-processing 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 script 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 wildcard
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's keywords you can run this script:

```
python3 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:

```
python3 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:

```
python3 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:

```
python3 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 "\änd put the command in one single line.

3.3.1 Asterisk (*) wildcard

You can use the asterisk wildcar 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 authorKeywords -t "device*"
```

ScientoPy will print the topic found for the previous search using the asterisk wildcard:

```
Topics found for device*:
"devices;device management;Device Interactions;Device objectification;Device;Device integration"
```

You can use this information, to find the statics of each specific topic found, like this:

```
python3 scientoPy.py authorKeywords -t \
"devices;device management;Device Interactions;Device objectification;Device;Device integration"
```

3.3.2 Parametric plot

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

```
python3 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:

```
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 (1):

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

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:

```
python3 scientoPy.py authorKeywords --trend
```

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.

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):

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

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 multiplicated 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:

```
python3 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:

```
python3 topCited.py --yearWeight 1.75
```

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

```
python3 topCited.py -h
```

4 ScientoPy graph types

ScientoPy has 4 different ways to graph the results described on Table 2.

Table 2: ScientoPy output graphs types

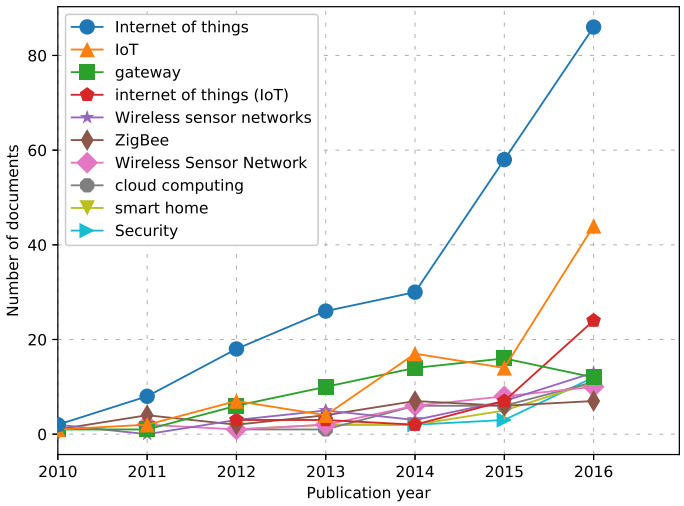
Graph type	Argument	Description
Time line	No argument	Graphs the number of documents of each topic vs the publication year
Horizontal bars	`--bar`	Graphs the total number of documents of each topic in horizontal bars
Word cloud	`--wordCloud`	Generate a word cloud based on the topic total number of publications
Parametric	`--parametric`	Graphs two plots, one with the accumulative number of documents vs the publication year, and other with the AGR vs the h-index

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

4.1 Time line graph

Command:

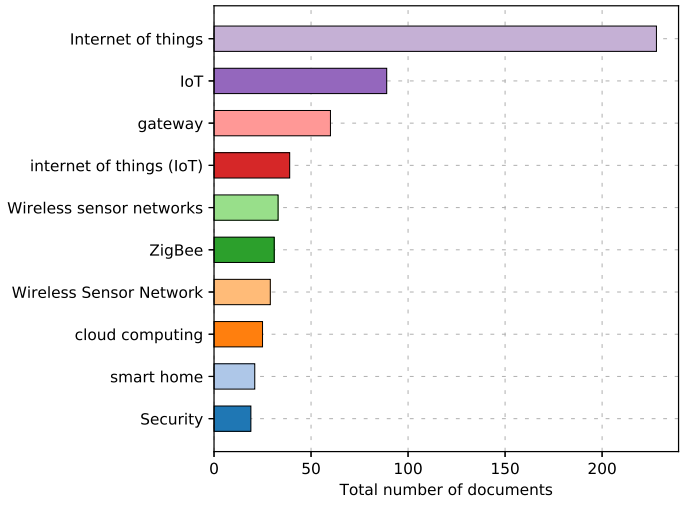
```
python3 scientoPy.py authorKeywords --startYear 2010 --endYear 2016
```



4.2 Horizontal bars graph

Command:

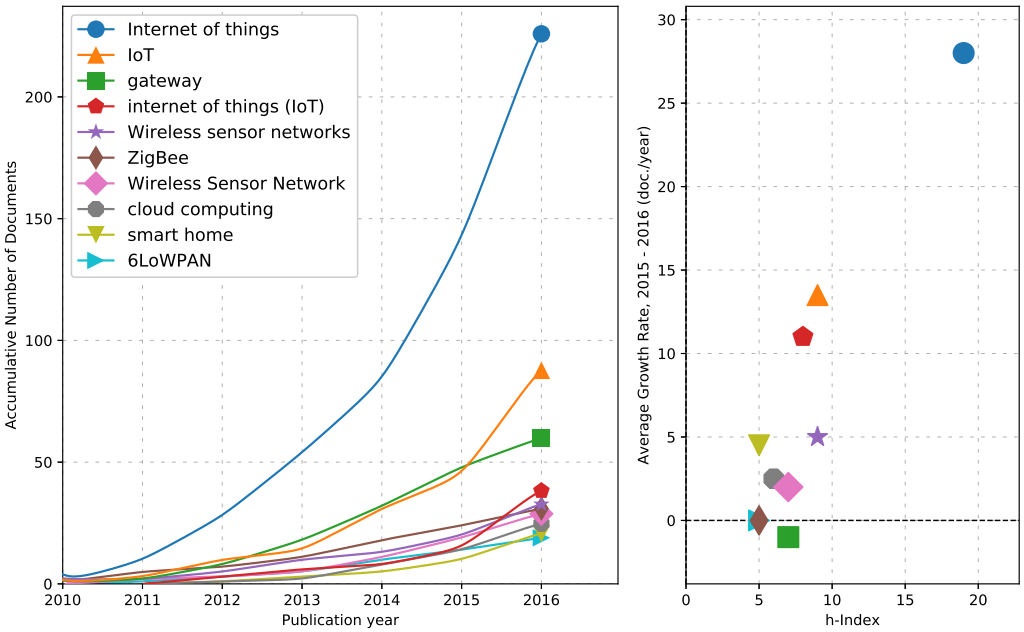
```
python3 scientoPy.py authorKeywords --startYear 2010 --endYear 2016 --bar
```



4.3 Parametric graph

Command:

```
python3 scientoPy.py authorKeywords --startYear 2010 --endYear 2016 --parametric
```



4.4 Word cloud graph

Command:

```
python3 scientoPy.py authorKeywords --startYear 2010 --endYear 2016 -l 500 --wordCloud
```