EDGAR CLEANER

Python project for cleaning EDGAR data

Version 1.0

JUNE 5, 2019

TEXAS TECH

Table of Contents

Description

The processing is by years. The original edgar files are used, keeping the necessary columns. There might be some rows that are robots, these rows are deleted. In addition, it is necessary to match the information provided in edgar files with the information from the master files, so a join is performed to add these master's files information.

Python files:
- clean_edgar.py: Main file.
- config.py: Manage the configuration.
- import_masters.py: It contains the code to import master's files.
- processor.py: It contains the code that handles the processing of the files.
- transfer.py: It handles the upload to Dropbox.

Clone repository

Clone the repository in your home directory.

```
git clone https://github.com/mikaelapisani/edgar-cleaner.git
```

Configure environment

Docker

Build image:

```
docker build -t edgar_cleaner .
```

It is necessary to specify 3 volumes, which are the directories for the data folders to process,
the masters files and the temporary results dir. (In case that the upload to dropbox fails, the results files will be at that directory).

```
docker run -ti -v <data_dir>:/root/data -v <masters_dir>:/root/masters -v <results_dir>:/root/results edgar_cleaner bash
```

Once inside the container follow the execution manual to execute the program.
The configuration root would be at '/root/edgar-cleaner/config.properties'

Virtualenv

```
python3 -m venv edger-cleaner
source edgar-cleaner/bin/activate
cd edger-cleaner
pip install --upgrade pip
pip install -r requirements.txt
```

TTU cluster

1. It is necessary to have installed conda. Follow these guide if it is not installed:

   http://www.depts.ttu.edu/hpcc/userguides/application_guides/python.local_installation.php

2. Create virtualenv:
   ```
   bash . $HOME/conda/etc/profile.d/conda.sh conda activate cd $HOME/edgar-cleaner pip install --upgrade pip pip install -r requirements.txt conda deactivate
   ```

3. Download masters files executing the following command
   ```
   bash mkdir $HOME/masters qsub $HOME/edgar-cleaner/mpi/mpi_masters.sh
   ```

4. Check that the masters files are downloaded
   ```
   bash ls $HOME/edgar-cleaner/masters
   ```

5. Remove logs
   ```
   bash sh $HOME/edgar-cleaner/remove_logs.sh
   ```

6. Download the repository at your home directory.

7. Edit config.properties file with the corresponding paths.

8. Edit mpi.sh with the correspondent year, for example:
   ```
   bash export YEAR=2015 sed "s/YEAR/$YEAR/g" $HOME/edgar-cleaner/mpi/mpi.sh > $HOME/edgar-cleaner/mpi/mpi_$YEAR.sh
   ```

9. Execute the following command:
   ```
   bash qsub $HOME/edgar-cleaner/mpi/mpi_$YEAR.sh
   ```

MPI

In order to run the jobs in the cluster it is used MPI.
It can be found more information about how to run jobs in the cluster in the following link:
http://www.depts.ttu.edu/hpcc/userguides/general_guides/job_submission.php
In this case it is configured only 1 process as it has to be serial.
In the future, the code can be modified in order to process in parallel the information and clean the data faster.

Commands

- list jobs: qstat
- kill job: qdel
- see output: tail -f .*
- see information about a failed job: qacct -j

The project name is in the file mpi/mpi.sh in the -N parameter: "MPI_EDGAR_YEAR"

Execution manual

For all years: python clean_edgar.py -c
For specific year: python clean_edgar.py -c -y

Configuration

Configure the file `config.properties` as follows:

```
log_level=INFO
since_year=1993
threshold=50
error_code_limit=300
output_size_mb=5000
access_token=
dropbox_folder=/EDGAR data 14-17/
dropbox_timeout=2000
dropbox_chunck=4194304
data_path=/home/mikleal
master_path=/home/mikleal/masters
results_path=/home/mikleal/results
```

- log_level: Log Level (INFO/DEBUG)
- since_year: Year since master files are downloaded
- threshold: Limit of firms for a given IP in a day, if this threshold is reached it is considered as robot
- error_code_limit: HTTP code from which rows are going to be removed
- output_size_mb: limit size in MG for output files
- access_token: Dropbox access token

- dropbox_folder: Dropbox folder name
- dropbox_timeout: Timeout for Dropbox connection when uploading files
- dropbox_chunck: Chunk size to read files to upload to Dropbox
- data_path: Path for data, inside this folder there should be one folder per year
- master_path: Path for masters' files
- results_path: Path for temporary result files.

Data Cleaning

The process of cleaning the data consists in the following steps:

1. Remove crawerls
2. Remove index
3. Remove codes >=error_code_limit
4. Remove robots: based on the number of unique firms that a given IP address downloads on a given day
   If it is more than the threshold is a robot
5. Merge data with masters' data by cik and accession number

Columns from result

- ip: with ###.###.###.xxx – first three octets of the IP address with the fourth octet obfuscated with a 3 character string that preserves the uniqueness of the last octet without revealing the full identity of the IP
- date: apache log file date
- time: apache log file time
- cik: SEC central index key associated with the document requested
- accession: SEC document accession number associated with the document requested
- extention: document file type (e.g., html, txt, etc.)
- Form Type: submission type

- Date Filed: date when the form was filed

Save the data

The result data is saved in Dropbox.
As the files are going to be analyzed with SAS it is necessary that the size would be less than 5GB.
In order to achieve these, for each day, it is checked whether or not when appending the result for the day the size would be greater than the threshold.
In that case, the accumulated result is uploaded to Dropbox and continue with the processing until there are no more files to process.

It is necessary to create in Dropbox an app to save the files into a folder. Generate a token for that app. This token is the one that should be in the property access_token in the configuration.

Handling errors

- In case the file could not be uploaded, it would be saved in the results folder and continue processing the rest of the files.

- In case that other errors occur, the process would show a message in the console with a description of the error.