**Project 1 Final Report**

Planned Experiments

In this project, I addressed the question of what the most common words on the Internet are. I decided to approach this question using different datasets that contained words found on the Internet as well as their frequencies. In order to provide an answer to this question, I broke my project down into three separate experiments. My first experiment was intended to be the most direct answer to the question at hand, as it displays a frequency table and a bar graph of the 25 most frequently used words on the Internet. Additionally, I decided it could be useful to generate a word cloud. For my second experiment, I shifted the focus more towards the individual datasets rather than the combined one so that I could analyze the differences between each dataset. In this experiment, I focused more on the length of the words in each dataset, and I created histograms of the length of words for each dataset. Finally, I calculated and displayed the average word length for the top 25 words within each dataset. Finally, the third experiment attempted to shift the focus away from words such as “the”, “and”, “like”, etc. by eliminating words less than six letters long from the merged dataset (in hopes that it would leave words that have a little bit more context when left by themselves). After this, I calculated and displayed how many entries were eliminated and then created another bar graph of the new top 25 most frequent words*.* Lastly, one change I made to my experiments was that I added a column for relative frequency of words for both the combined dataset in experiment 1 trimmed combined dataset into experiment 3 (relative frequency is frequency divided by total frequency). I then printed the top five relative frequencies for both.Finally, I also added the overall mean of word length for each dataset into experiment two for more context instead of just the means of the top 25 words in each.

Data Sources

I was originally unsure of what kind of datasets I wanted to use and how I wanted to procure them in order to best answer the provided question. I considered scraping the homepages of different online news outlets and getting the frequencies of words from there, but ultimately, I decided to use larger datasets that already had words commonly found on the Internet and their frequencies (as well as other information). My first data source comes from <https://www.wordfrequency.info/samples.asp> , which contains a dataset of the top 60,000 lemmas (or words) on the Internet based on data from the Corpus of Contemporary American English. However, this site only allowed me download about the top 5000. The next data source is <https://www.kaggle.com/datasets/rtatman/english-word-frequency> , which provided me with a csv file of the 333,333 most used English words on the Internet according to the Google Web Trillion Word Corpus. My third and final data source that I used is from <http://norvig.com/google-books-common-words.txt> , which contains close to 100,000 of the Internet’s most common words. I was originally planning on using a fourth data source from <https://github.com/filiph/english_words/blob/master/data/word-freq-top5000.csv> , but I ultimately decided against it because it was a smaller dataset in comparison to the others and from what I could tell, it did not appear that it would change the overall data very much. Once I had the each of datasets downloaded into their own csv file on my local machine, I had to do some cleaning/transformation on each of them for later ease of use. For the first dataset, there were many columns of data that I was not interested in, such as part of speech, dispersion, and frequency of the word within different categories. Because I was strictly interested in the words and their frequencies, I modified the original dataset by deleting everything except for these two columns. For the Kaggle dataset, the downloaded csv was just a column for word and a column for count, so I did not have to remove any data. However, I did change the column name from count to frequency so that it would match the other dataset and I would not have to remember different names of columns between each dataset. Finally, for the Google Books dataset, it was originally downloaded as a txt file, so I had to copy and paste it into a csv file as two separate columns. After that, I had to add the column titles of word and frequency to match the other two datasets.

Results/Code

After importing the necessary libraries, I loaded in the three different datasets using the pandas read\_csv() method. Because the data from the Google Books dataset was capitalized, I changed all of its words to lower case using str.lower(), so that the words could match with those in the other datasets. Next, I combined the three datasets into one larger dataset using the merge() method and adding the frequencies then only displaying the word and new frequency columns. When merging in the third dataset, I used groupby() to combine some of the duplicate words and their frequencies. Following this, I used the files library to download the combined dataset (which I first sorted by frequency) to my local machine. I did all of this in a separate file so that I could then just import the large, combined dataset into the file with the experiments similar to how I did with the other three datasets. The actual project1 file starts the same way as the file used to merge the datasets, as I still had to import the needed libraries and then load in the original three datasets and the new merged one.

The next section of the code relates to experiment 1 and starts by sorting the merged dataset based on descending frequencies (most common words at top) using sort\_values(). I then displayed the frequencies of the top 25 words using head(25). Similarly, I created a bar graph of those 25 words and their frequencies using plot.bar with word as the x value and frequency as the y value (used methods from matplotlib.pyplot to create labels and title).

Chart, histogram

Description automatically generated

From this bar graph, we can see that some of the most common words are “the”, “in”, “of”, “and”, and “that”, with “the” and “in” having around double the frequencies of the next few words (“the” is slightly more common than “in”).Up next, I created a word cloud based on each word’s frequency with the following code:

wordFrequencies = dict(zip(sortedCombined['word'], sortedCombined['frequency']))

cloud = WordCloud(background\_color='white').generate\_from\_frequencies(wordFrequencies)

plt.imshow(cloud, interpolation='bilinear')

plt.axis('off')

plt.show()

This gave me the following image:

Text

Description automatically generated with medium confidence

For the last part of this experiment, I created a column for relative frequency in the combined dataset by diving each word’s frequency by the sum of all of the frequencies. I then printed the top 5relative frequencies in this dataset, which showed the most popular words, “the” and “of”, with relative frequencies around 0.045.

After this, I moved onto experiment 2, where I started by adding a column for word length to each dataset and defining it as dataset[‘word’].str.len(). I then created a histogram of word length for each dataset and added labels using code like this:

wordData1.hist(column="length", bins=20, range=[0,25])

plt.xlabel("Word Length")

plt.ylabel("Frequency")

plt.title("Distribution of Word Length for Dataset 1")

plt.show

This code produced several histograms, the first of which looked like this:

Chart, histogram

Description automatically generated

From the three histograms (other two can be seen from code), we can see that each dataset has a similar distribution. They all appear to be unimodal and slightly skewed to the right with a median likely around 7 characters per word. After this, I printed the average word length for each dataset using dataset[‘length’].mean(), and I found that they each had an average in the 6-8 letters range. The shortest mean length was the third dataset (also had the least entries) around 6.4 characters per word. The other two datasets were a bit higher, with the second dataset (Kaggle) having an average close to 7.5 characters per word, and the first dataset (Google Books) having the highest average at about 7.6. From these averages, I inferred that there is a good amount of variation in the length of the words in the datasets, as I know that there are many shorter words near the top. From there, I went through a similar process but sorted each of the datasets by highest frequencies using sort.values(), and then displayed the mean of the 25 most frequent words of each dataset using:

sorted['length'].head(25).mean()

With the means of the 25 most common words, we see much different data. Each dataset has an average close to 2.5, with the first dataset at 2.52, the second at 2.6, and the third at 2.68 words per character. Clearly, this tells us that some of the most common words on the Internet tend to be very short (this makes sense based off of words such as “the”, “and”, “at”, etc.).

Finally, I moved to experiment 3, which starts with getting the total number of words in the combined dataset (using the count() method). After this, I created a new dataset that eliminated all words that were less than 6 letters long from the combined data using the following code:

trimmedLengthSet = combinedData.drop(combinedData['length'] < 6)

I then calculated how many words were eliminated by using count() on the trimmed dataset and subtracting that from the original total. From this calculation, I saw that over 85,000 entries were eliminated through this line of code, which is a sizeable chunk of the data. Clearly, we can see that a notable percentage of the most common words on the Internet are shorter words.I then sorted the trimmed dataset based once again on highest frequencies, and then created a bar chart of what are now the 25 most common words after getting rid of shorter words (process was similar to bar chart for the experiment 1, but I will show the code below this time).

trimFreqPlot = sortedTrimmed.head(25).plot.bar(x='word', y='frequency', rot=90, title="Most Common 6+ Letter Words", legend=False)

trimFreqPlot.set(xlabel='Word', ylabel='Frequency')

print(trimFreqPlot, "\n")

**Chart

Description automatically generated**

From this bar graph, we can see that the most common longer word in the dataset is “before”, closely followed by several others such as “general” and “search”. We can also see that there is not as steep of a drop off after the first few words as there was in the original bar graph of most common words. Finally, I created a new column in the trimmed dataset for relative frequency, which I calculated by dividing frequency by the total frequency (as mentioned above). This list of relative frequencies made it clear that none of the longer words appear super often in the grand scheme of the whole dataset, as the top relative frequency of the trimmed dataset was around 0.005.

Conclusion

Overall, I saw that “the” was the most common word on the Internet (as many people would expect), closely followed by “of”, and next words such as “and”, “like”, “by”, and more. From the experiments, we can see that both the average and median word lengths of common words on the Internet tends to hover around seven letters, but the average for the most common words is closer to about three. One of my main takeaways from this project was the idea that you do not need to memorize everything in data analysis and manipulation/pandas because if you do not remember the name of a method or how to use it, it is usually somewhat quick and easy to figure out (whether it’s just a quick Google search or going through Stack Overflow, you can usually sort it out eventually). However, there are certain things that it definitely helps to have a good grasp on and remember, such as the head() method (which I found myself using very often), how to sort values of a data frame, and how to add or manipulate columns in a data frame. Another thing that I learned from this project is how it can sometimes be very difficult to combine more than two datasets into one if you do not have a lot of experience doing so. I spent a large portion of my time on this project trying to get the merge() method to work when merging in the third dataset. I had little to no issues merging the first two datasets, but when I tried to add in the third, I kept getting duplicate words. I eventually used the groupby() method which fixed this issue, but I quickly realized that this prohibited me from using DataFrame methods on my combined dataset because the groupby() method changes the type by returning a DataFrameGroupBy object. As a workaround, I realized that I could copy my code for merging the datasets into a new file, download the combined dataset to my local machine, and then read it as a new dataset in my actual project file. As a whole, this project helped me realize how efficient Python and Pandas can be when it comes to analyzing datasets in different ways and displaying your findings to other people in ways that are easy to comprehend. It also helped me sharpen my skills when it comes to thinking about different ways to compare similar datasets.

Websites/databases used (once again):

* <https://www.wordfrequency.info/samples.asp>
* <https://www.kaggle.com/datasets/rtatman/english-word-frequency>
* <http://norvig.com/google-books-common-words.txt>