PoCS What's The Story?

Principles of Complex Systems, Vols. 1, 2, & 3D
CSYS/MATH 6701, 6713, & a pretend number
University of Vermont, Fall 2023
Assignment 14
"We can't both do the zinger"

Due: Friday, February 16, by 11:59 pm
https://pdodds.w3.uvm.edu/teaching/courses/2023-2024pocsverse/assignments/14/
Some useful reminders:
Deliverator: Prof. Peter Sheridan Dodds (contact through Teams)
Assistant Deliverator: Chris O'Neil (contact through Teams)
Office: The Ether
Office hours: See Teams calendar
Course website: https://pdodds.w3.uvm.edu/teaching/courses/2023-2024pocsverse
Overleaf: LaTeX templates and settings for all assignments are available at https://www.overleaf.com/read/tsxfwwmwdgxj. If this link doesn't work, try https://www.overleaf.com/read/tsxfwwmwdgxj#456832

All parts are worth 3 points unless marked otherwise. Please show all your workingses clearly and list the names of others with whom you ~~conspired~~ collaborated.

For coding, we recommend you improve your skills with Python, R, and/or Julia. The (evil) Deliverator uses (evil) Matlab.

Graduate students are requested to use LaTeX (or related TeX variant). If you are new to LaTeX, please endeavor to submit at least n questions per assignment in LaTeX, where n is the assignment number.

Assignment submission:

Via Brightspace or other preferred death vortex.

Please submit your project's current draft in pdf format via Brightspace by the same time specified for this assignment. For teams, please list all team member names clearly at the start.

Semester goal: A paper per text studied, building through assignments.

Three stories to analyze:

- **Pride and Prejudice**
  https://www.gutenberg.org/ebooks/1342
- **Frankenstein; or the Modern Prometheus**
  https://www.gutenberg.org/files/84/84-h/84-h.htm

- **Moby Dick; Or, The Whale**
  https://www.gutenberg.org/ebooks/2701

Data:

For this assignment, the novels have been processed into 1-grams with an attempt to capture all elements including punctuation.

You can use the data below, or what you produced in the previous assignment.

The basic data format is as a time series with one 1-gram per line (links below).

For each story, also linked to below are the rank distributions of 1-grams by counts.

https://pdodds.w3.uvm.edu/permanent-share/pride_and_prejudice_narrativetimeseries.txt
https://pdodds.w3.uvm.edu/permanent-share/pride_and_prejudice_1grams.txt

https://pdodds.w3.uvm.edu/permanent-share/frankenstein_narrativetimeseries.txt
https://pdodds.w3.uvm.edu/permanent-share/frankenstein_1grams.txt

https://pdodds.w3.uvm.edu/permanent-share/moby-dick_narrativetimeseries.txt
https://pdodds.w3.uvm.edu/permanent-share/moby-dick_1grams.txt

Instrument—Shifterator:

- For word shifts, use either:

  The Python package described in Ref. [1].

  Or the D3.js shifterator created by Andy Reagan here:
  https://observablehq.com/@andyreagan/d3-shifterator-v4-preloaded.

  Links to paper versions (arXiv is always best), Github repository, and an exhilarating Twitter feed can be found here:
  https://pdodds.w3.uvm.edu/research/papers/gallagher2021a/.

  (Various Matlab versions made by the Unreliable Deliverator do exist and need to be shared on Gitplaces. A more sophisticated map+list version has long been in development.)

  Github repository: https://github.com/ryanjgallagher/shifterator

  Important note: You will need Python 3.8!

1. Measure the average happiness of each text using the labMT word list with the lexical lens:

$$\mathcal{L} = \{\tau \in \Omega \| h_{\text{avg}}(\tau) \leq 4 \text{ or } h_{\text{avg}}(\tau) \geq 6\} \tag{1}$$

where Ω is the labMT lexicon and τ is a word in Ω.

2. (3 points each)

   For the following T_{ref} and T_{comp} for each novel, generate a collection of word shifts as described below.

   Continue to use the same lexical lens $\mathcal{L}$ as above.

   Pride and Prejudice:

   - Pride and Prejudice:
     T_{ref} = the first 40% of the book,
     T_{comp} = 70% to 75% of the book.
   - Frankenstein:
     T_{ref} = the first 20% of the book,
     T_{comp} = the last 10% of the book.
   - Moby Dick:
     T_{ref} = the whole book,
     T_{comp} = 80% to 90% of the book.

   (a) For each novel, produce word shifts comparing text T_{comp} relative to text T_{ref}. Important: Use the average happiness of text T_{ref} as the baseline (this is not the default in the Python package).

   (b) Interpret the word shifts. Does what you see make sense? Are there any surprises? Are some words being used in what the average person might not think is their primary meaning? For example, "crying" in Moby Dick means yelling, and "sick" can mean "awesome."

   (c) Reverse the comparison: Produce word shifts comparing text T_{ref} relative to text T_{comp}. Now use the average happiness of text T_{comp} as the baseline.

   (d) Comment on any asymmetries you see (the basic word shifts we use are asymmetric).

   (e) Produce word shifts comparing text T_{ref} relative to text T_{comp}. Now use 5 as the baseline reference score (neutral on the happiness-sadness spectrum of 1–9 for labMT).

   (f) Compared to your first word shifts, how interpretable are these ones?

References

[1] R. J. Gallagher, M. R. Frank, L. Mitchell, A. J. Schwartz, A. J. Reagan, C. M. Danforth, and P. S. Dodds. Generalized word shift graphs: A method for visualizing

and explaining pairwise comparisons between texts. *EPJ Data Science*, 10:4, 2021. Available online at https://arxiv.org/abs/2008.02250. pdf