# Report(part)

## 1.Data collection and clean

First, we begin to collect data, we write a spider “data\_collect.py” to scrip data from <http://www.gutenberg.org/ebooks/search/?sort_order=release_date> and clean data.

The spider has five functions download,update,merge,clean and toxml. Download function will download e-books by size to “downPATH” folder, and write the largest and smallest id to "conf” file. Update function will get the latest e-book in local host by reading “conf” file and get the latest e-book in website, if there are different, it will download all new books. Merge function will merge all text file in a folder to one text file. Clean function will extract all English words in a file and add a id at the beginning. Toxml function will add a <text> tag to plain text file to convert it to a xml file. For details of it, please cheek the help by commend “python data\_collect.py -h” and the part ”1.data\_collect.py” of “README.md” in my github repository <https://github.com/walkureHHH/juner_1_dp2_porject1> .

## Inverted index by python

Than we will do inverted index by python, we should use a cleaned and merged text file whose split character between id and content is TAB. To do inverted index with python, I write two python file “mapper.py” and “reducer.py”. The usage and example please cheek the part “2.mapreduce\_python” of “README.md” in my github repository <https://github.com/walkureHHH/juner_1_dp2_porject1> .

## Ranking by TFIDF

Than we will do search and ranking, I write the “search.py” to do this job. The file be searched should be cleaned and merged whose split character between id and content is SPACE. The usage and example please cheek the part “3.tfidf” of “README.md” in my github repository <https://github.com/walkureHHH/juner_1_dp2_porject1> .