

Text Mining “Re-” in Victorian Poetry

Adam Mazel, Digital Publishing Librarian
Scholarly Communication Department, IUB Libraries

2023-11-08

Why “Re-” in VP?

- NAVSA Conference 2023: “Revision,
RReform”
 - Panel: “Re: Re: Victorian Poetry”
 - Adela Pinch, Professor of English,
University of Michigan
 - Emily Harrington, Assoc. Prof. of
English, UC Boulder
 - Naomi Levine, Asst. Prof. of English,
Yale University
 - Me!

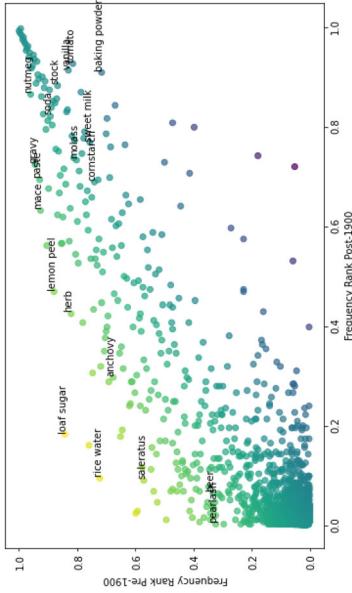


Methodology

- “Re-” 🍞 Text Mining
- Python



- Keyword Frequency
- Key Word in Context (KWIC)
- Term Frequency over Time
- Sentiment Analysis
- Exploratory Data Analysis



Python logo used with permission

Data: Dante Gabriel Rossetti

- *Poems* (1870)
- *Poems: A New Edition* (1881)
- *Ballads and Sonnets* (1881)



Self Portrait, 1861

Image from the [Rossetti Archive](#) in public domain in US

Data: Algernon Charles Swinburne

- *Atalanta in Calydon* (1865)
- *Poems and Ballads* (1866)
- *Songs Before Sunrise* (1871)
- *Songs of Two Nations* (1875)
- *Erechtheus* (1876)
- *Poems and Ballads, Second Series* (1878)
- *Songs of the Springtides* (1880)
- *Studies in Song* (1880)
- *The Heptalogia, or the Seven against Sense. A Cap with Seven Bells* (1880)
- *Tristram of Lyonesse* (1882)
- *A Century of Roundels* (1883)
- *A Midsummer Holiday and Other Poems* (1884)
- *Poems and Ballads, Third Series* (1889)
- *Astrophel and Other Poems* (1894)
- *The Tale of Balen* (1896)
- *A Channel Passage and Other Poems* (1904)

Data: Michael Field

- *Long Ago* (1889)
- *Sight and Song* (1892)
- *Underneath the Bough* (1893)
- *Wild Honey from Various Thyme* (1908)
- *Poems of Adoration* (1912)
- *Mystic Trees* (1913)
- *Whym Chow: Flame of Love* (1914)



Katherine Bradley & Edith Cooper, aka Michael
Field

Image from [Wikimedia Commons](#) in public domain in US

Data: Thomas Hardy

- *Wessex Poems and Other Verses* (1898)
- *Poems of the Past and the Present* (1901)
- *Time's Laughingstocks and Other Verses* (1909)
- *Satires of Circumstance* (1914)
- *Moments of Vision* (1917)
- *Late Lyrics and Earlier with Many Other Verses* (1922)
- *Human Shows, Far Fantasies, Songs and Trifles* (1925)



Image from [Wikimedia Commons](#) in public domain in US

Who Uses “Re-” Words More / Less?

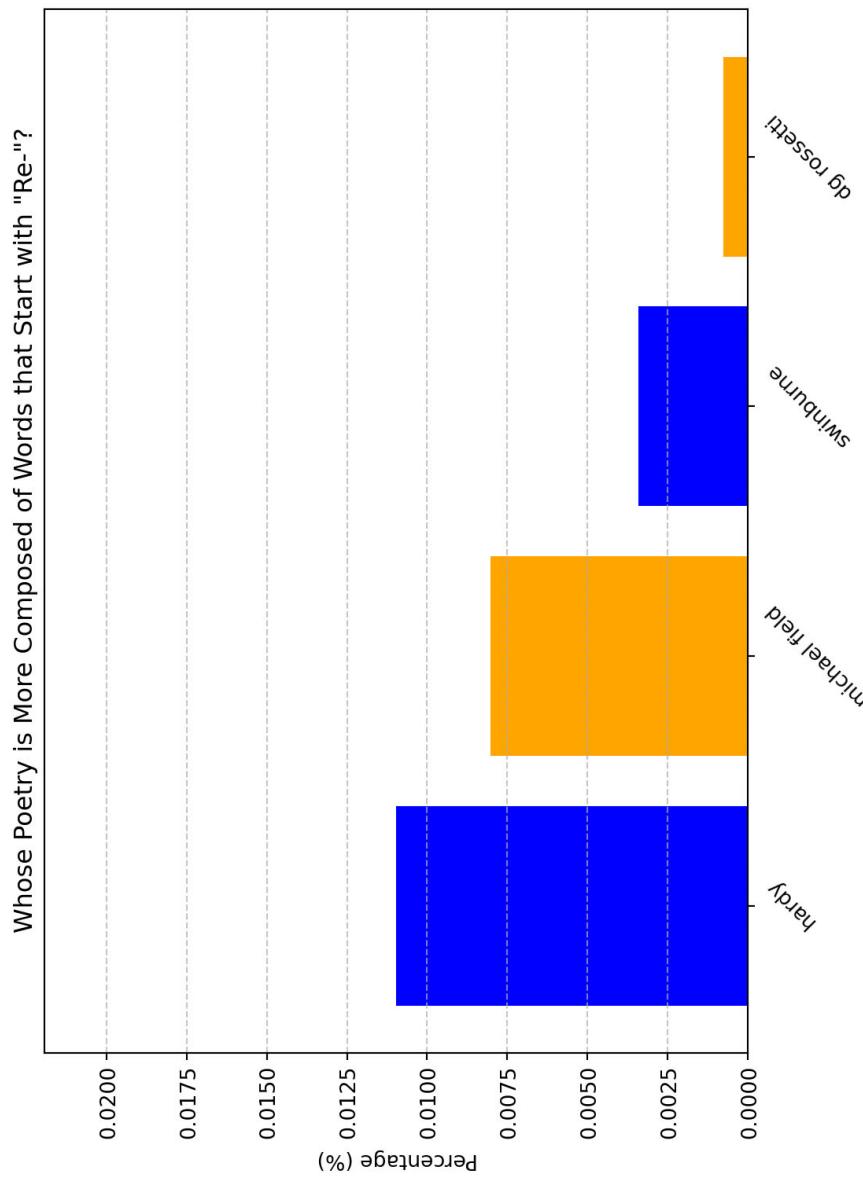
- Keyword Frequency Analysis
 - What percent of each poet’s corpus is composed of words with the prefix “re-”?
 - Compare “Re-” Word Percentages

▼ Code

```
1 # import software libraries
2 import nltk
3 import os
4 import matplotlib.pyplot as plt
5
6 # Download NLTK tokenizer data
7 nltk.download('punkt')
8
9 # Define a function that tokenizes a text and then counts how many of its w
10 def count_re_words(text):
11     words = nltk.word_tokenize(text)
12     return sum(1 for word in words if word.lower().startswith("re-"))
```

```
13 # Specify the directory paths for the two poets' corpora
14 corpus_directories = {
15     'swinburne': '/home/adammazel/Documents/Digital_Scholarship/re-victoria
16     'hardy': '/home/adammazel/Documents/Digital_Scholarship/re-victorian-po
17     'michael_field': '/home/adammazel/Documents/Digital_Scholarship/re-vict
18
```

Who Uses “Re-” Words More / Less?



Which “Re-” Words Are Most Frequent?

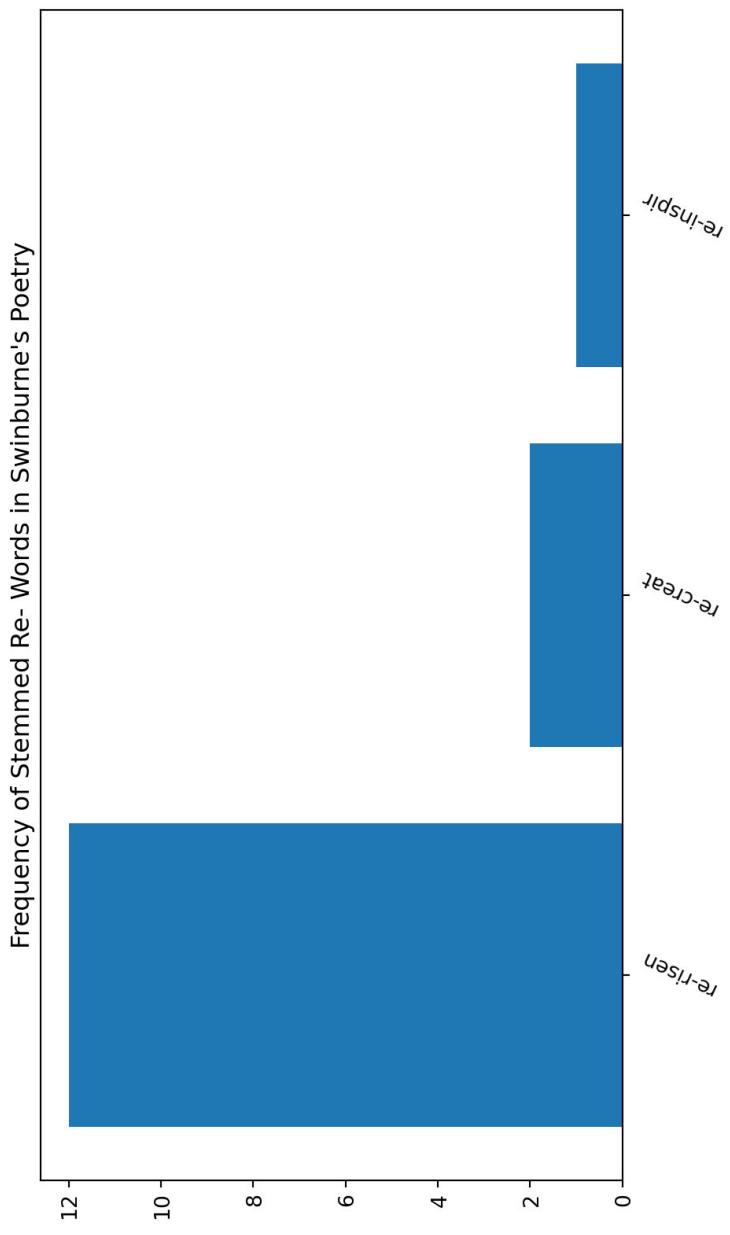
- Keyword Frequency Analysis
 - Which “re-” words are used and how often?

▼ Code

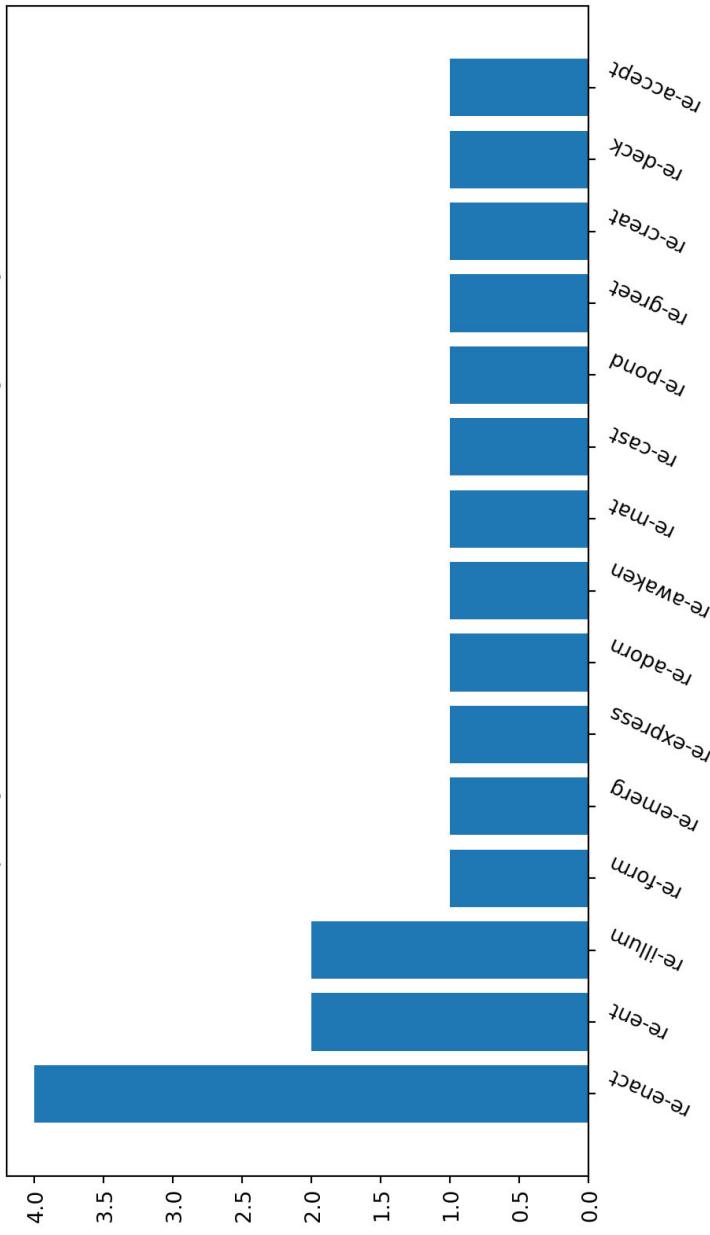
```
1 # import software libraries / dependencies
2 import nltk
3 import os
4 import re
5 import matplotlib.pyplot as plt
6 from collections import Counter
7 from nltk.stem import SnowballStemmer
8
9 # Download NLTK tokenizer data
10 nltk.download('punkt')
11
12 # Initialize Stemmer for English
13 stemmer = SnowballStemmer("english")
14
```

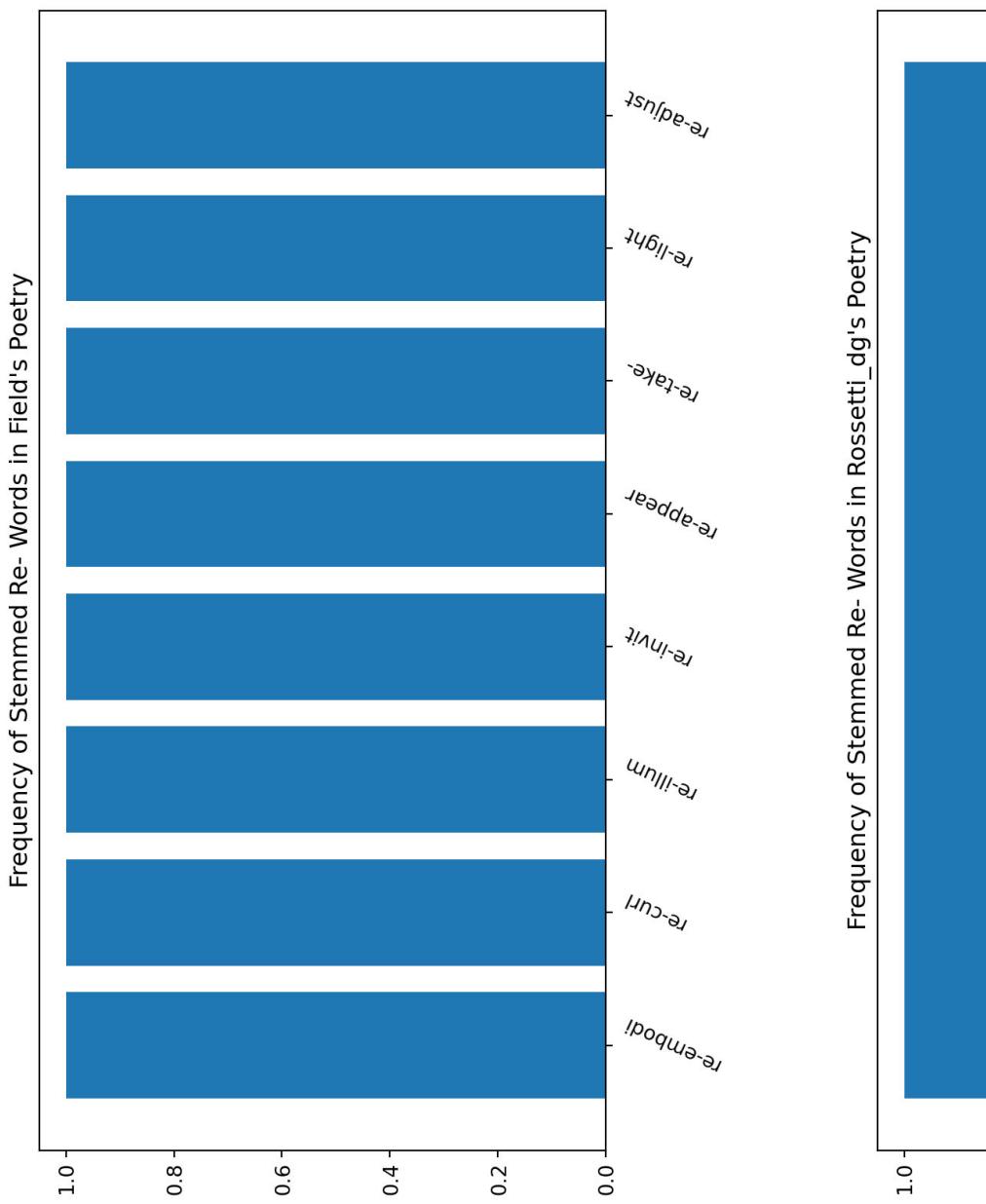
```
15 # create function to preprocess and process files of each directory: create
16 def process_directory(corpus_directory):
17     corpus = []
18
```

Which “Re-” Words Are Most Frequent?



Frequency of Stemmed Re- Words in Hardy's Poetry





Field's and Hardy's "Re-Illume" in Context

- Re-illumine
 - Extremely rare
 - Chieftly poetic
 - 1758 – present
- Field: 1x
 - “[XXII: Sleeping together: Sleep](#)”: Whym Chow: *Flame of Love* (1906, 1914)
- Hardy: 2x
 - “Two Rosalinds”: *Time’s Laughingstocks and Other Verses* (1909)
 - “For Life I had never cared greatly”: *Moments of Vision and Miscellaneous Verses* (1917)

Google Books Ngram Viewer

reillumed+reillume+reilluming+reillumes X ②

1779 - 2019 ▶ British English (2019) ▶ Case-Insensitive Smoothing ▶



(click on line/label for focus)

Search in Google Books

reillumes > 1779 - 1800 1801 - 1839 1840 1841 - 1949 1950 - 2019 English (2019)

reillumming > 1779 - 1820 1821 - 1835 1836 - 1839 1840 - 1958 1959 - 2019 English (2019)

Do “Re-” Words and “Re” Words Co-occur in Hardy’s Poetry?

- Key (Re-) Word in Context (KWIC)
- Have Python return Hardy’s sentences that contain word(s) that start with “re-” and word(s) that start with “re”

▼ Code

```
1 # import libraries
2 import os # open directories on local machine
3 import nltk # nlp
4 import re # reg ex
5 from nltk.tokenize import sent_tokenize, word_tokenize # break text blob in
6
7 # Define regular expressions
8 re_pattern = r'\b(?:[Rr]e|[Rr]E)\w+\b' # Matches "rew" or "Re" or "rE"
9 re_hyphen_pattern = r'\b(?:[Rr]e-|[Rr]E-)\w+\b' # Matches "re-" or "Re-" or
10 # \b matches a word boundary.
11 #(?:[Rr]e|[Rr]E) is a non-capturing group that matches either "re" or "Re"
12 # \w+ matches one or more word characters following "re" or "Re" or "rE" or
```

```
13 def matches_pattern(word, pattern):
14     return re.search(pattern, word, re.IGNORECASE) is not None
15
16 # Define function to surround matched words with asterisks
17
18 def bold_matched_words(match, pattern):
```

Do “Re-” Words and “Re” Words Co-occur in Hardy’s Poetry?

Within there
Too mocking to Love’s *re-expression*
Was Time’s *repartee*!

IX

“The words, sir?” cried a creature
Hovering mid the shine and shade as ‘twixt the live world and the
tomb;
But the well-known numbers needed not for me a text or teacher
To *revive* and *re-illumine*.

When Are “Re-” Words More / Less Frequent?

- Time Series / Term Frequency over Time
 - Y axis: re- word percentage of book
 - X axis: publication year of book

▼ Code

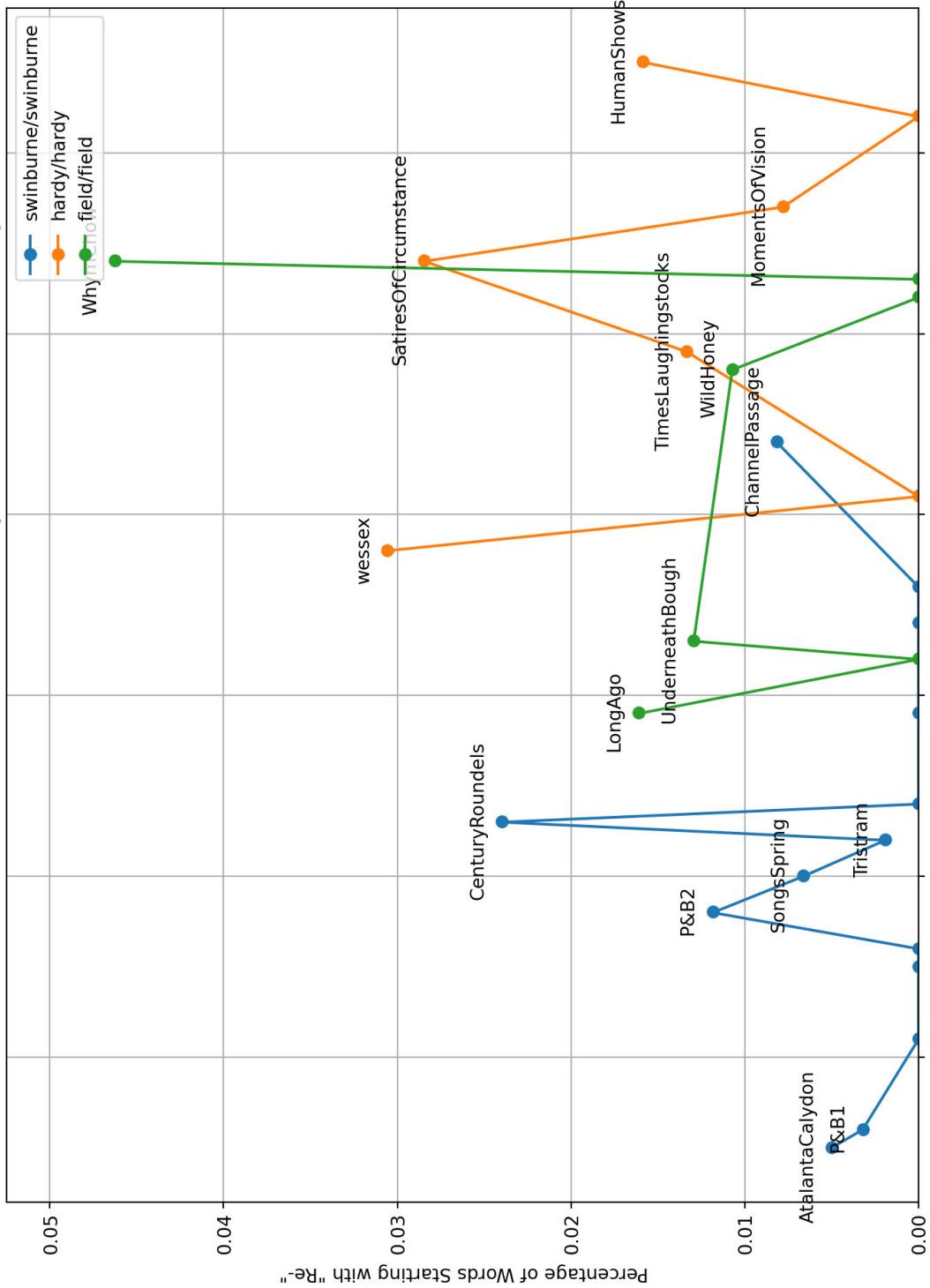
```
1 # Import software libraries
2 import nltk # nlp
3 import os # interact with directories on local machine
4 import re # reg ex
5 import matplotlib.pyplot as plt # data visualization
6 import string # punctuation removal
7
8 # Download NLTK tokenizer data
9 nltk.download('punkt')
10
11 # Define a function that takes each text, tokenizes it into individual word
```

```
12 def count_re_words(text):
13     words = nltk.word_tokenize(text)
14     return sum(1 for word in words if word.lower().startswith("re-"))
15
16 # Define a function to remove all punctuation except hyphens
17 def remove_punctuation_except_hyphens(text):
18     translator = str.maketrans(''',''.'', string.punctuation.replace('-', ''))
```



When Are “Re-” Words More / Less Frequent?

When Are "Re-" Words Used in Field's, Hardy's, and Swinburne's Poetry?



Are “Re-” Words Used Positively or Negatively?

- Sentiment Analysis
 - determines a text’s emotional tone (positive / negative / neutral)

▼ Code

```
1 # import software libraries
2 import nltk #nlp
3 from nltk.sentiment.vader import SentimentIntensityAnalyzer # VADER sentime
4 import os # enable engagement with directories and files on local machine
5
6 # Initialize the VADER sentiment analyzer
7 sia = SentimentIntensityAnalyzer()
8
9 # Function to calculate aggregate sentiment (positive / neutral / negative)
10 # start counts at zero
11 def calculate_aggregate_sentiment(sentences):
12     positive_score = 0
```

```
13 negative_score = 0
14 neutral_score = 0
15 total_sentences = 0
16
17 # for each sentence, calculate sentiment polarity score: if score is positive
18 for sentence in sentences:
```

Are “Re-” Words Used Positively or Negatively?

AC Swinburne
Total Sentences Analyzed: 15
Average Positive Score: 0.1239333333333333
Average Negative Score: 0.152
Average Neutral Score: 0.7242666666666665
Overall Sentiment: Negative

Thomas Hardy
Total Sentences Analyzed: 21
Average Positive Score: 0.08904761904761904
Average Negative Score: 0.11304761904761904
Average Neutral Score: 0.7979047619047619
Overall Sentiment: Negative

Michael Field
Michael Field

Conclusion

- Text mining morphemes can:
 - illuminate significant micro-elements of poetic style
 - show how seemingly trivial features contribute to poetic meaning
- Distant Reading 🎧 Close Reading
- Thank You!
- Contact Info
 - Adam Mazel
 - Digital Publishing Librarian
 - Indiana University Bloomington
 - amazel@iu.edu

