Nick Videtti

nvidetti@syr.edu

Assignment 3

Document Vectorization and Visualization of CSV Data Using Pandas DataFrames

SciKit Learn CountVectorizer, SciKit Learn TfidfVectorizer, Word Clouds, PyPlot

4/25/2023

| Word Clouds By Category | |
| --- | --- |
| **Introduction** | The Brown Corpus is a collection of text that was created at Brown University in 1961. It was the first million-word electronic corpus and contains text from hundreds of sources. Each document in the Brown Corpus is classified into one of 15 different categories: adventure, belles-lettres, editorial, fiction, government, hobbies, humor, learned, lore, mystery, news, religion, reviews, romance, and science fiction.  Given the size of the Brown Corpus, and the corresponding classifications of each document, it can give a good approximation of written English language as a whole including categorization, at least for the 15 categories in the corpus. Using Word Clouds, the most common words in each category of the Brown Corpus can be found and visualized.  For some background on Word Clouds, they are graphs that have words as their data points. While that is great in and of itself, the most useful thing about World Clouds is that the sizes of the words are based on the frequency of the corresponding word. That is, the most frequent word will be the largest, the second-most frequent word will be the second largest, the least frequent word will be the smallest, etc.  Some specifics of these Word Clouds in this research are as follows. All words will be in lowercase, only words that have exclusively letters will be included, and stop words will be excluded. For the first two points, this means that “Dan” and “dan” will both be “dan”, “The” and “the” will both be “the”, “didn’t” would be either excluded or treated as “did”, etcetera. The last point however may require some additional explaining. To clear up any questions, stop words are words that do not usually change the meaning of a phrase. They are usually short and included in the phrases simply for grammar reasons. Some of the stop words that will be excluded from the Word Clouds are “i”, “we”, “myself”, “are”, “was”, “were”, “be”, and roughly 170 other words.  Three random categories of text from the Brown Corpus will be selected and Word Clouds will be created for each category. |
| **Analysis** | The Brown Corpus will be accessed via the Natural Language Toolkit (NLTK) in Python, specifically nltk.corpus.brown.sents in Python 3.10.6. Data will be stored as a list of lists, with each “nested” list containing a sentence from the corpus as one element and its corresponding category as the other element. From there, the user will be prompted to save the data in a directory of their choice and under any file name, albeit the expectation of saving as a “.csv” file is extremely necessary. Using the same file that is saved from the previous step, data will be read into another list which will be used to create a Pandas DataFrame.  The Pandas DataFrame will use the list of read-in data to store all relevant data and become the main data source for the analyses. Columns will include “CATEGORY”, “DOCUMENT”, “WORDS”, “CLEANED WORDS”, and “CLEANED DOCUMENT”. The “CATEGORY” and “DOCUMENT” columns are what was read-in from the data file, which are just the category of each sentence and the sentence itself, respectively. The “WORDS” will be the sentence/document split into words, which is done using NLTK’s word\_tokenize algorithm. “CLEANED WORDS” is a transformation of “WORDS”, specifically converting all words to lowercase, removing stop words using NLTK’s English stop words in nltk.corpus.stopwords.words(‘english’), and keeping only words that contain exclusively letters. Finally, “CLEANED DOCUMENT” is simply “CLEANED WORDS”, but converted from a list and combined into a single string with a space between each word.  Once the data are prepared, document vectorization will be the next step. For that, the list of distinct categories will be randomly shuffled, then the first three categories will be selected from to use for the Document Vectors and Word Clouds. For each of these three categories, the “CLEANED DOCUMENT” column from the DataFrame will be used to create document vectors using both SciKit Learn’s CountVectorizer and SciKit Learn’s TfidfVectorizer. This will result in 6 total sets of document vectors that will be used to generate 6 different Word Clouds.  The Word Clouds will be generated using wordcloud.WordCloud, and will be visualized using PyPlot. All 6 Word Clouds will be graphed on the same figure in the Python script, but screenshots in the “Results” and “Conclusions” sections will be separate for each Word Cloud in the best interest of size and readability. |
| **results** | The resulting Word Clouds for one iteration were as follows. |
| **conclusions** | The three random categories selected for one run-through were “hobbies”, “editorial”, and “science fiction”. Each category had two different Word Clouds created, one using standard word counts, and one using what is called “Term Frequency - Inverse Document Frequency”, or simply “TF-IDF”. TF-IDF essentially uses word frequencies in combination with word relevance based on those frequencies. In theory TF-IDF should give a more valuable and representative Word Cloud than the standard word frequency Word Clouds. The resulting Word Clouds were as follows.              It is seen that many different insights can be made from just these 6 Word Clouds. For example, the “hobbies” category tends to include words like “one”, “new”, and “first”, and “time”. “Editorial” category tends to include “one”, “editor”, “people”, and “united”. “Science Fiction” category tends to include “one”, “time”, “said”, “ekstrohm”, and “mercer” (the latter two are likely assumed to be the names of fictional characters). An interesting insight that sticks out like a sore thumb is the high prevalence of the word “one” in every single Word Cloud. In future experiments, the word “one” could be simply excluded, but it would be much more interesting to see what the root cause of its dominance is. Is the wrong set of stop words being used to filter? Is the word “one” just used with extremely high frequency across the entire Brown Corpus? Or even more interestingly, across all written English language? Are there any other words that have a relatively high frequency across all categories in the corpus? Is the word “one” only this relevant across these three specific categories? Potentially making Word Clouds for all 15 categories and for the entire Brown Corpus as a whole would be a good way to investigate during future research. |