**Classifying Characters with Disabilities in Fiction from the Words Most Associated with Them: A Pilot Project**

**Research Question**

My research question was relatively simple to frame, but difficult to answer: can we build a classifier that reliable identifies characters in fiction with disabilities based on the words associated with them? A follow-on question was also simple: what do we learn when analyzing these words for both classes of characters?

**Definitions**

For this project, I am using physical disabilities—varyingly manifested in the literature as lost limbs, paralysis, extensive burns or physical disfigurement—and a wide definition of mental disability ranging from intellectual disability such as Downs Syndrome, autism spectrum disorders, post-traumatic stress disorder, schizophrenia, and bi-polar disorder. For mental disabilities, things are more nebulous, both because I used the DSM-5 as a guiding framework, and because I was hoping to see how the classifier might react with such a wide definition. While I’ve had some training in disability studies and special education, I am not an expert, and sought the most inclusive definition for this project, which also helped with data gathering.

However, there was one notable inclusion of Morton in Charlotte Perkins Gilman's novel *The Crux*, who fits none of the above, but instead has a permanent venereal disease. I left Morton in though, as *The Crux* is a powerful look at eugenics, and the idea of “breeding out” undesirable conditions. So, though he contradicts my definition, he is clearly a figure of disability in the novel, and representative of a more targeted, philosophical novel I didn’t capture in my other sources.

**Data**

A huge amount of the work in gathering data was manual on this project, a both comforting and frustrating aspect, at times. I first needed to do a lot of web searching for books that feature characters with disabilities, and then pick from those results which books 1) were available in the ~94,000 volumes for which Dr. Underwood had character data for and 2) were from, where possible, more adult fiction. A lot is written about disability in young adult fiction, but I was more interested in adult fiction, which is often less simplistic with regard to themes, and is likely absent from the overt moral content of children’s and young adult books that feature disability. This search was long and tedious, but, in the end, yielded decent results for a project of this scope and rigor. The second step was to get the data and to then identify, again manually, the characters that had disabilities within the volume. This project was worked on over a few weeks, and as such, I often had to reinvestigate which characters drew me to include a volume in my corpus. Further, I then needed to cross-check the character name as I knew it with how it manifested in the text, which was another tedious process that involved printing all names from that volume’s rows of the dataframe, and updating an offline spreadsheet with their name as it appeared, and whether or not I had added a 1 to their column of “has\_disability” (about which I’ll speak more in a few paragraphs). However, the nice part about this manual work is that it was easy to save, and to create data from which I could start my analysis all over again with little fuss.

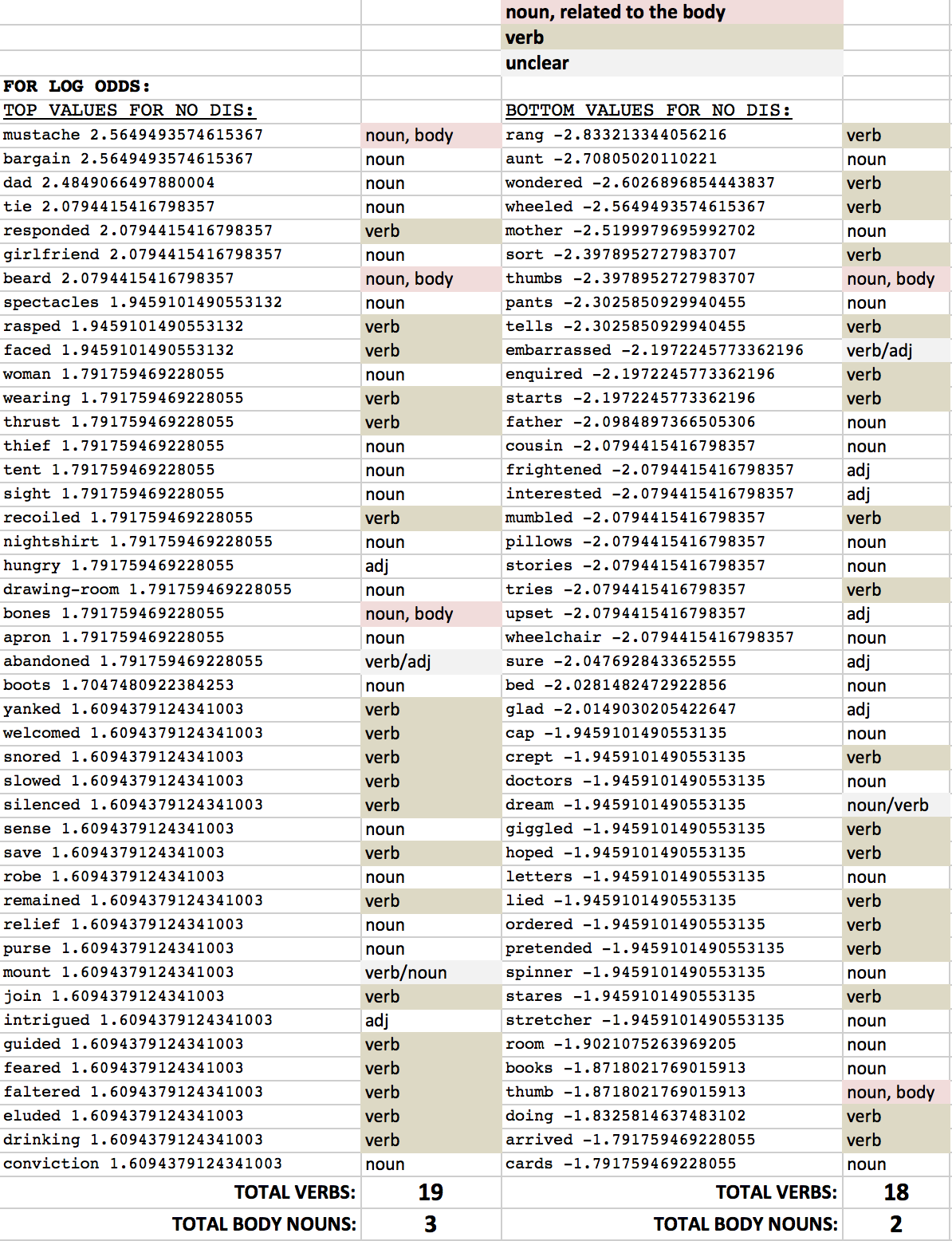
Data was ingested from a .csv into a Pandas dataframe, and superfluous rows were dropped, the first an arbitrary index that was unnecessary for this work. I had initially hoped to manipulate the data in Python to add a column for “has\_disability,” a binary used in training along with adding a column for “bookname” which would help me populate the former. However, I was having trouble with both getting my data to persist after multiple sessions of coding and getting the data to accurately update, so I decided to manually add the “has\_disability” column and programmatically add a column for book names, based on the “bookid” column.

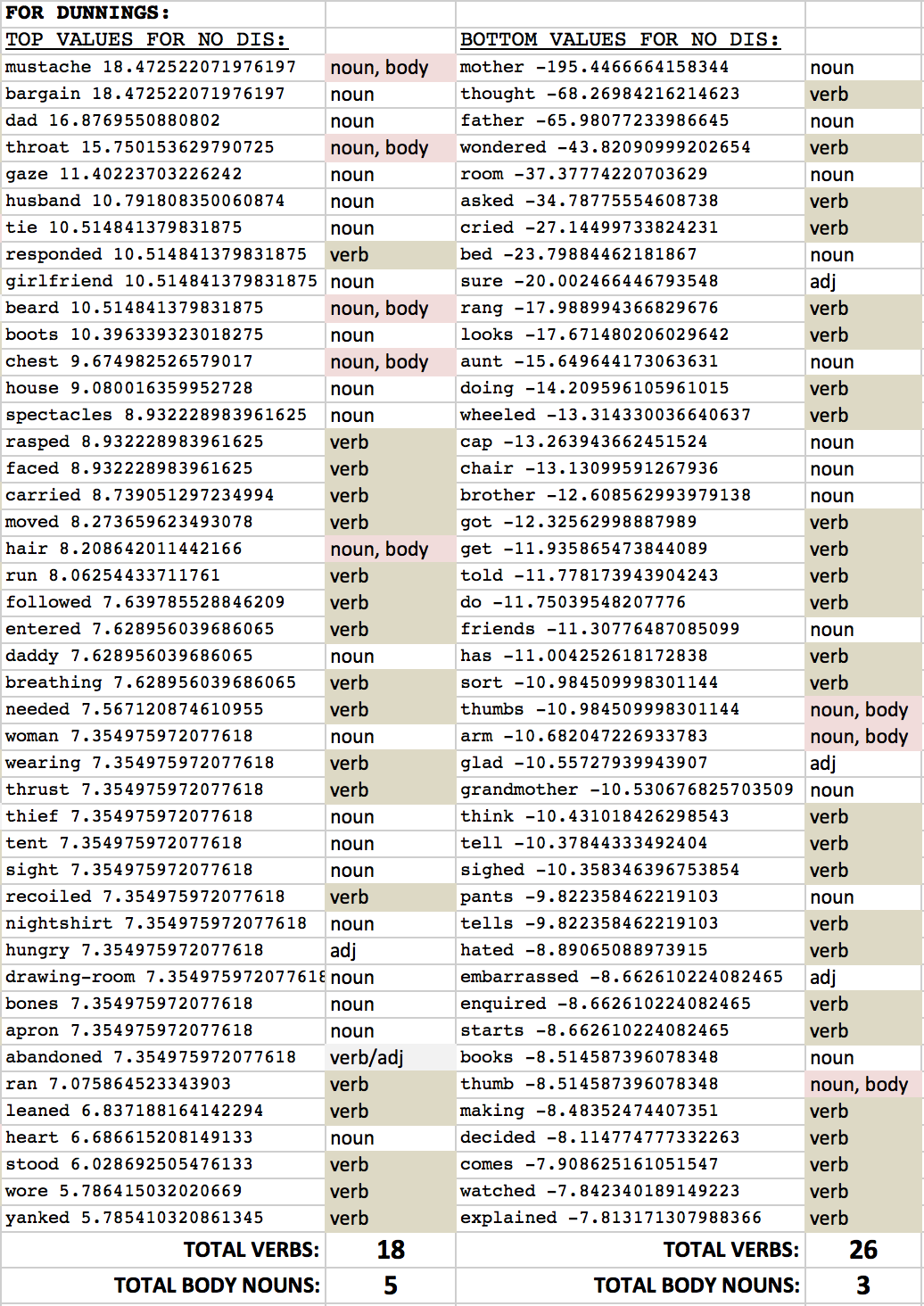
In order to make sure I didn’t skew my classifier with disparate amounts of data for either class, I culled the data for characters without disabilities to 65 rows, taken as a random sample, based on rows that would equal a total word count as close as possible to the total for the 54 characters with disabilities. In the end, this was a total of 119 characters, with 7,079 words associated with characters with disabilities and 5,819 words with characters without. While not ideal, this was only 22% more than characters with disabilities, and the greater number of characters without disabilities (itself a 20% higher number in favor of characters without disabilities), I hoped, would help offset this disparity at least somewhat.

**Analysis**

I then went about training my naïve Bayes classifier using 5 folds of data, with folds randomly assigned. I held back one-fifth of my data for testing and used the other four-fifths for training. This produced a classifier that was able to correctly classify characters, after five-fold cross-validation—randomly switching training and test corpus over all five folds and testing the classifier’s results—an average of 92% of the time. Given the varied types of disabilities, I am pleased with this result, and it hints that large-scale running of this classifier against the full fiction corpus for which we have character data could be a fruitful analysis.

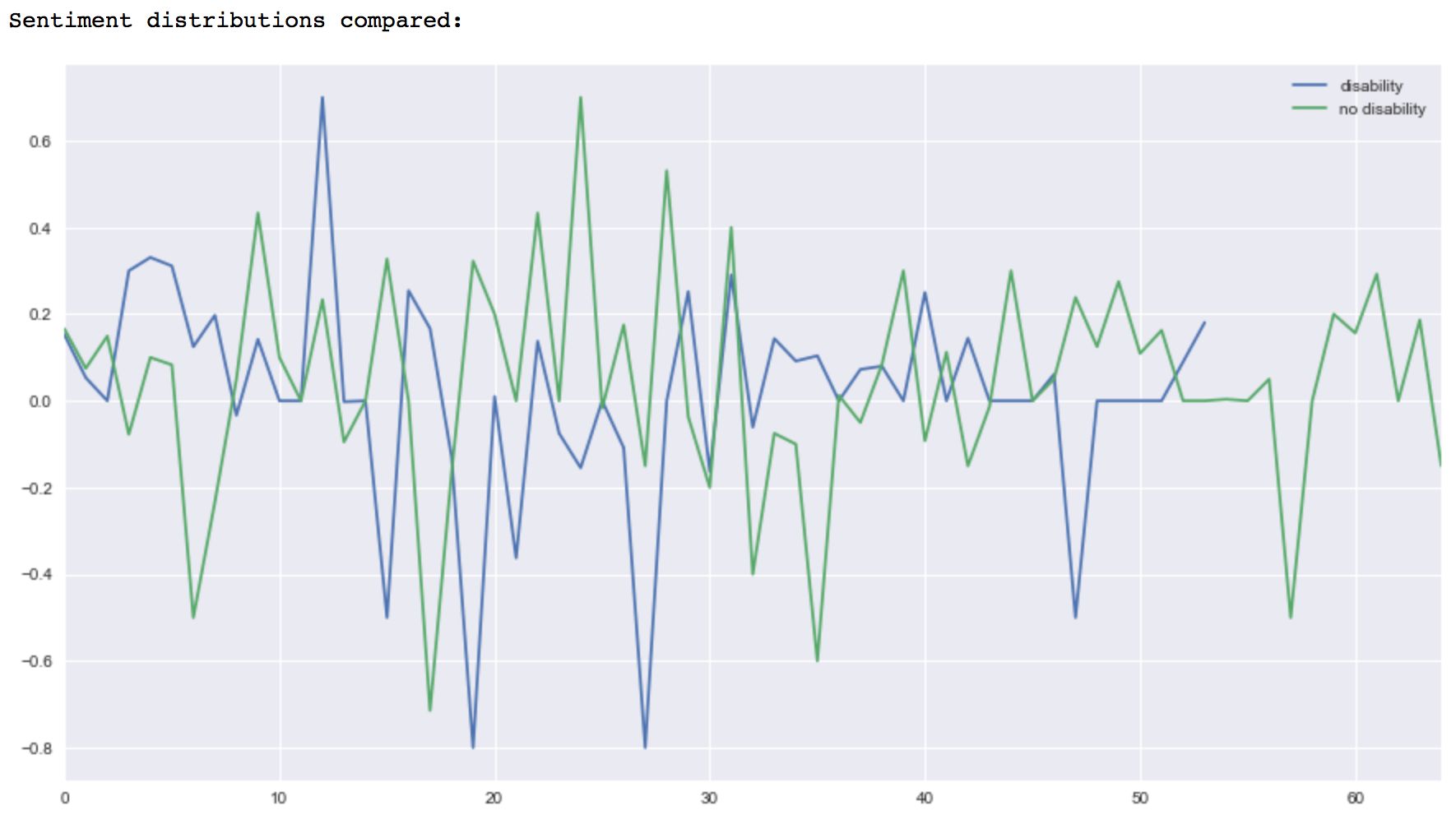
Beyond the classifier, I decided to take a closer look at the most under- and overrepresented words for characters without disabilities, and see if I could glean more from the specific words. I found a few surprising things, but here’s a screenshot of the 50 most over and underrepresented words, per both Dunnings log likelihood and straight log odds. Remember, the words that are negatively correlated will be words most overrepresented for characters with disabilities:





I’ve highlighted a few things of interest in the images above. First, surprisingly, nouns referring to the body were more represented for characters without disabilities. This could be seen as a hint that characters with disability didn’t have a lot of words devoted to describing their disability, but could also be a hint about less physical descriptors in general. Also quite a surprise is that more verbs are associated with the characters with disabilities. I was expecting to see fewer verbs, relaying less action, for characters with disabilities. Instead, the opposite seems to be true, with the classes even in log likelihood and characters with disabilities having 44% more verbs in the top 50 words than those without. It’s hard to say what this means for characters in fiction, but it’s unexpected, especially given some of the adversarial, or non-heroic characters in my corpus (the full list is available in the project folder on GitHub, as “dis\_df.csv”). I’m hesitant to make any broad claims due to issues with bias—as all the characters were manually identified, and the volumes were chosen based on the inclusion of character(s) with disabilities—and without looking more closely at classifying the words themselves with part-of-speech tagging, which this project ended up not affording me the time for that analysis, sadly. Nonetheless, both of these discoveries were surprising, and might signal that the division between words describing the two classes of characters is not as stark as I had expected.

Less surprising was my sentiment analysis, though. In this, traditional assumptions about depictions of disability in media were affirmed. Sentiment for words associated characters with disabilities was 2.5 times more negative than for characters without. Distributions for both were relatively varied, but sentiment for characters without disability hovered more closely around the slightly positive point of 0.2, while for characters with disability, sentiment as closer to 0.0.

Given the above, it is less surprising that objectivity—TextBlob’s metric for evaluating it’s own confidence in the sentiment score—was lower for characters with disability, whose average sentiment was far closer to neutral, a range of scores more difficult for the algorithm to identify. With characters without disability having much higher average sentiment, their objectivity ratings were higher:

**Conclusions & Future Work**

This project was necessarily exploratory in nature, and I hoped it might serve as a proof-of-concept that characters with disability might be programmatically identified accurately, which I think I’ve shown. This has moderate implications for identifying characters in the whole fiction corpus, and especially for the characters that have already been extracted using Book NLP. This would be interesting for a number of reasons, not least of all because it seems that we are unclear of the number of characters in fiction that have disabilities, let alone their names, and the books in which they appear. Further, we might find characters not traditionally seen to have disabilities associated with the same words that characters we know do have disabilities. This could be caused by a few things, but it might help better identify a group of people that are often described using discrete, coded words, or whose disability is often ignored all together. Having a corpus of books with characters that have been identified as having a disability along with character data similar to that I used on this project for each, would be a valuable resource, and would open the depictions of disability to much broader and deeper inquiry.

The findings on sentiment are in-line with the thought that depictions of people with disabilities in media are typically unkind. But I’d like to see how this might change over time, if at all, and see how the most negative or positive words might also change. Time, in fact, is a factor I’d love to better include, in this study and a larger one, to see how things are affected. It’d also be interesting to see how sentiment is affected by folding in more and more characters and words. It’s possible that, due to my selection of the books and characters, sentiment was influenced too much, and a wide-scale crunch over many thousands of characters might show this finding to be an aberration.