Graphfetcher user guide

Hitoshi Yamauchi

2013-1-26(Sat)

Abstract This is a user guide of graphfetcher tools. graphfetcher tools are for analyzing a Wiki's graph structure and computing the rank of the web pages by eigenanalysis (also known as Google PageRank). I use this tool for my computational literature research project. Though, it can be used in any area that is written in Wikipedia, e.g., ranking of the musician, ranking of politician, and so on.

1 Introduction

One of my friends who studies literature asked me how I would analyze the relationships between authors. For instance, "How can we measure the influence of Shakespeare in English literature compared to other writers?"

This question inspired me to develop these tools. The detail of the story was published as an article "Authors in a Markov matrix: Which author do people find most inspiring?" [**?**].

You can find the following tools here.

- `Link_Vector_Extractor`: Generate the author vector from the data.
- `Graph_Extractor`: Generate the adjacency matrix from: the data and the author vector.
- `Page_Rank`: Compute PageRank.
- `Remapper`: Re-map the author vector according to the PageRank result.

These tools are developed on a notebook computer, CPU: Intel(R) Core(TM) DUO CPU P8400, 2 Cores, OS: 64bit Linux 3.2.0.32, Kubuntu 12.04. The programming environment is: Python 2.7.3, Beautiful Soup 4.0.2, matlab R2006a, octave 3.2.4.

2 A step by step example

Here I will show you an example using the list of Italian writers.

2.1 Download the writer's pages

I first need download the writer's Wiki pages. I found a list of Italian writer Wekipage, `http://en.wikipedia.org/wiki/List_of_Italian_writers`. I use this as the root page for the download.

```
% cd graphfetcher
% mkdir -p data/italian_writer
% cd data/italian_writer
% cp ../english_writer/ wget_download_en_wiki_en_writer.sh \
  wget_download_en_wiki_de_writer.sh
```

I edit the `wget_download_en_wiki_de_writer.sh` to change the list and run the program. After download the pages, please create an output directory, `wiki_out` at the downloaded directory. For instance, if you are at `graphfetcher/data/italian_writer`, and English wiki pages are downloaded, you have `en.wikipedia.org` directory,

```
% mkdir -p en.wikipedia.org/wiki_out
  # path: graphfetcher/data/italian_writer/en.wikipedia.org/wiki_out
```

2.2 Getting the author vector

First we need an author vector. To extract this vector, we use the `vectorextractor` tool. One easy way is reuse one of the `test_linkvectorextractor_*.py` file.

```
% cd ../vectorextractor
% cp test_linkvectorextractor_en_en_0.py \
  test_linkvectorextractor_italian_en_0.py
```

These programs are also unit tests. There are baseline file comparison test in the test methods: `test_linkvectorextractor_ascii`, `test_linkvectorextractor_utf8`. For the first time run, you should disable these comparison test, otherwise you will get test failed errors.

2.3 Getting the adjacency matrix

From the author vector and all the wiki pages, we can analyze the link structure for each authors. We generate an adjacency matrix as a result. We use `graphextractor` to get the matrix. Same as the `vectorextractor`, it is easy to copy the one of `graphextractor` test and modify it.

```
% cd ../graphextractor
% cp test_graphextractor_en_en_0.py \
  test_graphextractor_italy_en_0.py
```

The default options are silent. You can find the option settings in the `test_graphextractor_italy_en_0.py` file, search `opt_dict` definition. If you want to know what is processed, change the following parameters:

- `'log_level': 3`
- `'is_print_connectivity': True`

This settings reports the analysis result of each page's link structure. This program is also a unit test program. Therefore, the result comparison is performed. For the first time run, there is no baseline, so you will see the error in that case.

This tool can also output a dot graph file. But if the vector size is more than 100, it may not help to see the structure of the graph.

There are two kinds of graph output in the current code. matlab's sparse matrix format and my own format. If you need another format to support, you should develop your exporting code.

2.4 Compute the Eigenvector

Note: This needs matlab software. This is mathematical analysis part. I use matlab. Please see the m-file `test_pagerank_italian_en_0.m`. You should update the author vector file name and adjacent matrix function in the following `madj` line in the m-file.

- `madj = italian_en_writer_adj_mat();`

There is also a tool to visualize the sparse matrix. See `test_show_adjacency_mat_0.m`.

2.5 Remap the result

My program use the index of the author vector as identifier. Because the PageRank algorithm has the sink link removal. The vector size may change during the processing. I used the author name as the identifier in my first implementation, but, it turned out the matlab's utf-8 support was not sufficient for me. A utf-8 string becomes a number array, so you can not see the character. Therefore, I switched the identifier to the index of the vector. After PageRank is computed, the PageRank values are associated with the indices. The remapper re-maps the indices to the author names.

Please see `test_remapper_0.py`.

3 Tips

- If you see UnicodeEncodeError (e.g. UnicodeEncodeError: 'ascii' codec can't encode character u'\xf2' in position *: ordinal not in range(*)), set utf-8 environment. For instance, environment variable, `export LC_ALL=en_US.utf-8` in bash. You usually should be able to read the utf-8 file name in your terminal.

4 Conclusion

I tested in this document with Italian writers on English wikipedia.