US Patent Web Crawler Documentation

Introduction

This program can automatically download the information in US Patent Database and save it into .csv file.

Guide to install python environment on Windows machine

Goto Anaconda and download Python 3.7 for Windows, or simply click here.

Open Anaconda Prompt, and you can use the command line. Simple usage in command line::

```
# Go to Disk D:/
$ D:
# Go to folder /sample/
$ cd sample
# Go to parent folder
$ cd ..
```

Installation

Use `requirements.txt` to install all package dependencies.

In command line:

```
# Navigate to the folder of main.py
$ cd [TOP_FOLDER]
# Install all packages
$ pip install -r requirements.txt
```

Single mode: get one patent at a time

Single mode is to download the information of 1 patent. For example, if you want to download the information of this patent. In `input_URL.txt`, give the URL in the first line.

In command line:

```
# Run the python code and specify the mode
$ python main.py --mode single
```

And the result .csv file would be saved in `output/patent_info.csv`. Note that this will overwrite the file if there already exist `patent_info.csv`.

Many mode: get many patents at a time

Let's say you want to download all the patents information in this query page, including the next list so total of 560 patents. In `input_URL.txt`, give the URL in the first line.

In command line:

```
# Run the python code
$ python main.py --mode many
```

And the result .csv file would be saved in `output/patent_info.csv`. Note that this will overwrite the file if there already exist `patent_info.csv`.

Specifying input file or output file path

You can specify output file path and name.

In command line:

```
$ python main.py --mode [MODE] --input [INPUT_FILE_PATH] --output
[OUTPUT_FILE_PATH]
```

For example

```
$ python main.py --mode many --input ./my_URL.txt --output
./output/my_result.csv
```

Checkpoint

At run time, the program would create a checkpoint file in `./output/checkpoint.pkl`. The program would automatically load the checkpoint and continue from the previously disrupted point. If you stop the program manually and want a fresh restart, you have to delete the checkpoint file manually.

When loading from the checkpoint, it is recommend to check the .csv file and ensure that the information of the last patent is correct and complete.

Automatic restart

Since the program would be interrupted and forced to stop by bad internet connection, website not responding, or some random errors, an automatic restart program is used to restart the program from the checkpoint.

Open `restart.py`

```
while True:
    os.system("python main.py") # Modify your command line argument here
    print("Restarting...")
    time.sleep(1) # 1 sec to CTRL+C twice
```

Modify the command line arguments to suit your own need. Then execute by:

```
# python restart.py
```

If there is any interruption that cause the program to terminate, it automatic restarts. Note that even after all the patents are download, the program would not stop. Manually stop the program by quickly pressing CTRL+C twice (or many many times).

Simplify command line options

Command line option can be simplify to one letter.

In command line:

```
$ python main.py --mode many --input ./my_URL.txt --output ./output/my_result
```

can also be used by:

```
$ python main.py -m many -i ./my_URL.txt -o ./output/my_result
```

Output file format

The program will store the output in a .csv and a .json file. The single CSV files (five of them) store the data of all patents, while each JSON file stores a single patent information (including forward citing and backward citing). CSV file is useful to manually examine if there is any error in the crawled data. JSON is for the program to read the saved data later on for index calculation.

Here a list for tag name used in .csv and .json file:

Tag Name	Description	Example
ID	Patent number (9~11 digits)	5479556
title	Patent title	Rotation control apparatus employing a comb filter and phase error detector
date	Patent date (YYYY/M/D)	1995/12/26
inventor_name	Name of inventor	Oh
inventor_city	City of inventor	Seoul
inventor_country	Country inventor	KR
assignee_name	Name of assignee	Goldstar Co., Ltd.
assignee_city	City of assignee	Seoul
assignee_country	Country of assignee	KR

Tag Name	Description	Example
US_class	First 3 digits of US class number	388/805
CPC_class	First 4 digits of CPC class number	H02P
international_class	First 4 digits of international class number	H02P
reference	This is a reference patent	
referenced_by	This is a patent that referenced by	

Duplicate appearance of US class, CPC class, international class would only be shown once in the csv file.

Calculate index

Specify the input foler, output folder, and target years of index calculation.

```
$ python calculate_index.py --input [folder of all input json files] --output [folder of output] --region [LIST of target regions] --year [starting year] [ending_year] --window [List of window lengths] --city_of_collaboration [LIST of cities of collaboration]
```

For example.

```
$ python calculate_index.py -i ./output_Penang/ -o ./index_result/ -r Penang Pen-nang -y 1970 2015 -w 5 3 -cc Penang Pen-nang
```

Note that the names of the cities is case-sensitive (first letter should be capital).

If city_of_collaboration is not set, then all cities are acceptable.

Warnings

- Remember to close output .csv file before running the program, or it can not access the output file.
- Some time some particular patents might not have particular information, the program would give warning messages and the program would continue. if you want to suppress all warning messages, you can use:

  ```
  $ python main.py -m many --warnings False
  ```

Debug

Use `--debug True` to turn on debug message.

```
$ python main.py -m many --debug True
```

Contact

Please send to tommyrpg1010@gmail.com if you have any question or need any modification of the functions.