Pagelyzer

Installation and Configuration Manual

Andrés Sanoja, LIP6 / Alexis Lechervy
Université Pierre et Marie Curie

Responsibles WP :
Matthieu CORD/UPMC
Stéphane GANÇARSKI/UPMC

This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

Enviroment Verification and Configuration

The tools *pagelyser_analyzer*, *pagelyzer_changedetection*, *pagelyzer_capture* and *pagelyzer_trin* are written in Ruby 1.9.1. In the other hand for the change detection a dependency to MarcAlizer tool written in Java. Therefore, this should be taken into account in the enviroment verification process. The development enviroment was Linux Ubuntu 11.4, the package description is done following its repositories, but in theory should be compatible with Debian repos.

Ruby Installation

We need to be carefull with this step because the software won't work on the 1.8.x versions of Ruby.

```
sudo apt-get install ruby1.9.1-full
```

After that we should check that both, ruby and *rubygems*, are been properly installed.

```
$ ruby -v
1.9.2p290 (2011-07-09 revision 32533) [i686-linux]
```

It is enough to match the version number. Any doubts there are several tutorials to do this [1]. Now we check the *rubygems* package manager:

```
$ gem -v
1.3.7
```

Instalation of Pagelyzer 0.9.1

Pagelyzer is a set of components that can be used (most of them) independently, but in the case of change detection they are all used as a chain for simplicity of integration based on data streams.

The software can be downloaded from:

```
http://www-poleia.lip6.fr/~sanojaa/pagelyzer-ruby-0.9.1-standalone.zip
```

It is enough to unzip the compressed file into the desired destination and after all dependencies are met we are ready to go.

Installing Dependencies

After the language and the package manager are properly configured and installed, we may proceed to install the dependencies:

```
$ sudo apt-get install libxslt-dev libxml2-dev
$ sudo apt-get install openjdk-7-jdk
$ sudo apt-get install imagemagick
$ sudo apt-get install xvfb
```

Note1: Installing the selenium-webdriver may cause some warnings in text encoding that should be fine, in almost all the cases.

Note 2: The java installation is a reference to remember that it should be present.

Note 3: ImageMagick 6 is optional, only needed for thumb-nailing and cropping web page viewport. This thumbs area useful for integrating with other tools and for future optimization of change detection process . If you don't want to use it ignoring parameter “--thumb” should do the trick

We need to install also some ruby libraries needed by the software. This step can be done simple using Bundler gem. To install it:

```
$ sudo gem install bundler
```

Get into the project folder and type:

```
$ bundle
```

When finished we will have all dependencies installed.

The folder structure is the following:

```
• bin
    ○ pagelyzer_capture
    ○ pagelyzer_changedetection
    ○ pagelyzer_analyzer
    ○ pagelyzer_train
• data
    ○ js
        ▪ decorated_mini.js
        ▪ decorated.js
        ▪ compress_js.rb
• doc
• ext
    ○ ex1
        ▪ bin
        ▪ dico
        ▪ images
    ○ ex_hibride.xml
    ○ ex_images.xml
    ○ ex_structure.xml
    ○ others examples files
• Gemfile
• lib
```

```
  ○ DIFF.jar
  ○ MarcAlizer.jar
  ○ MarcAlizer_lib
  ○ pagelyzer_block.rb
  ○ pagelyzer_capture.rb
  ○ pagelyzer_convex_hull.rb
  ○ pagelyzer_dimension.rb
  ○ pagelyzer_heuristic.rb
  ○ pagelyzer_point.rb
  ○ pagelyzer_item.rb
  ○ pagelyzer_separator.rb
  ○ pagelyzer_url_utils.rb
  ○ pagelyzer_util.rb
  ○ pagelyzerjavalib.jar
```

```
• out
• pagelyzer
• README.md
```

Note: *out* folder is intended to be an output folder, but it is optional. Can be overridden with parameters.

Command-line Parameters

Pagelyzer:

```
USAGE: pagelyzer [--help|--version] [<command> <command_options>]
```

Capture:

```
USAGE: pagelyzer capture --url=URL [--output-folder=FOLDER] [--
browser=BROWSER_CODE] [--no-screenshot] [--thumbnail] [--help]
[--headless]
```

This tool aims to produce an HTML document with the visual cues integrated, called Decorated HTML. This allows to save the state of a browser at the moment of capture

Browsers code are the same as defined in selenium. For instance:

- firefox (default)
- chrome
- iexploreproxy
- safariproxy
- opera

Analyzer:

```
USAGE: pagelyzer analyzer --decorated-file=FILE [--output-
file=FILE] [--pdoc=(0..10)] [--version] [--help]
```

Change Detection:

```
USAGE: pagelyzer_changedetection [--url1=URL --url2=URL |
urls=FILE] [conf=CONF_FILE] [--doc=(1..10)] [--output-
folder=FOLDER] [--browser=BROWSER_CODE | --browser1=BROWSER_CODE
--browser2=BROWSER_CODE] [--verbose] --type=[images|structure|
hybrid] [--url-archive]
```

The browser code have the same restriction as the previous tools, that is normal because the change detection tool uses them.

If no Degree of Coherence is given, a default of doc=6 will be chosen.

Train

```
USAGE: pagelyzer train --urls=FILE_TEXT --param_init=FILE_XML
--url_archive [--doc=(1..10)] [--output-folder=FOLDER] [--
browser=BROWSER_CODE | --browser1=BROWSER_CODE
--browser2=BROWSER_CODE] [--verbose] --type=[images|structure|
hybrid]
```

Examples

Capture a web page with default parameters:

```
$ ./pagelyzer capture --url=http://www.google.fr
```

It will copy to $HOME_FOLDER/pagelyzer the outcome. If the folder does not exist it will be created. It will create three files:

- firefox_www_google_fr.html (rendered version of the web page)
- firefox_www_google_fr.dhtml (rendered version with visual cues included for segmentation algorithm)
- firefox_www_google_fr.png (webshot of the page)

Change detection on two pages with default parameters

```
$ ./pagelyzer changedetection
--url1=http://www.host.com/page1.html
--url2=http://www.host.com/page2.html
```

Change detection on two pages with hybrid method and headless mode

```
$ ./pagelyzer changedetection
--url1=http://www.host.com/page1.html
--url2=http://www.host.com/page2.html –type=hybrid --headless
```

Change detection with different browsers

```
$ ./pagelyzer changedetection
--url1=http://www.host.com/page1.html
```

```
--url2=http://www.host.com/page2.html --browser1=firefox
--browser2=chrome
```

url1 will be evaluated with *browser1* and *url2* with *browser2*

Change detection with same browser (the most common case)

```
$ ./pagelyzer changedetection
--url1=http://www.host.com/page1.html
--url2=http://www.host.com/page2.html --browser=firefox
```

same browser for both urls

Remarks:
Firefox driver is the default to selenium. For installing other browsers can reference to [2], e.g. to run pagelyzer on your chrome/chromium instance, you should install the ChromeDriver before: Download the appropriate version from http://code.google.com/p/chromedriver/downloads/list
Unzip it and copy it to a visible folder, e.g: $ sudo cp chromedriver /usr/bin/

External References:

[1] http://answers.oreilly.com/topic/2845-installing-ruby-1-9-on-a-debian-or-ubuntu-system/