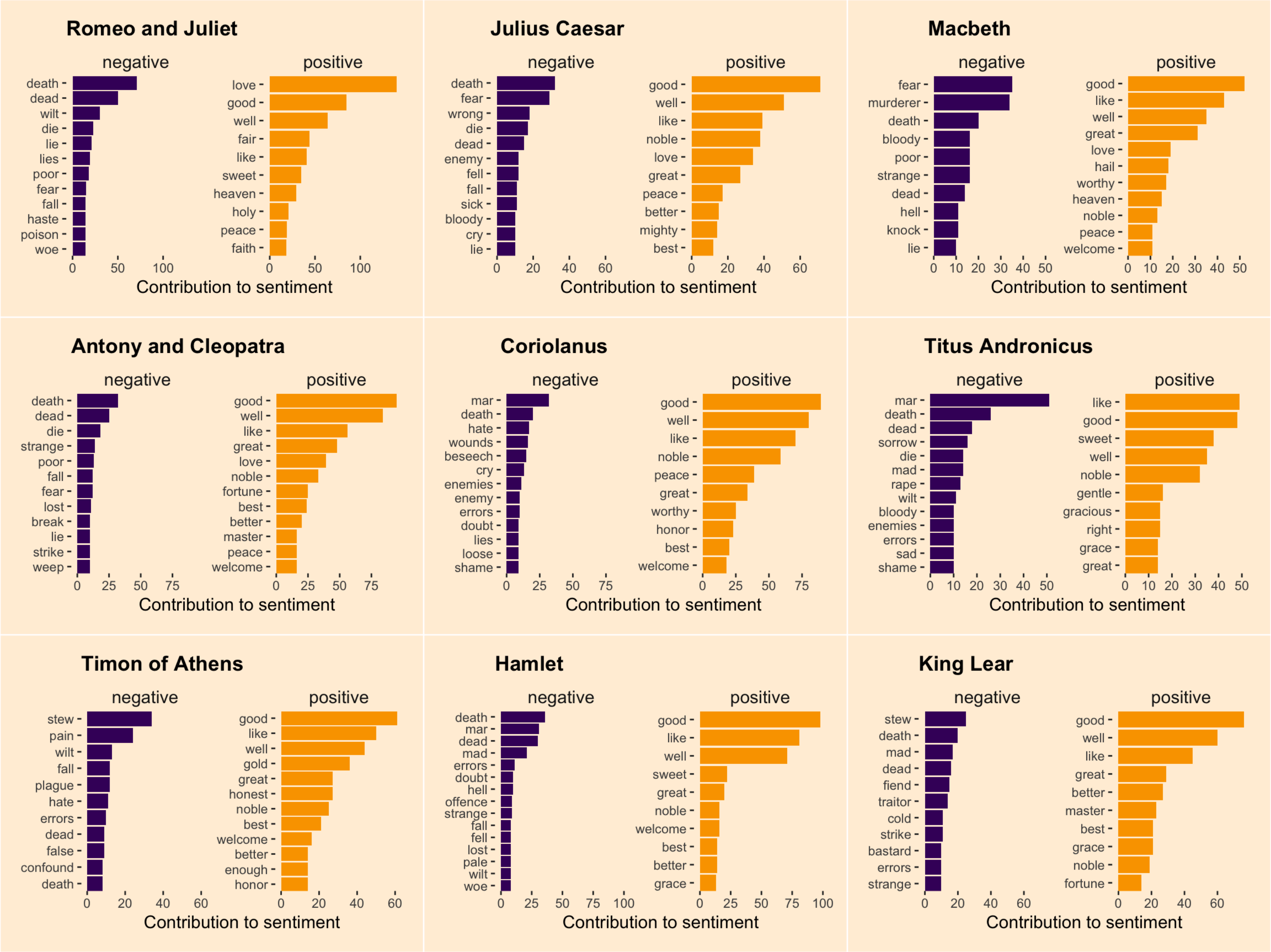
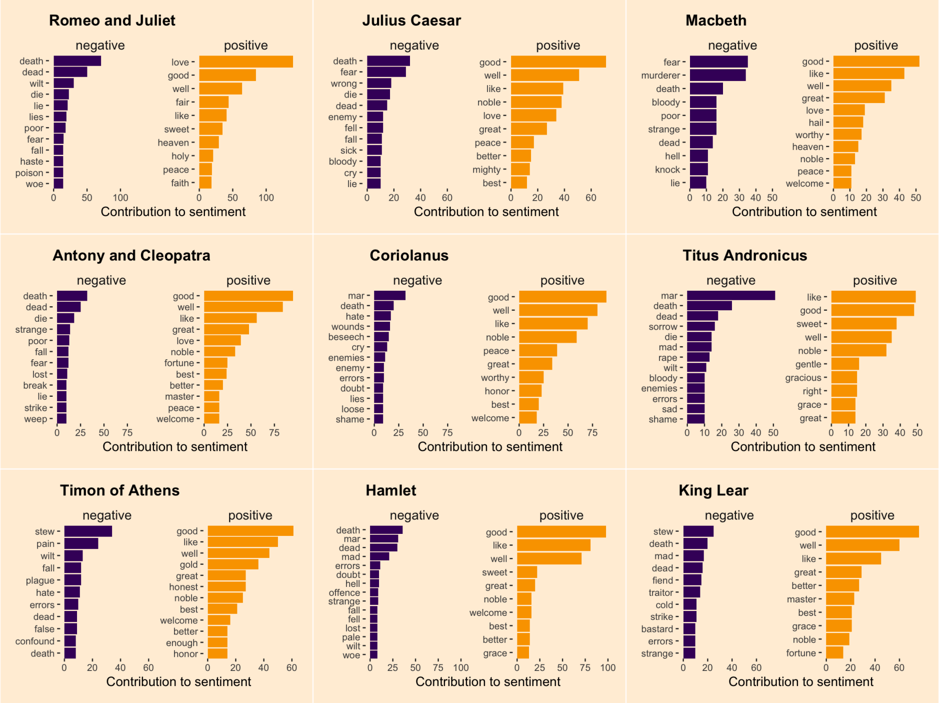
Natural Language Processing (NLP)

Text Analysis



https://peerchristensen.netlify.app/post/fair-is-foul-and-foul-is-fair-a-tidytext-entiment-analysis-of-shakespeare-s-tragedies/

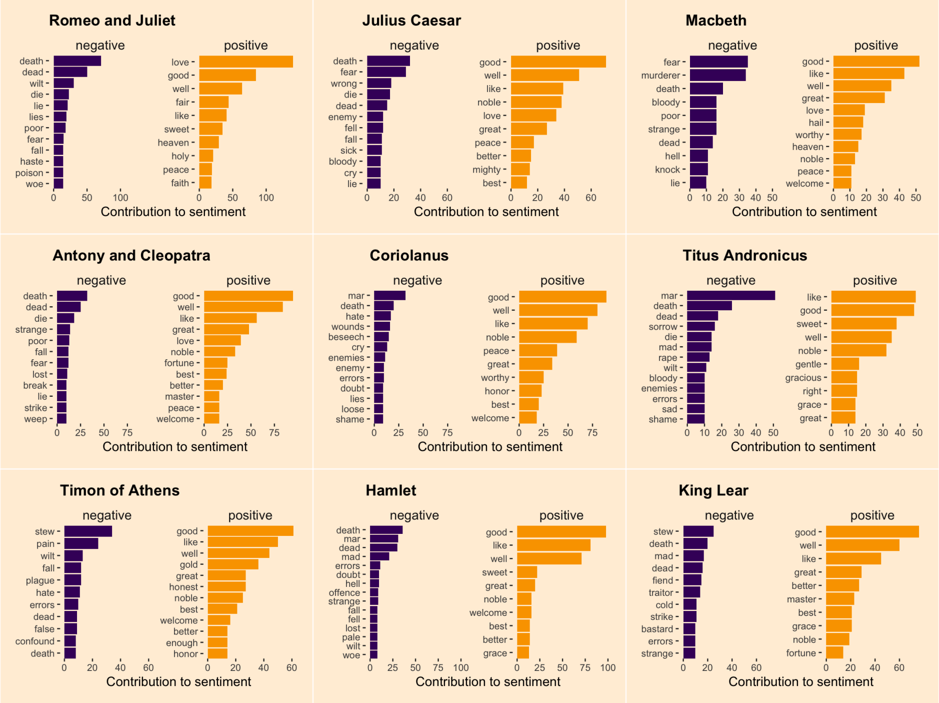
Text Analysis



- What words/topics does Shakespeare commonly use to set a positive/negative tone?
- How many words/topics?
- What variety of language does he use?
- Does he use some words more commonly than other authors?
- What model describes the word frequency distribution in Shakespearian plays?

https://peerchristensen.netlify.app/post/fair-is-foul-and-foul-is-fair-a-tidytext-entiment-analysis-of-shakespeare-s-tragedies/

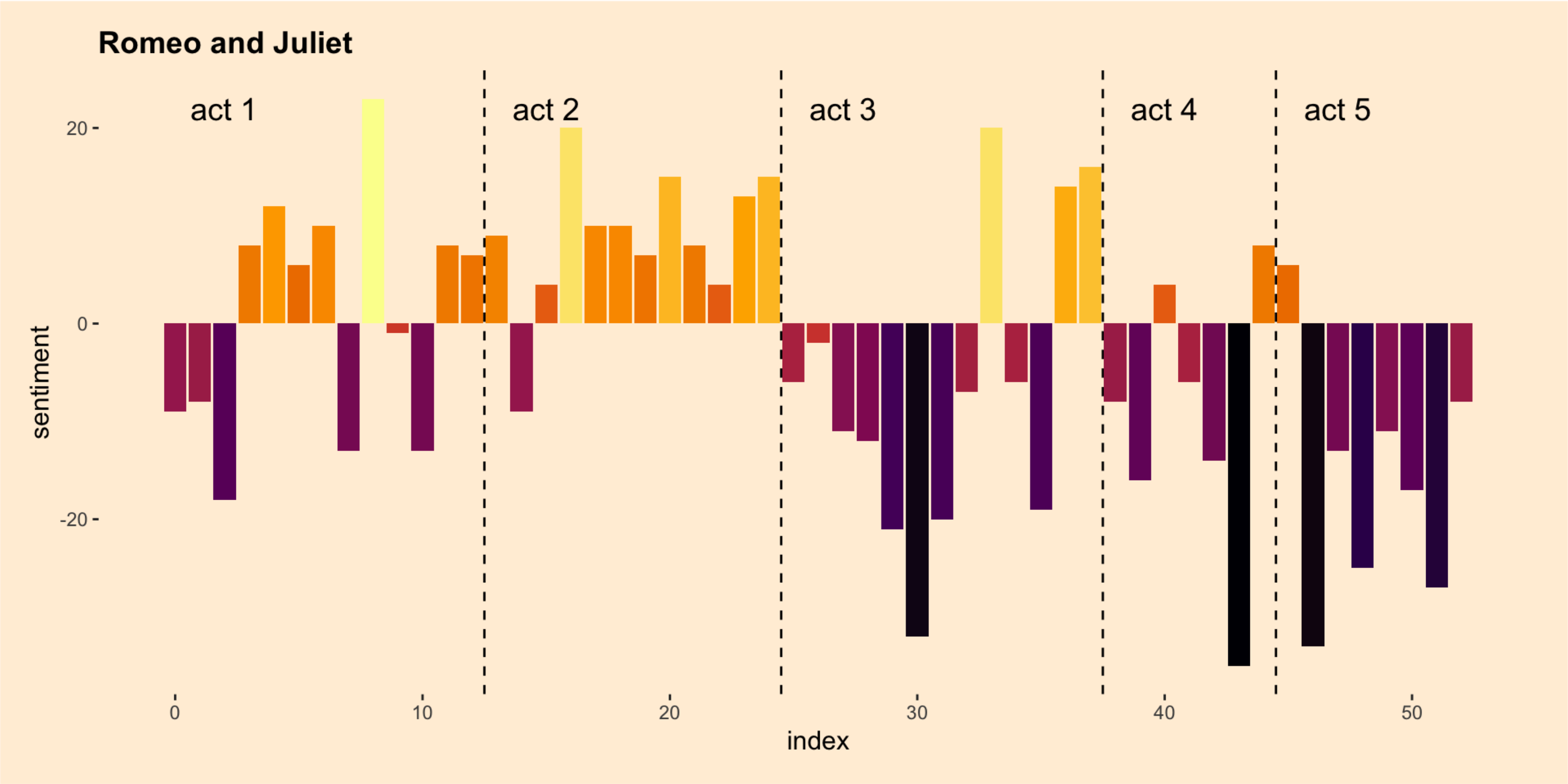
Text Analysis



- What words/topics does Shakespeare commonly use to set a positive/negative tone?
  - What is "common"?
  - Max number of times?
  - Average frequency across all plays?
  - More times than other words?
- How many words/topics?
- What variety of language does he use?
  - How do the word frequencies deviate from each other in the play?
- Does he use some words more commonly than other authors?
  - How do the frequencies deviate from frequencies in other plays?
- What model describes the word frequency distribution in Shakespearian plays?

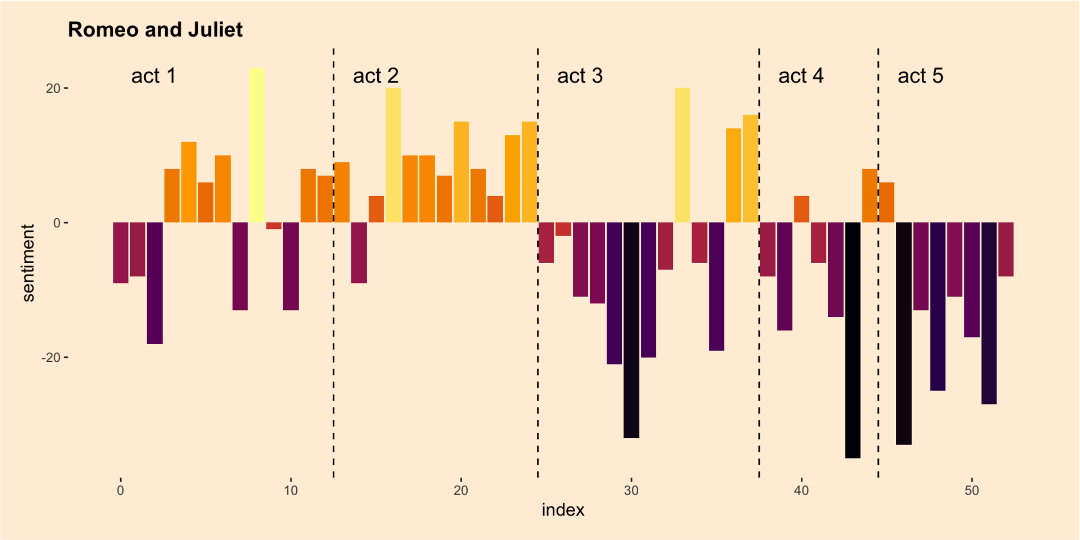
https://peerchristensen.netlify.app/post/fair-is-foul-and-foul-is-fair-a-tidytext-entiment-analysis-of-shakespeare-s-tragedies/

NLP Data Visualization



https://peerchristensen.netlify.app/post/fair-is-foul-and-foul-is-fair-a-tidytext-entiment-analysis-of-shakespeare-s-tragedies/

NLP Data Visualization



- How do we visualize differences between scenes (which have hundreds of words) with one single bar per scene?
- How do we visualize categorical differences?
- Does sentiment evolve similarly in other Shakesperean tragedies? In general tragedies?
- Can we model the evolution of sentiment in a play?
- What model underlies the sentiment evolution of tragedies?

https://peerchristensen.netlify.app/post/fair-is-foul-and-foul-is-fair-a-tidytext-entiment-analysis-of-shakespeare-s-tragedies/

Text Analysis with Code

- Specifically for text we’ll be looking at today:
  - Auto-summarize text
  - Identify frequent words and distinctively frequent words in a text
  - Identify topics in news posts
  - Identify sentiments

This is the realm on Natural Language Processing (NLP)

- Common tasks:
  - Tokenization
  - Removing stopwords
  - Identifying pairs/triples/etc of words
  - Word sense disambiguation
  - Parts of speech tagging
  - Stemming
  - Lemmatization

NLP Tasks: Tokenization

Once upon a time, there was a cat. It was black and fluffy.

NLP Tasks: Tokenization

Once upon a time, there was a cat. | It was black and fluffy.

NLP Tasks: Tokenization

Once | upon | a | time, | there | was | a | cat. | It | was | black | and | fluffy.

NLP Tasks: Remove Stopwords

Once upon a time, there was a cat. It was black and fluffy.

NLP Tasks: Remove Stopwords

Once upon ■ time, there was ■ cat. It was black ■ fluffy.

NLP Tasks: N-Grams

Once upon a time, there was a cat. It was black and fluffy.

(Once, upon)
(upon, a)
(a, time)
(time, there)
(there, was)
(was, a)
….

(Once, upon, a)
(upon, a, time)
(a, time, there)
(time, there, was)
(there, was, a)
(was, a, cat)
….

NLP Tasks: Word Sense Disambiguation

Once upon a time, there was a ***cat.*** It was black and fluffy.

Cat: feline

Cat: clear air turbulence

Cat: computerized axial tomography

Cat: a man (especially among jazz enthusiasts)

NLP Tasks: Part-of-speech tagging

pronoun

adverb prep art noun exis verb art noun verb adj conj adj

Once upon a time, there was a cat. It was black and fluffy.

NLP Tasks: Stemming

Sail

Sailing

Sailed

Sails

NLP Tasks: Stemming

Sail[redacted]

Sail

Sail

Sail

NLP Tasks: Lemmatization

Lemmatization is more careful with words than stemming.

It usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally only removing inflectional endings and to return the base or dictionary form of a word, which is known as the *lemma*.

Should *saw* be considered similar to *“see”*? Or to the noun *“saw”*?

Challenges of NLP

- Contextual words and phrases and homonyms
- Synonyms
- Irony and sarcasm
- Ambiguity
- Errors in text or speech
- Colloquialisms and slang
- Domain-specific language
- Low-resource languages
- Lack of research and development

NLTK Intro Notebook

Using NLP to wrangle text

- We’re breaking text apart and extracting information

- For example, we can get numerical attributes from text
  - Frequency of words that occur in the text

- Use that to our advantage

Using NLP to wrangle text

- Let's say that we have lots of journal articles to read but not enough time, so we only want to read the highlights.
- Auto-generate the bullet points
  - ID the most important sentences

Auto-bulleter

1. Find most important words

2. Assign score to sentences based on their words

3. Output the top-scoring sentences

Auto-bulleter

1. Find most important words
    - Authors tend to repeat important words -> use word frequency

2. Assign score to sentences based on their words

3. Output the top-scoring sentences

Auto-bulleter

1. Find most important words
   - Authors tend to repeat important words -> use word frequency

2. Assign score to sentences based on their words
   - Take the words it contains and sum their "importances"

3. Output the top-scoring sentences

Auto-bulleter

1. Find most important words
   - Authors tend to repeat important words -> use word frequency

2. Assign score to sentences based on their words
   - Take the words it contains and sum their "importances"

3. Output the top-scoring sentences
   - Rank the sentences

Bullet Points Notebook

Distinguishing word usage

- Term frequencies are useful, but sometimes we want to distinguish word usage between different texts

- Example:
  - Moby Dick might use ”starboard” more than Charlotte’s Web
  - Charlotte’s Web might use “pigpen” more than Moby Dick
  - They may both use “sky” or “cold” more frequently than either “starboard” or “pigpen”

Distinguishing word usage

- Term frequencies are useful, but sometimes we want to distinguish word usage between different texts

- For that, TF-IDF is useful
  - Term Frequency – Inverse Document Frequency
  - Takes the term frequencies and gives a low weight to the ones that are common across all documents, but a high weight to ones that are more distinctive
    - TF-IDF of word = (TF of word) * log [(total document number) / (documents with word)]

    - (there are other math options behind this which we'll ignore)

US Inaugural Speeches Notebook

Text analysis

- Workflow
  - Pick text or group of texts
  - Pick problem type
    - Identifying import words/sentences
    - Using words as a distinguishing feature in text comparisons
  - Represent data using numerical attributes
  - Apply algorithm

Text analysis

- Workflow
  - Pick text or group of texts
  - Pick problem type
    - Identifying import words/sentences
    - Using words as a distinguishing feature in text comparisons
    - **Regression**
    - **Classification**
    - **Clustering**
    - **Recommendations**
  - Represent data using numerical attributes
  - Apply algorithm

Classification

- Identify distinct categories or topics
  - Cat or dog
  - Sunny or cloudy
  - Tumor or not tumor
  - Spam or not spam
  - Positive tweet or negative tweet
  - Happy lyrics or sad lyrics
- Problem instance
  - An email, a tweet, a song lyric, a picture of an animal
- Assign this instance to a category
  - Spam or not spam, positive tweet or negative tweet, cat or dog

Clustering

- What if groups are **unknown** beforehand?
- Clustering
  - Divide articles into different groups based on content
- Might later look at the divisions and determine meaning
  - For example, themes, topics, character associations, …
- Useful to determine patterns you may not realize exist

News Topics Notebook

Sentiment Analysis and Classifiers

- Language is incredibly complex and nuanced
- Trying to ascertain the emotional sentiment of a piece of text is still a developing area of study

Sentiment Analysis and Classifiers

- Language is incredibly complex and nuanced
- Trying to ascertain the emotional sentiment of a piece of text is still a developing area of study

Sentiment Analysis and Classifiers

- Language is incredibly complex and nuanced
- Trying to ascertain the emotional sentiment of a piece of text is still a developing area of study

- For the moment, a lot of sentiment analysis simply looks at whether a text has a positive or negative connotation
  - Polarity is positive or negative
  - Emoji is happy or not

Sentiment Analysis and Classifiers

- Language is incredibly complex and nuanced
- Trying to ascertain the emotional sentiment of a piece of text is still a developing area of study

- For the moment, a lot of sentiment analysis simply looks at whether a text has a positive or negative connotation
  - Polarity is positive or negative
  - Emoji is happy or not
- Binary Classification problem

Methods to classification

- Is an email spam or not?
- Rule based method
  - Manually write a list of rules that labels email as spam or not based on the words in the email (identifying keywords)
  - May be difficult, and may not be easy to update
- Machine learning method
  - Use the computer to tailor rules based on historical data
  - If you have a large amount of examples (like 100s of emails in your folder that you've already manually classified as spam and 1000s that you said were ok)
  - And/or if the patterns are dynamic (spam senders getting smarter)

Sentiment Notebook