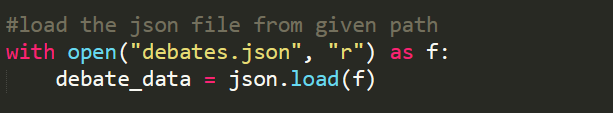
**Documentation for *plotting.py* file:**

The code starts with loading the “debates.json” file using the “json” Python module. The loaded data is then saved in the variable “debate\_data”. The type of this variable is “dict”.

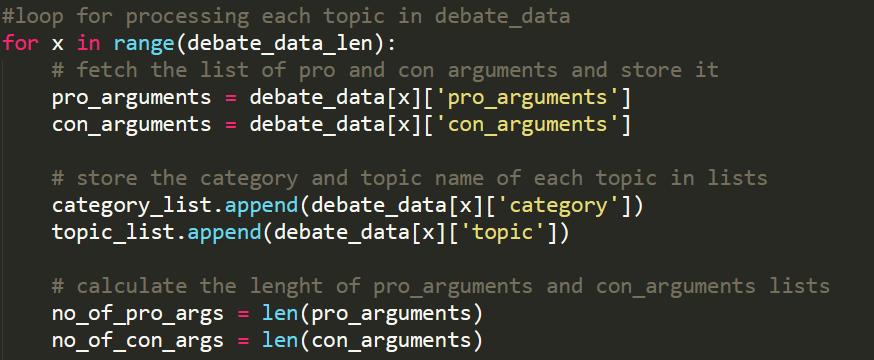


Then we have initialised some empty arrays to store the no. of arguments, no. of words, etc. found in the data.

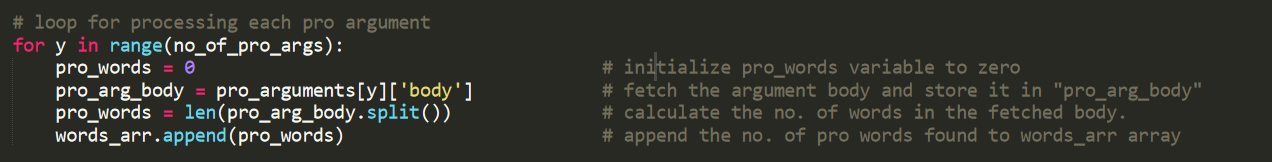
The data crawled by the crawler is store in a list in the json file. The number of topics in the “debates.json” is extracted using the len() method of python on the “debate\_data” variable and stored in “debate\_data\_len”.

Now, we extract the data from each topic by putting it inside a loop.

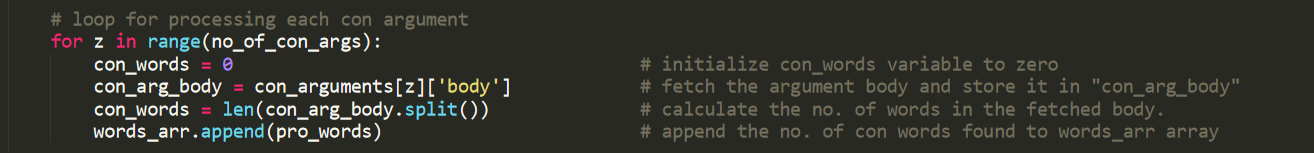
* First the lists of pro and con arguments are fetched and store in “pro\_arguments” and “con\_arguments” variables. Then the *category*, to which this topic belongs to, and the *title of the topic* are appended to “category\_list” and “topic\_list” lists respectively.



* Then, we have calculated the length of “pro\_arguments” and “con\_arguments” lists and stored it in “no\_of\_pro\_args” and “no\_of\_con\_args” variables.
* Now, we processed each pro argument in the “pro\_arguments” list by extracting the body of the argument and calculating the number of words in it. This number is then stored in “pro\_words” variable and appended to the “words\_arr” list.



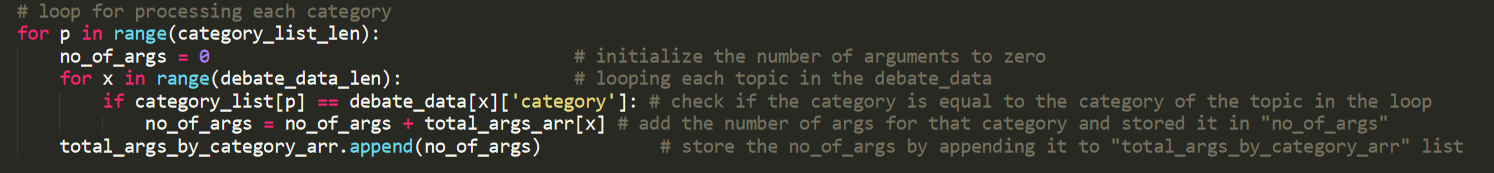
* Similarly, each con argument is fetched and processed, and the number of words of the body is stored in “con\_words” variable and appended to the “words\_arr” list.



* Finally, the total number of words is calculated by adding the pro words and con words found and is then appended to “words\_arr” list . Similarly, the total number of arguments is calculated by adding the number of pro\_arguments and con\_arguments and is then appended to “total\_args\_arr” list.

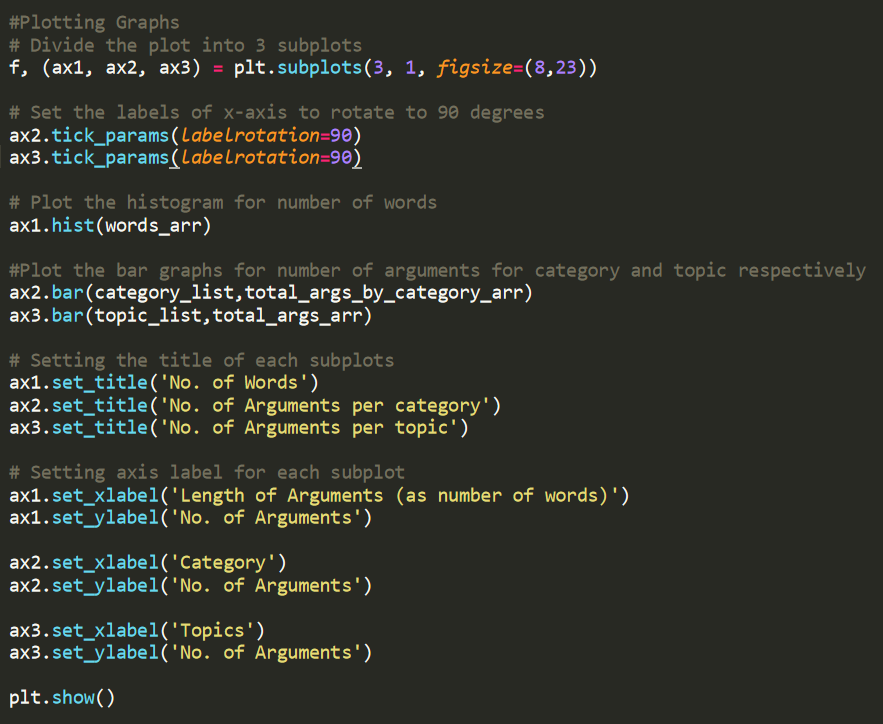
After exiting the loop, we processed the “category\_list” list. First, the duplicate elements are removed using the unique() of pandas library. Then the length of the newly generated unique list is calculated using the len() method.

Now, we process each category to find the number of arguments for each category by using loop. The number of arguments found for each category is appended to “total\_args\_by\_category\_arr” list.



**Plotting graphs:**

* Use the plot from mathplotlib and divide the plot into subplots of 1 column and 3 rows and assign it to 3 axes.
* In the first subplot, we have plotted the histogram of *number of words per argument*. The x-axis gives the length of arguments (number of words per argument) and the y-axis denotes the number of arguments.
* For the second subplot, we have plotted a bar graph of number of arguments per category. Here the x-axis represents the category and the y-axis denotes the number of arguments.
* Finally for the third subplot, we have plotted a bar graph of number of arguments per topic. Here the x-axis represents each topic and the y-axis denotes the number of arguments.

****