**Cover page**

Student Name: Anni Li

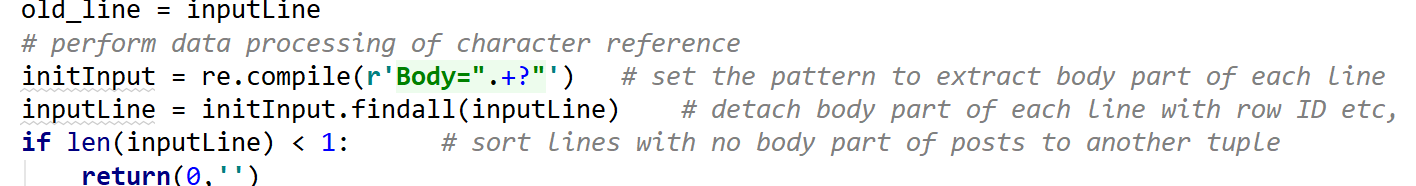
Student ID: 30269776

**User Documentation of HardwareRecs Analysis**

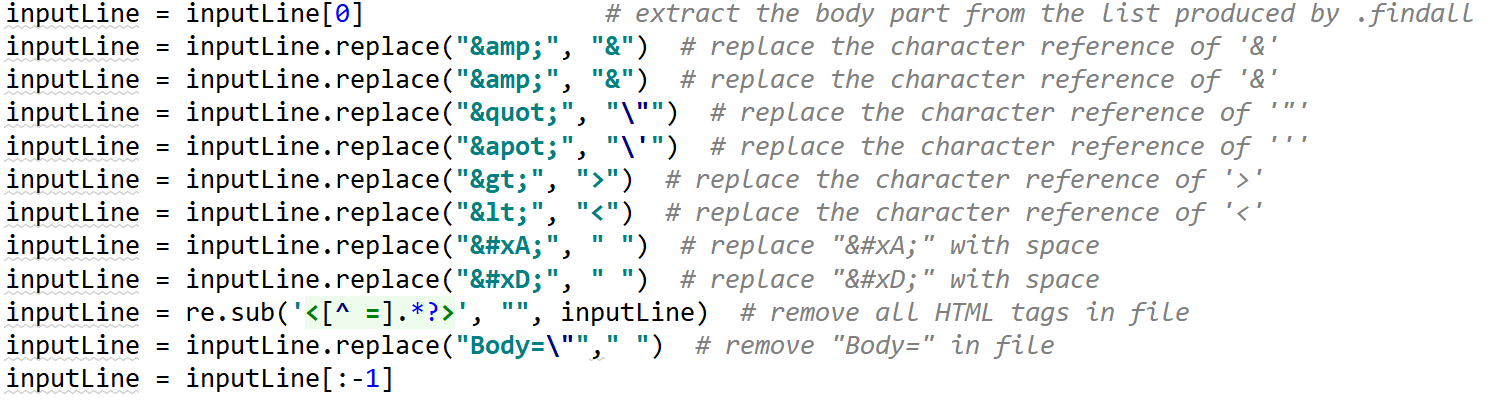
1. **Handling with File Contents and Preprocessing**

Preprocessing program read an input file by lines. The main body of data is extracted. For the purpose of data filtering, the body part needs to be cleaned first. Regular expression module is imported to help refine data.

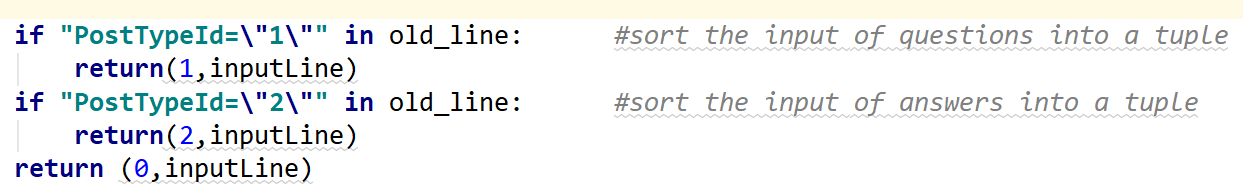
For the part of file which does not include any body part, the line is filtered by the program. Figure 1.1 below show this mechanism.

Figure 1.1 Exclude content doe not include body part

The body part filtered with special reference characters removed and replaced with space, including “&amp”, “&quot;”, “&apos;”, “&gt”, “&lt”, “&#xA;”, “&#xD;”, all HTML tags and containing inside HTML tags. The coding of cleaning body part is shown below in Figure 1.2.

 Figure 1.2 Removal of special reference characters and HTML elements

The clean body is then sorted by “PostTypeId”, where when PostTypeId is 1, the line of body is categorized to question, whereas PostTypeId is 2, the line of body is categorized to answer. The other type of body part is recognized as other types, like shown in Figure 1.3.

Figure 1.3 Find body part by type of post

After the clean body is recognized into questions and answers, the lines are written into two output files and exported in txt format. The input file and exported files are closed after all processes are completed. which is shown in Figure 1.4 below.

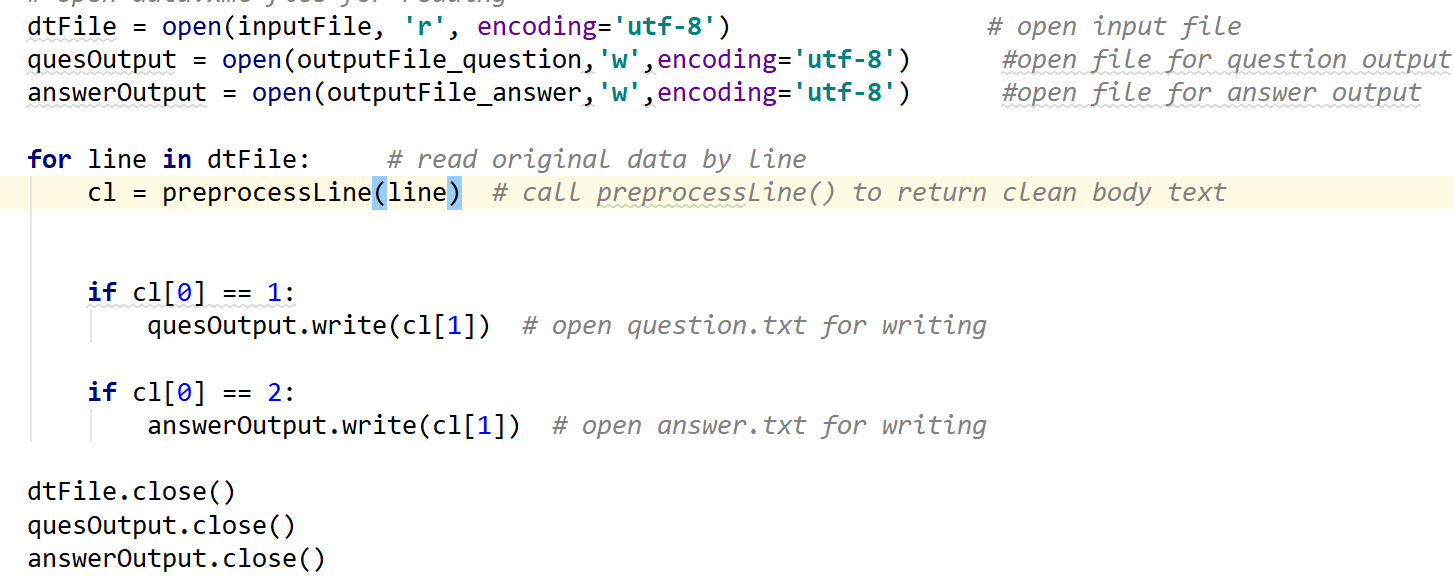


Figure 1.4 Write and export files in txt form

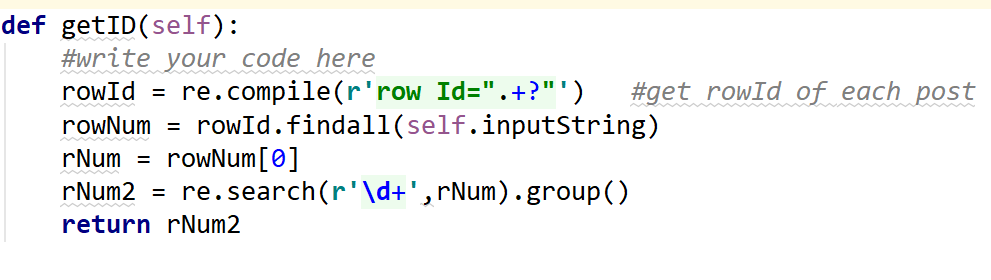
If the program works properly, the process should finish with exit code 0.

1. **Data Parser**

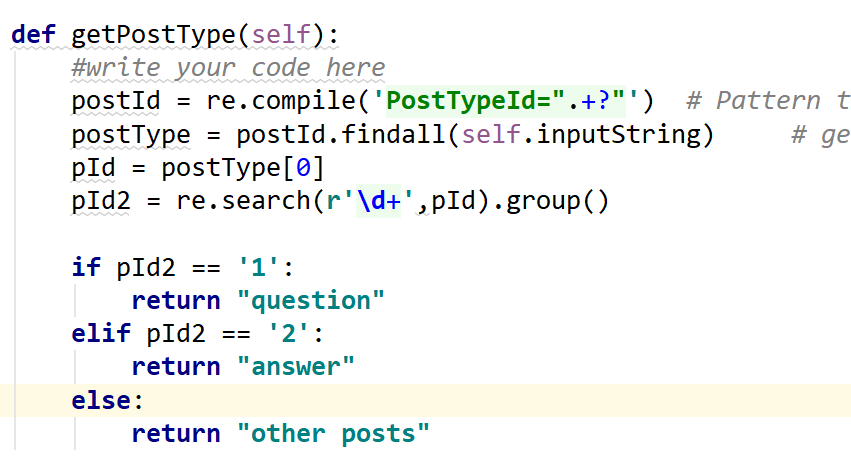
The program imports preprocess data part and re module to help modify input data.

A string including post ID, type of post, post creation date, and the clean body of posts is returned after extraction, with unwanted signs or characters removed.

The post ID is searched out in the original input file with re.compile(), re.findall() and re.search(), and put into a group. The result is returned to output.

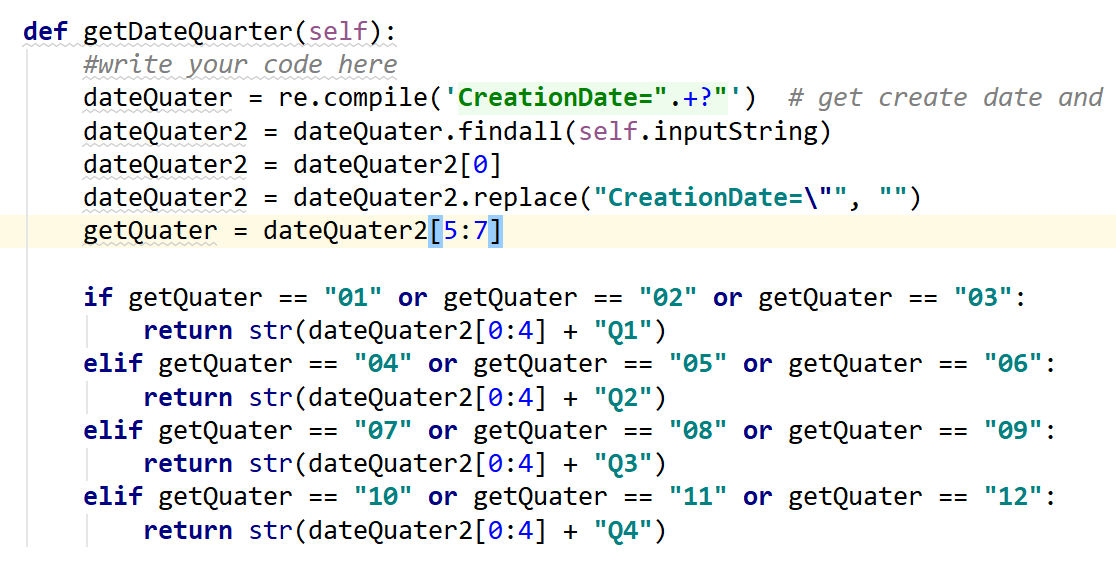
Figure 2.1 Mechanism of get post Id

Similar mechanism is used for getting post type of input file. However, in this part, the posts are recognized into different groups by post type ID as shown in Figure 2.2. The recognition of each line is returned by the function as strings.

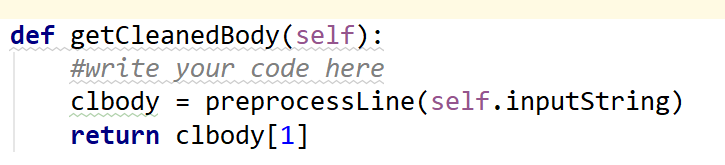
 Figure 2.2 Mechanism of sorting post type

The mechanism of extracting post creation dates is shown in Figure 2.3. In getDateQuarter function, the “Creation Date” part of original file is extracted with double quotes removed.

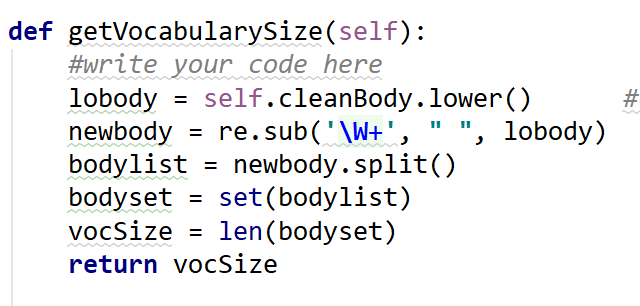
The size of each line of clean body is given by getDateQuarter, where punctuation and replicated items are removed. The dates of posts are grouped in quarter intervals and append to corresponding year in time sequence. For example, a post created on “2015-09-09” will be allocated to the group of “2015Q3”. Codes for getDateQuarter is shown in Figure2.4.

Figure2.4 Mechanism of get date and quarters

To get clean body of input file, the preprocess data function imported. Since the result is returned in a list form with some remaining punctuations, the second item of the list will be the actual clean body in the content returned by the function (Figure 2.5).

 Figure 2.5 Launch preprocess line function

After getting the clean body part, the content is taken to getVocabularySize. This function firstly converts all contents to lower case. Then the punctuations in clean body content are removed with re.sub() function. To remove the replicated words in context, the content is converted to a set. The count of set length gives the vocabulary size of clean body contents (Figure 2.6).

 Figure 2.6 Mechanism of getVocabularySize

1. **Data Visualisation**

This program imports function of parser and matplotlib module to sort the data in input file. The visualizeWordDistribution function sorts given data by the vocabulary size and count the total number of each group. The group is divided by interval of 10 on a 100 basis (Figure 3.1).

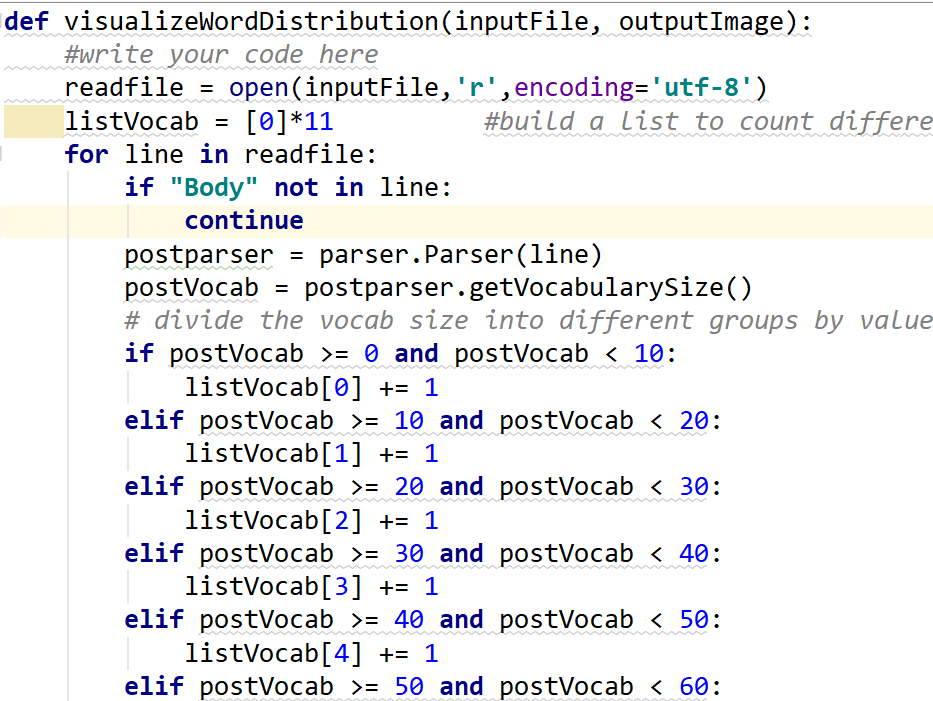
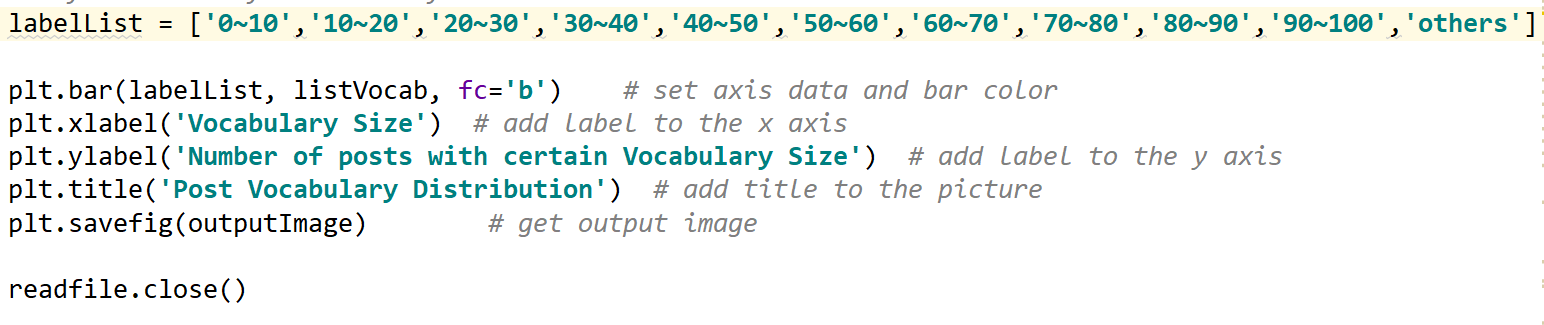
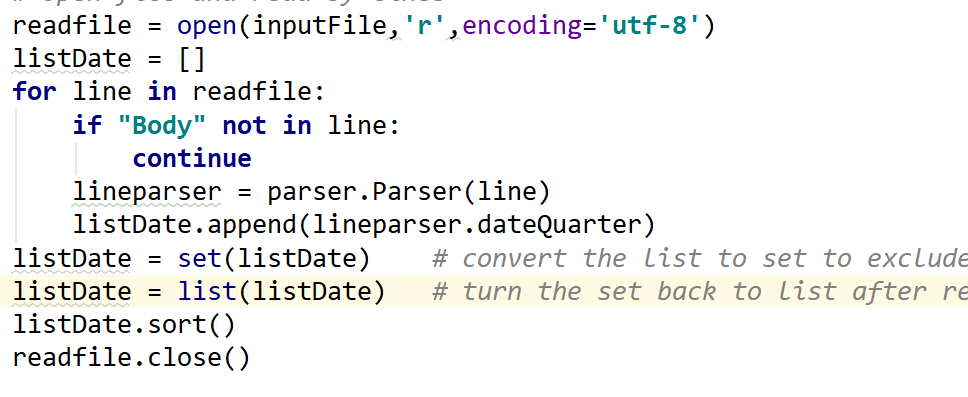


Figure 3.1 visualizeWordDistribution functon, group sorting

The visualizeWordDistribution function produces a bar chart of vocabulary size vs number of word count (Figure 3.2) and export the chart as png.

Figure 3.2 Plot of Vocabulary in Bar Chart

The visualizePostNumberTrend function reads in the date and corresponding quater of input file (Figure 3.3).

 Figure 3.3

The function also matches the type of body content of input file to questions and answers (Figure 3.4).

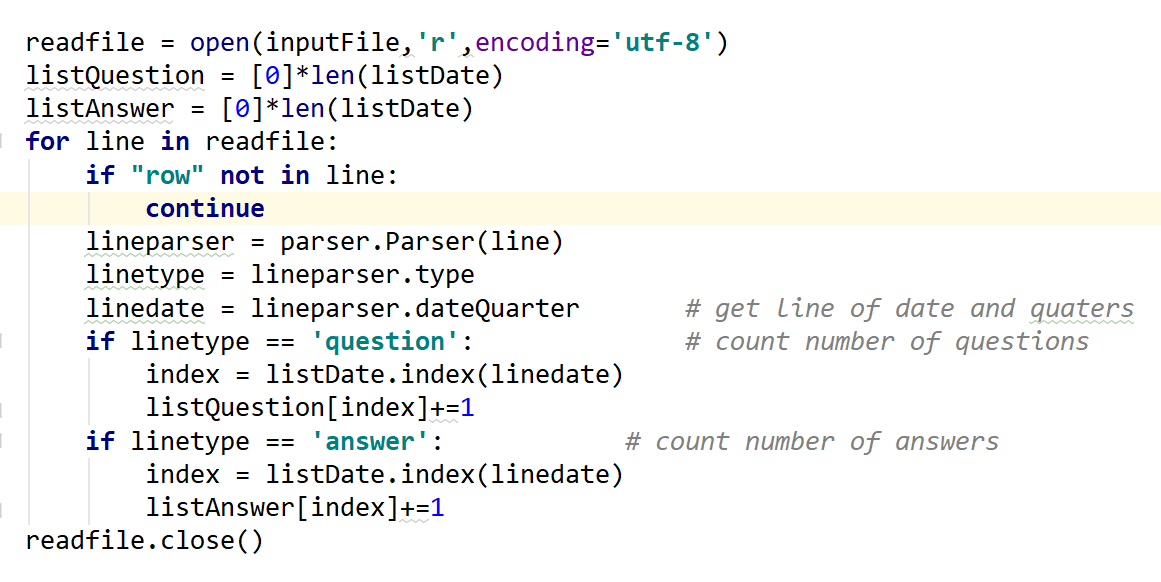


Figure 3.4

Finally, a line plot of numbers of question post and answer post is painted and exported as png.