Data preprocessing and summary statistics

Data preprocessing

For preprocessing data, we use Beautiful Soup 4, a python library that is used to clean html tags and nltk, to tokenize the word after getting cleaned. . We use normal html parser as the parser for Beautiful Soup. We decided to preserve some of html tags, such as #p, to indicate that it’s the start of the sentence. We also preserve pre and blockquote tag to indicate it’s the end of the sentence as not all sentence use dot to end it. One of the example is:“ Here is an example : <pre>…</pre>”. In this case, our script would only save pre tag as “#pre” as a substitution to dot and discard everything else. The script take the input of raw json file obtained from our extract.py output, then write the preprocessed data into a new json file that has been cleaned and tokenized.

Summary statistics

Statistics

For statistics, we use matplotlib.py plot to plot the histogram for number of answers per question from all of our dataset. Because the question that has answers more than 20 is pretty disperse, their number would be added to 20 answers instead.

As for to get the total questions and total answers from all of the posts, the script will check each of the post, get the answer\_count data from the json for total answers while counting the number of questions checked. Then it will store the data into txt. On the next post, it will read the previous txt, add the previous numbers of questions and answers with the current numbers, and re-store it into the same txt.

To get the length of each thread, the script will check the input, use regex to find each valid word from the sentence, then add it into the list. If the script found another question id, then

It will store the total length to txt, reset everything and keep iterating until the last file. After that, it will store the total word length for all of the data into the same txt.

Stemming

The script will output 2 txt files, one which is before stemming and one is after stemming.

The script will search for eligible word that is a valid word and not a stopword, then the word will be appended to the new json file or will be stemmed first before adding it to the json file if the parameter after is set to True. The script then will count the most frequent words that appears in the json file. The output will show 20 most frequent words that are used in the dataset.