Probabilistic Detection of Character Voices in Fiction

J. Reeve

December 17, 2016

In James Joyce's novel *Ulysses*, the school headmaster Mr. Deasy quotes Shakespeare in a lecture in financial responsibility to his employee Stephen Dedalus. "[W]hat does Shakespeare say?" he asks, "Put but money in thy purse." As Stephen remembers it, however, this is not merely a saying of Shakespeare's, but one spoken by Shakespeare's infamous character Iago. So while Deasy thinks himself to be quoting the wisdom of the early modern playwright, he is, in fact, quoting the Bard's most notorious arch-villain. This distinction—one between an author and the author's fictional creations—is, it need not be said, crucial to the understanding of literature. It is that which the following experiment hopes to probabilistically detect.

The problem of computational character attribution is one of literary knowledge production. In concrete terms, it is the difference between the sentence "Madam, I never eat muscatel grapes" and its TEI XML markup, `<said who="Edmond Dantès">Madam, I never eat muscatel grapes</said>`. In the first case, a reader familiar with *The Count of Monte Cristo* might recognize it as spoken by Edmond Dantès, in the second case, the reader need not know the work to attribute the sentence to its speaker. This markup allows for answers to a wide range of questions, such as the size of the work's cast of characters, the distribution of character speech, and stylistic properties of an individual character. These are queries that are useful for both close and distant reading—they can provide insight about particular characters and a novel as a whole. They are useful both to the close study of a single work and to the distant study of hundreds or thousands of novels at a time. This markup, although a laborious task for a human annotator, might be generated semi-automatically from stylistic signatures of the character.

1 Experimental Design

The overall design of this experiment is based on Box's Loop, an iterative process for refining a probabilistic model based on its predictive performance [@blei_build_2014 205]. The meta-analysis, then, is one of model selection, analysis, model criticism, and improvement. The analysis itself consists of four steps: chunking, vectorizing, dimensionality reduction, and prediction. Chunking involves the choice of documents, and the modification of those documents to fit a certain length. Each document only contains text in one character's voice, but these documents might be of varying length. Vectorizing is the transformation of those documents into numeric representations, whether through traditional "bag-of-words" term frequency representations or more semantic techniques. Dimensionality reduction transforms those high-dimensional vectors into lower-dimensional ones that are more easily manipulable by the predictive step. Prediction takes the transformed set of vectors and performs probabilistic inference, effectively assigning character voices to each document.

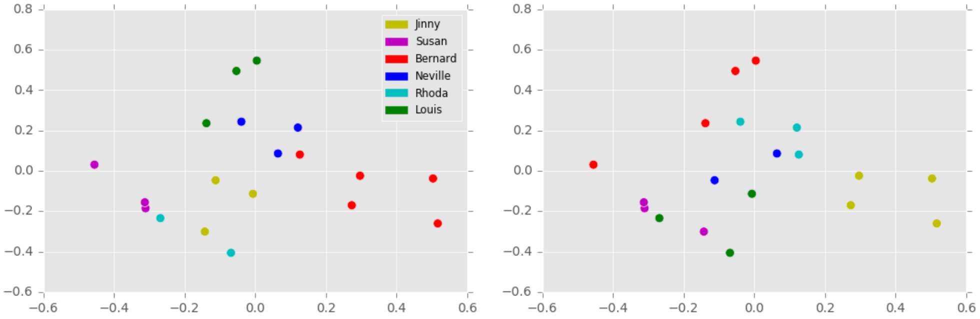
For each of these steps, there are many available techniques and parameters. To identify the best ones, I used a cross-validation grid search that performs a meta-analysis by testing all permutations of these techniques and parameters against an adjusted Rand score comparison of the labeled data with its predicted clusters. This metric accounts for chance, so that a score close to zero indicates a parameter configuration that performs no better than chance, while a score close to 1 is a perfect clustering, identical with the clustering of the original labels, although not necessarily labeled identically.

To test the efficacy of voice detection, we use two TEI XML texts, distant from each other in time and genre: Virginia Woolf's experimental 1931 novel, *The Waves* and Samuel Richardson's classic 1748 epistolary novel *Clarissa*. *The Waves* is notable in that it is almost all monologue spoken by six characters. *Clarissa* is composed almost entirely of letters, mostly from four of the novel's approximately 30 characters. These novels were chosen simply because they were already available in TEI XML format, which made possible the extraction of substantial amounts of labeled text in their characters' voices.

2 Experiment 1: The Waves

A manual, preliminary parameter search showed that character attribution worked best with utterances longer than four thousand characters. Furthermore, very long utterances, such as Bernard's 66,000-character speech that ends the novel, threw off the analysis. With this in mind, I restricted the total 240 utterances of *The Waves* to just the 19 that were between 2000 and 20000 characters in length. The lengths of these documents conform roughly to a normal distribution, with a mean of 6487 characters, and a standard deviation of 1960.

From there, I vectorized these documents using a TF-IDF vectorizer, which counts the frequency of each word, and reweights these according to how frequently they are used in the corpus. I set a maximum document frequency of 30% to ignore corpus-specific stop words, and then limit the vocabulary to the top 500 words (these are paramers suggested by the cross-validation search). The resulting vectors I then reduce to five dimensions using principal component analysis. This 19x5 matrix become the input for probabilistic inference. A two-dimensional projection of this matrix is shown in Figure 1 (a), with each



(a) Initial Labels

(b) Final Convergence

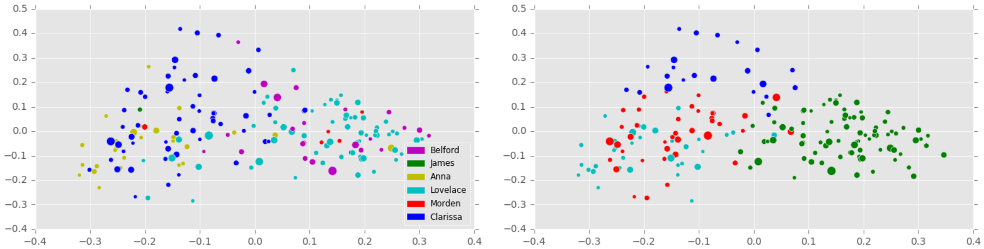
Figure 1: Character Clusterings in The Waves

point representing a single character's utterance. Of note here is the apparent separation in this projection between the male and female characters, with the male characters in the upper right and the female characters in the lower left.

The probabilistic model used for clustering assumes that the dimension-reduced, TF-IDF-weighted word frequencies can be modeled with a mixture of Gaussians. I performed inference using scikit-learn's `GaussianMixture` class, which uses the expectation-maximization (EM) algorithm to cluster the data. Given six components in which to cluster data, the class clustered the data into the six groups shown in Figure 1 (b). The labels aren't identical with the original labels in Figure 1 (b), but the groupings are similar. The inference correctly groups together four out of five of Bernard's utterances, three out of four of Louis's, two out of three of Neville's, Rhoda's, and Susan's, but misidentifies Jinny's. After twenty trials with this configuration, the mean adjusted Rand score was 0.443, with a standard deviation of 0.076—performing better than chance, although not perfectly. The code used for this analysis, written in the Python programming language and using the Scikit-Learn machine learning library, is available at this project's GitHub repository.

3 Experiment 2: Clarissa

After manually tagging and extracting character-labeled letters from Richardson's *Clarissa*, I generated test documents by selecting only letters longer than 8,000 characters and shorter than 50,000 characters. This generated a corpus of 180 letters of varying lengths, the respective lengths of which are indicated by the sizes of the dots in Figure 2. I then vectorized these documents using the top 500 most frequently used words, and reduced the resulting matrix to 25 principal components using PCA, the first two dimensions of which are shown in Figure 2 (a). As in the *Waves* experiment, these were all parameters suggested by the cross-validation grid search. Unlike the Waves experiment, however, the



(a) Initial Labels

(b) Final Convergence

Figure 2: Character Clusterings in Clarissa

grid search suggested a slightly different inference model: a Bayesian Gaussian mixture model. This model differs from the `GaussianMixtureModel` class in that it uses variational inference. Additionally, it doesn't always provide the number of clusters requested, but only no more than those requested.

Curiously, the grid search suggested that we request that the Bayesian Gaussian mixture model provide four clusters, fewer than the number of characters. However, this turned out to have not been an error, since the amount of text represented by the relatively minor characters James and Morden (as evidenced by the paucity of green and red dots in Figure 2 (a) is very small, and it would almost be fair to assume that there are really only four characters represented here.

The final clustering is shown in Figure 2 (b). It incorrectly clusters together the correspondence of villains Lovelace and Belford, but forgivably, since these characters are friends and associates, and at most points in the novel partners in crime. It correctly identifies most of Anna's correspondence, but divides her best friend Clarissa's into two groups: one closer to Anna, and another closer to Lovelace. A closer analysis, which is perhaps beyond the scope of the present experiment, might reveal that those of Clarissa's letters closest to Anna's are letters in fact written to Anna, while her letters that appear closest to Belford and Lovelace might, in fact, be those written to them. In fact, the clustering here, though inaccurate, might be more useful to literary analysis than an accurate clustering: they might reveal not only the character voices themselves, but degrees or modes of these voices. They might show how voice changes according to addressee.

The adjusted Rand score for this clustering is a slightly lower 0.357, faring much better than chance, but still worse than *The Waves*. Although these experiments used different parameters and clustering techniques, this discrepancy might be telling. Maybe the respective scores indicate the degree to which a writer is able to write in the styles of his or her characters. Conversely, maybe this difference indicates the degree to which a writer's choice of characters is stylistically different. If that is the case, novelists with many classes of broadly-painted characters such as Charles Dickens would show higher scores than novelists like Jane Austen who deal with the social subtleties.