Observations on the Datasets

Difficulty of Rules-Based Classification

The initial takeaway I get from looking at the dataset is that is seems specifically tailored to be difficult for a rules-based system to classify. Many of the non-antisemitic tweets have words typically used by antisemites to express antisemitism and the vocabulary overlaps between the antisemitic tweets and the tweets that simply mention Judaism. There are a few words that appear to only come up in the antisemitic tweets, such as mentions of George Soros, but those words appear in few tweets and even in the case of filtering out mentions of Soros, that would also affect innocent mentions of him.

In the directory labeled "data", there is a subdirectory labeled "counters" that have files that coule be useful for comparing the frequency of different words in tweets that were identified as being different types of antisemitic. The file's title gives the type of antisemitism (a number 1 through 4, corresponding to political, economic, religious, or racial antisemitism), and in the file is a list of words next to a number. This number is the frequency of that word in the dataset of that type of antisemitism subtracted by the frequency of the same word in the non-antisemitic tweets, to give a view of which words were more or less common in these datasets. Looking at the words, it seems as if none of those that are significantly more or less common appear to be necessarily antisemitic words, so just using the appearance of these words and nothing else would likely not be very useful for classification. Instead, I decided to take a more holistic approach to the rules-based classifier.

Rules-Based Classifier Design

I begin by taking in an input text, putting it all in lowercase, removing stopwords, removing punctuation, and reducing it to a percentage frequency counter of each unique word in the text. For example, `"The big, big brown fox"` would be represented as {`"big": 50, "brown": 25, "fox": 25`}. This process is done for the combined text of all antisemitic tweets and non-antisemitic tweets, as well as individually for the tweets of each of the four types of antisemitism. I stored the resultant frequencies in a subdirectory of the "rulesbased_model" directory entitled "frequencies." In order to determine if a text is antisemitic, I do this same basic process to represent it as a series of frequencies. I then read the frequencies of the antisemitic and non-antisemitic datasets, removing any words from those datasets that do not appear in the input text and adding a new row with a frequency of 0 for words that appear in the input text but not the dataset. After that, I order the words alphabetically in all three `DataFrame` objects so they are all in the same order. Then, I stored the second column of the `DataFrame` objects (the one containing the frequency percentages) in three vectors, one for the input, and the other two for the two datasets. I then got the distance between the input vector and each of the two dataset vectors, and chose whichever dataset had the smaller distance as the one that more closely represented the input data.

For example, for the counter {`"big": 50, "brown": 25, "fox": 25`}, the antisemitic dataset would have a counter {`"big": 0.056159, "brown": 0, "fox": 0`}, and the baseline would have {`"big": 0.125219, "brown": 0, "fox": 0.008348`}. The next step would be to sort these words alphabetically, but luckily the order they come in in the text is already alphabetical. Next, they would be represented as vectors and the distance would be found between them and the input text, as shown:

$$\begin{bmatrix} 50 \\ 25 \\ 25 \end{bmatrix} - \begin{bmatrix} 0.056159 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 49.943841 \\ 25 \\ 25 \end{bmatrix} = \vec{a}, |\vec{a}| \approx 61.19$$

$$\begin{bmatrix} 50 \\ 25 \\ 25 \end{bmatrix} - \begin{bmatrix} 0.125219 \\ 0 \\ 0.008348 \end{bmatrix} = \begin{bmatrix} 49.874781 \\ 25 \\ 24.991652 \end{bmatrix} = \vec{b}, \left|\vec{b}\right| \approx 61.13$$

Since the vector representing the non-antisemitic dataset is slightly closer to this input text, it would be classified as not antisemitic. The distances between these vectors becomes smaller as larger texts are inputted with frequency distributions that more closely match those of the datasets.