Chinese Character Component (部首) Scraper

S. Matthew English

February 2015

1 Introduction

This document details the process of creating a web crawler which aggregates the component radicals (部首) of Chinese characters. This will be useful for such endeavours as POS tagging, amongst others. The working prototype of the system can be found on my GitHub page, the final output of this software is a table of Chinese characters decomposed into their component radicals which can be found in the following file, or through this link.

`character_component_scraper\character_component_document.txt`.

2 Scraper Construction

The basis for this web crawler is Scrapy, an open source framework for extracting data from websites.

The information that comprises the data was taken from `tool.httpcn.com`. The selection of this particular website from the myriad options, (`zdic.net`, `thn21.com`, etc.), was made on the basis of the logical structure under which the characters there are organized, since this arrangement significantly facilitates the link extraction process employed by the web scraper to crawl through the site.

The data is at first extracted using XPath syntax, this functionality can be observed in the file `spiders\component_character_scraper.py`. Owing to the structure of the HTML tags of `tool.httpcn.com` it was not practical to extract elements in a "clean" manner, which in turn necessitated a post-processing step for refining the data and reproducing it in a comprehensible form, this was done through the file `character_component_scraper\text_processing.java`.

3 Usage

The web crawler can be run from within the `character_component_scraper\` directory with the command:

```
scrapy crawl character_component_scraper -o items.csv -t csv
```

This will result in the creation of a file called `character_component_scraper_items.csv`. This is the raw data upon which the post-processing must be applied. That can be executed with the command:

```
java text_processing
```

One important caveat of the post-processing step is that the variable `input_location` of `text_processing.java` must be set to the location of `character_component_scraper_items.csv`.
To create the output file one can pipe the data to a `.txt` file, i.e.

```
java text_processing >> output.txt
```

4 Future Work

After having completed the post processing functionality in Java it came to my attention that the use of regular expressions for parsing HTML tags is ill-advised, and generally speaking a dedicated parsing tool such as jsoup is accepted to be the best-practice solution to this problem, integrating this library is an important next step in the amelioration of our system.

Additionally one further area of potential improvement is the recursive application of radical decomposition (分解查字). The data on `tool.httpcn.com` provided two layers of component analysis, but in certain instances this does not represent the "root node" of the character, which is to say it they can be further decomposed. This would involve querying our table for this compound character and further segmenting it's contingent pieces until we've reached a terminal radical.