# Visualizing James Joyce's Writing Style

Kevin Wang
CS 398 Visualizing Literature
May 13, 2014

My initial goal for this project was to make a visualization for the novel *Finnegans Wake* (1939) by James Joyce. This was the final work by James Joyce, best known for *Ulysses* (1922), and was written over the course of 17 years. Regarded as one of the most difficult works of fiction in the English language, *Finnegans Wake* is largely characterized by incessant wordplay and a stream of consciousness writing style that attempts to emulate the structure and flow of dreams.

I originally considered attempting to extract meaning from the novel using natural language processing, but given the somewhat disappointing effectiveness of currently available NLP strategies, as well as the fact that even the existence of a plot in *Finnegans Wake* is debated, I decided instead to focus on quantifying its writing style. For comparison, I chose to display the results side-by-side with James Joyce's *Ulysses* and a more conventionally written novel, namely T. H. White's *The Once and Future King*.

Rather than making a complex, interactive visualization, I aimed to model my project after a vertical infographic, using a collection of simple, digestible visualizations. To this end, I analyzed each work using three metrics: word count, "Englishness," and lexical novelty.

The word count graph is a simple D3 grouped bar graph. It conveys both the length of each novel as well as the sizes of the vocabularies of each. As expected, *Finnegans Wake* uses a much larger vocabulary than either of the other novels. More surprisingly, it achieves this despite also being the shortest work of the three. I chose a grouped bar graph for this visualization because not only does it allow both the total and unique words metrics to be

compared simultaneously, but also allow the viewer to easily gauge the number of unique words as a proportion of the total length of each work.

Upon reading the first few paragraphs of *Finnegans Wake*, I quickly realized how little the text actually resembles the English language. For example, the following sentence is from the beginning of the second paragraph of the text:

> Sir Tristram, **violer** d'amores, **fr'over** the short sea, had **passen-core rearrived** from North **Armorica** on this side the scraggy isthmus of Europe Minor to **wielderfight** his **penisolate** war: nor had **topsawyer's** rocks by the stream **Oconee** exaggerated **themselse** to **Laurens** County's **gorgios** while they went **doublin** their **mumper** all the time: nor **avoice** from afire **bellowsed mishe mishe** to **tauftauf thuartpeatrick** not yet, though **venissoon** after, had a **kidscad buttended** a bland old **isaac**: not yet, though all's fair in **vanessy**, were **sosie sesthers** wroth with **twone nathandjoe**.

The emboldened words above are all the words that my word processor's spell checker claims are misspelled. To represent the occurrence of non-English words in the text, I included the second visualization on the page, also created using D3, which depicts the proportion of words in each text that appear in the English dictionary. For this, I used the PyEnchant spell checking library. In this case, I chose to compare the words against the British English dictionary because both authors wrote in British English. In addition, because PyEnchant considers case when spell checking words, I did not convert words to lowercase as part of the tokenization procedure for this visualization. The results showed that while 1 in 40 words in *The Once and Future King* were not recognized as English, every fifth word in *Finnegans Wake* is not in the dictionary. This alone appears to be a promising measure of how difficult a fiction text is to read.

The final visualization of the three is a lexical novelty heatmap. Lexical novelty refers to how often a text uses terms that have not previously been used. The introduction of new terms may coincide with the introduction of new ideas or plot elements in the story. For this visualization, I processed the text in chunks of 1,000 words each. Within each chunk, I found the percentage of words that have not previously appeared in the text. I used this percentage data to generate a chronological heatmap of the text, with the beginning of the text at the left and the end at the right.

The design and code for the heatmap was inspired by the LotrProject graphs depicting the sentiment across a number of J. R. R. Tolkien's works. The colors of the strips indicate the lexical novelty of each chunk of text, with green representing the lowest degree of novelty, and red representing the highest. The colors are normalized relative to each other, which is why the chunks in *Ulysses* and *The Once and Future King* with the highest lexical novelty are not colored full-red. In addition, hovering over a chunk reveals a tooltip indicating numerically that chunk's degree of novelty. This visualization was not created with D3, but rather a series of `<div>` elements stacked side-by-side with varying background colors. The HTML code for this is generated by the Python text processing code and loaded into the body of the page using jQuery.

Although this visualization only compared three works, the text processing code can easily be run on any number of arbitrary texts. This would make it possible to create larger scale visualizations that compare large numbers of texts. Furthermore, on such a scale it may even be possible to utilize these metrics to attempt to attribute texts with unknown authors using classification, or to identify common classes of writing styles using clustering.