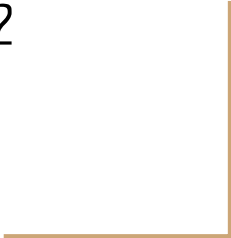


# Programming, Problem Solving, and Algorithms

CPSC203, 2023 W2



# Announcements

- Project 3 Autograder coming soon...
  - Apologies for the delay, we had some difficulties debugging the autograder for this project
- I just discovered that the last day of the term, so everyone will get 1 extra day on all due dates for this course!
- Final Exam Practice questions coming by Thursday (Friday at the latest)
  - Most of them will not be new questions, will give you additional opportunities for practice from existing Examlets

# Today's Plan...

1. Announcements! (10 mins)
2. Retrospective Overview of CPSC 203
3. Introduction to Visualizing Literature (30 mins)
  - Content is not examinable on the final exam
  - Another application of "Graphs"



# Retrospective Overview of CPSC 203



# Today's Plan...

## Part 1 - Introductions

Week 1 - Introductions! ✓

Week 2 - Python Review ✓

## Part 2 - Dataclass in Python

Week 3 - Efficiency and Dataclass ✓

Week 4 - Dataclass cont'd ✓

## Part 3 - Working with Data

Week 5 - Web Scraping ✓

Week 6 - Git and Version Control ✓

Week 7 - Reading week!

## Part 4 - Algorithms and Data Structures

Week 8 - Data Structures ✓

Week 9 - Graphs ✓

Week 11 - MHall ✓

Week 12 - State Spaces ✓

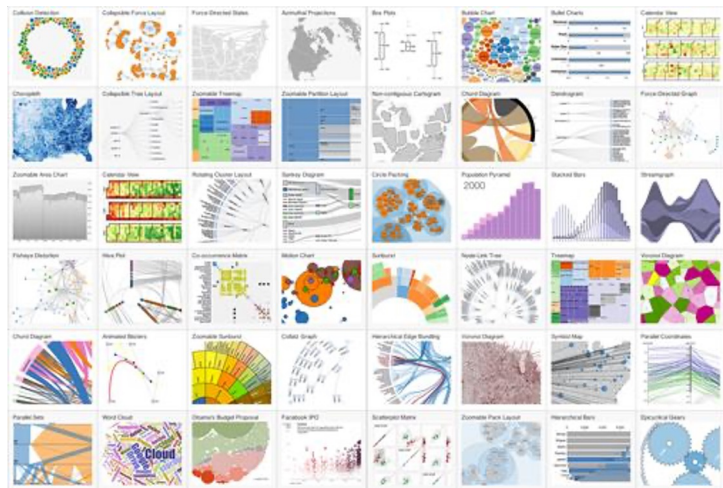
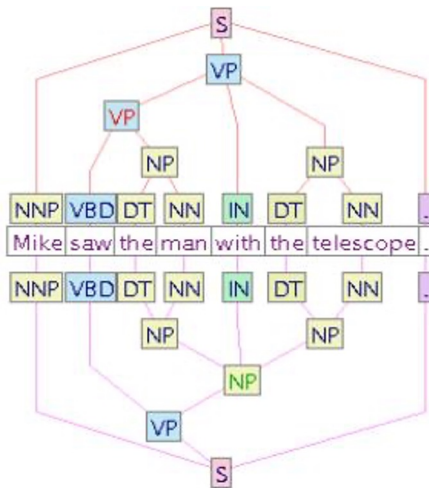
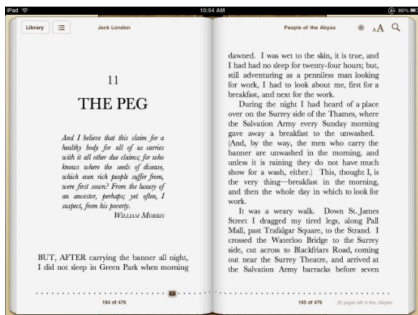
Week 13 - Maps ✓



# Slides from the Assigned Videos

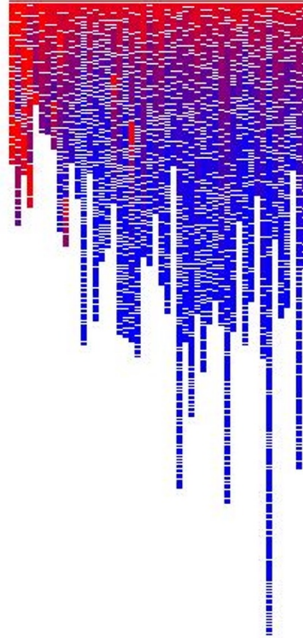


# Visualizing Literature

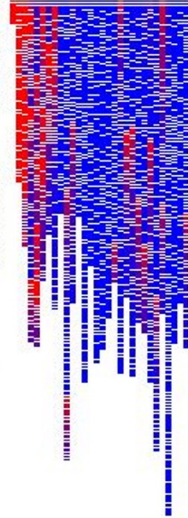


# Example

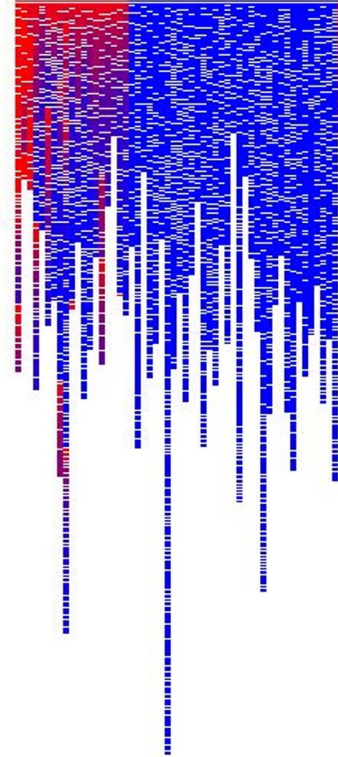
Sense and Sensibility



Kidnapped



Emma



[http://datamining.typepad.com/data\\_mining/2011/09/visualizing-lexical-novelty-in-literature.html](http://datamining.typepad.com/data_mining/2011/09/visualizing-lexical-novelty-in-literature.html)



# Example

## NOVEL VIEWS - Les Misérables - Word Connections

### Radial Word Connections

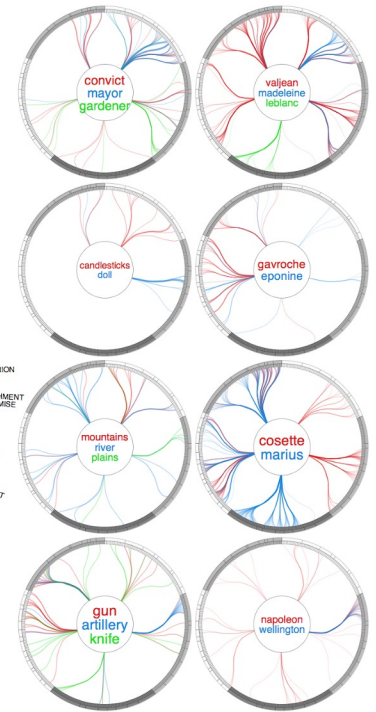
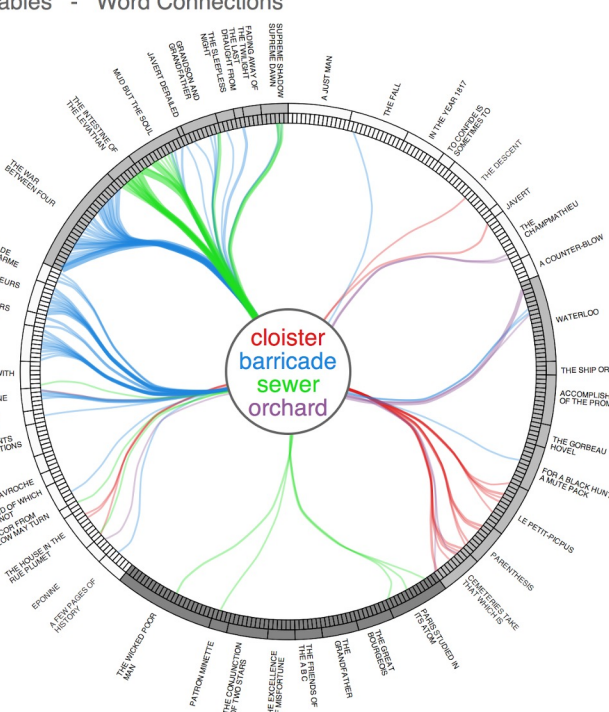
A word used in multiple places in a text can be interpreted as a connection between those locations. Depending on the word itself the connection could be in terms of character, setting, activity, mood, or other aspects of the text. This graphic shows, for the novel Les Misérables, a number of these word connections.

The 365 chapters of the text are shown with small segments on the inner ring of the circle with the first chapter appearing at the top and proceeding clockwise from there. The outer ring shows how the chapters are grouped into books of the novel and the book titles are shown as well. The words in the middle are connected using lines of the same color to the chapters where they are used.

This small example below shows that the author devoted a book to the battle of Waterloo and that there were a few scattered references elsewhere. Similarly, we can see with the blue that there is another book entirely about slang.

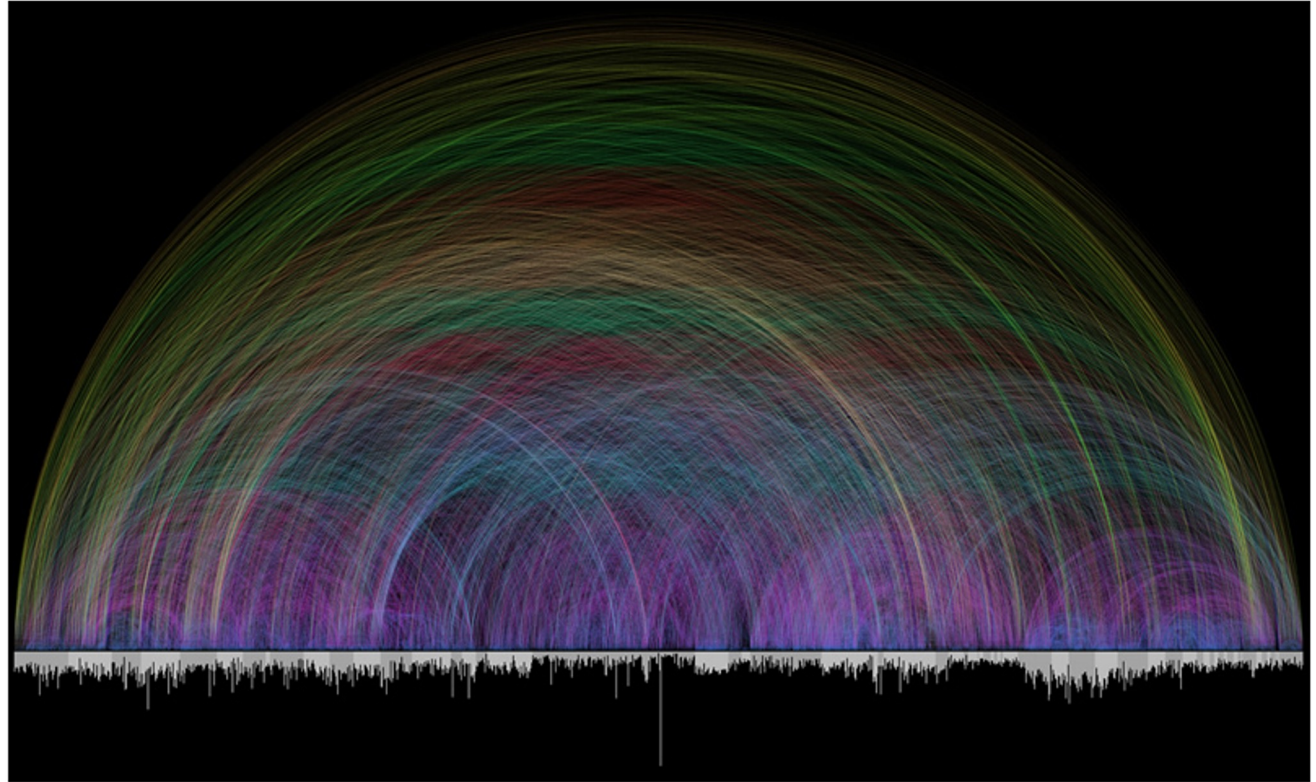


Jeff Clark - neoformix.com - © 2013



<http://neoformix.com/2013/NovelViews.html>

# Example



<http://www.chrisharrison.net/index.php/Visualizations/BibleViz>

# Example

## SENTIMENT ANALYSIS

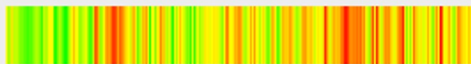
VIEW

Bars

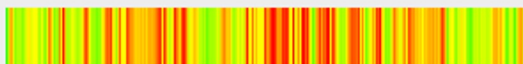
Graphs

These graphs show an analysis of the feeling for each page throughout Tolkien's works. The sentiment has been analysed for each sentence and then average over each page. Green, yellow and red indicate positive, neutral and negative sentiments respectively.

THE SILMARILLION



THE HOBBIT



THE FELLOWSHIP OF THE RING



THE TWO TOWERS



THE RETURN OF THE KING



<http://lotrproject.com/statistics/books/>

# Example

Topics in PMLA Overview Topic- Document Word Bibliography All words About

Grid Scaled List Click for more on topic 23 shakespeare play plays Drag to pan; scroll wheel or double-click to zoom Reset zoom

war man noch nur hat seiner	stern love fielding swift thackeray nike	academic research students university mia graduate	use makes seems point view comic	royal several new london sir company	divine church christian god religious christ	economic public political social society class	philosophy thought man nature science natural	sense use words word meaning language	theater dramatic plays stage drama
passion loves pastoral love lover lady	america white new american black african	montaigne classical greek renaissance epic roman	defoe yeats sonnet irish sonnets ireland	bat man pat pe hym ms	voyage cooper twain mark captain city	english old name beowulf king story	new woman sexual women female gender	man world life man human moral	
new life world first two time	english two line lines verse first	first ff king story two romance	first two play shakespeare plays scene	order carlyle number group whitman table	action play hamlet tragedy death tragic	artistic beauty esthetic art artist painting	swat mama cette plus aux dort	elizabethan hath english sir good man	
british piu italy italian translation petrarch	freud dreamer pearl dream hardy dreams	literary essay critical criticism poetry critics	lines poetry poet poem poems poetic	association languages english language modern study	popular songs song music ballad ballads	lines ms first text two line	first written letters letter two wrote	pues quin mas spanish spain lope	
novels fiction story novel narrative reader	hoc fol ms vnd latin daz	man trouss tale chaucer takes prolog	state law political king war england	dom plus voltaire rousseau diderot moliere	studies new literature literary history historical	made god time make never great	cultural prms world new human film	first two seems time evidence fact	
imagination nature coleridge wordsworth keats mind	two paris france french first century	participle latin use english examples verb	little less character great work form	book work two first found part	man world russian art reality nietzsche	marriage wife father man young old	old form word two words forms used	friedrich lessing goethe german faust germany	

<http://agoldst.github.io/dfr-browser/>

# Example



<http://www.emelyn baker.com/westeros.html>

# How do we begin?

```
textRaw = open('res/ofk.txt').read()
```

returns a string.

We want to analyze the data by word or by \_\_\_\_\_ or by \_\_\_\_\_ or by \_\_\_\_\_...

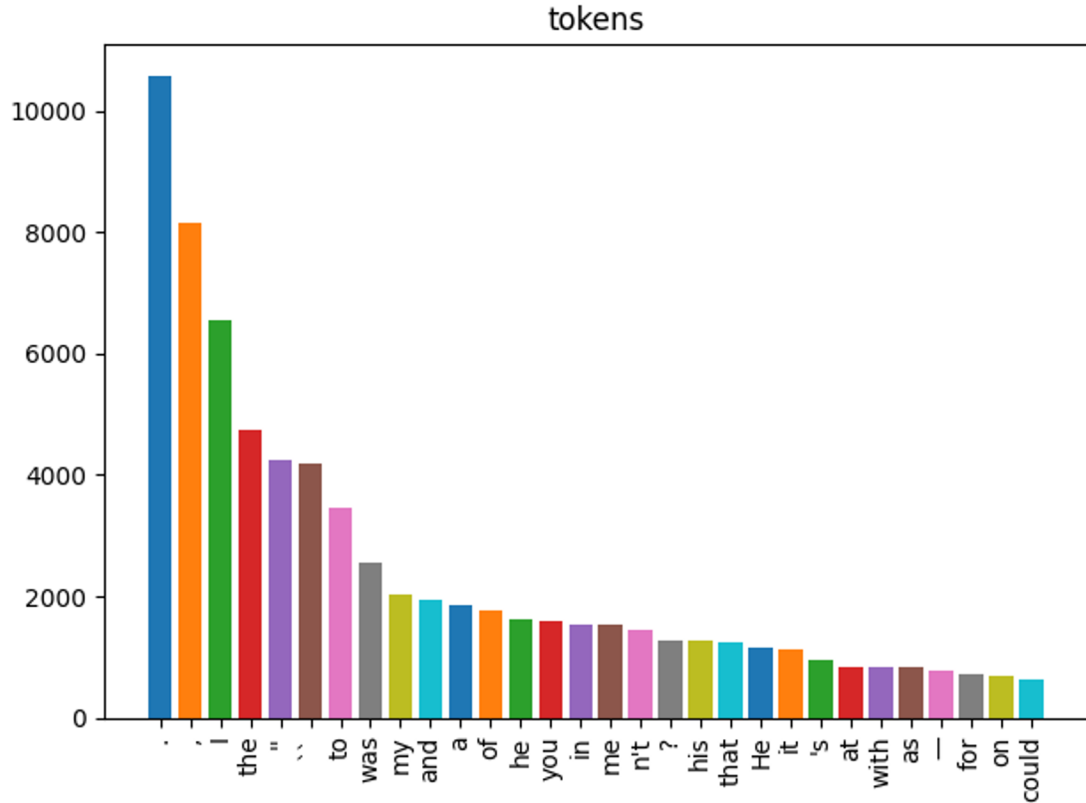
can separate the data into any of these using nltk's "tokenizer"

# Tokenization

Translate: "Astrology. The governess was always getting muddled with her astrolabe, and when she got specially muddled she would take it out of the Wart by rapping his knuckles. She did not rap Kay's knuckles, because when Kay grew older"

Into: ['Astrology.', 'The', 'governess', 'was', 'always', 'getting', 'muddled', 'with', 'her', 'astrolabe', ',', 'and', 'when', 'she', 'got', 'specially', 'muddled', 'she', 'would', 'take', 'it', 'out', 'of', 'the', 'Wart', 'by', 'rapping', 'his', 'knuckles.', 'She', 'did', 'not', 'rap', 'Kay', "'s", 'knuckles', ',', 'because', 'when', 'Kay', 'grew', 'older']

# Python Demo



The python script in “LecOFK” was assembled from examples in Ch1-3 of the NLTK book.

<http://www.nltk.org/book/>

This chart shows the 30 most frequent tokens in the mystery book.



# Pre-processing

49 begged so hard, cried even, I had to let him stay. It  
50 turned out okay. My mother got rid of the vermin and  
51 he's a born mouser. Even catches the occasional rat.  
52 Sometimes, when I clean a kill, I feed Buttercup the  
53 entrails. He has stopped hissing at me.

54

55 Entrails. No hissing. This is the closest we will ever  
56 come to love.

57

58

59

60 3 | Page

61

62

63

64 The Hunger Games – Suzanne Collins

65

66

67

68 I swing my legs off the bed and slide into my hunting  
69 boots. Supple leather that has molded to my feet. I

# A feasible sequence...

lower case

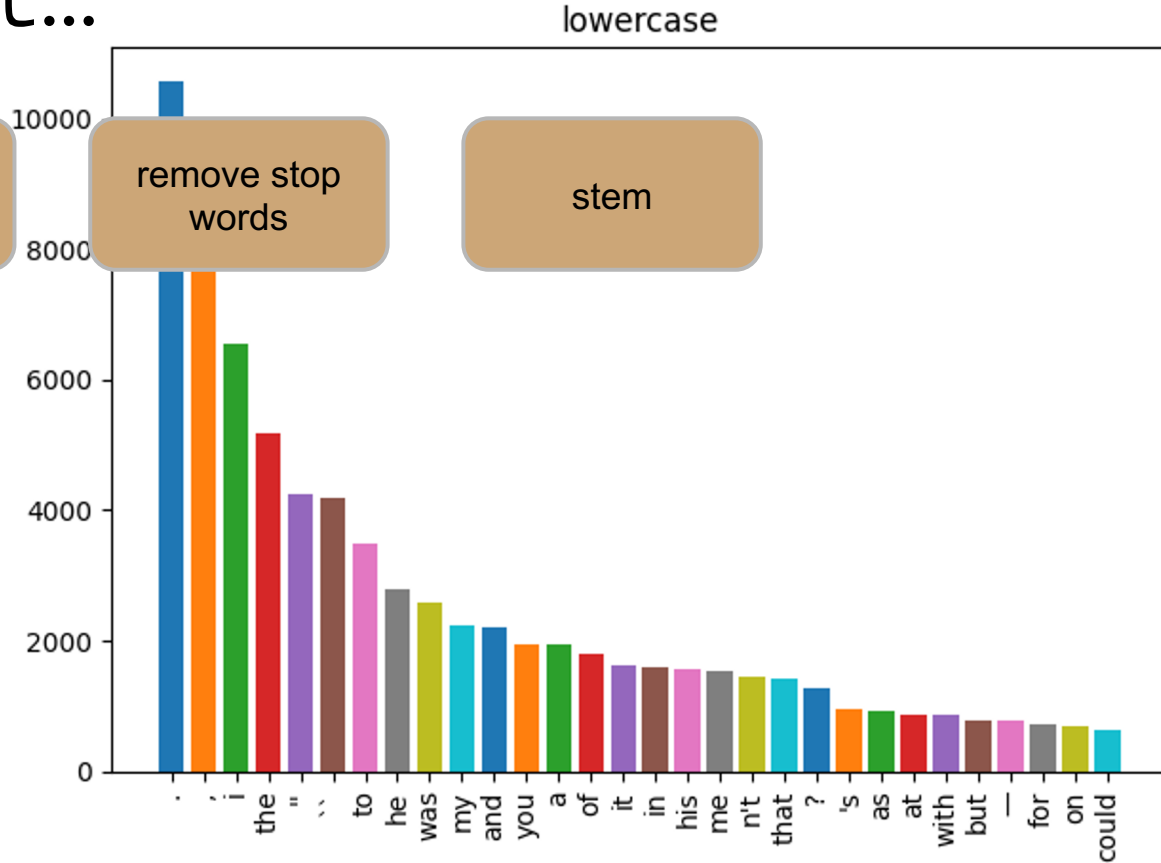
eliminate  
punctuation

remove stop  
words

stem

Unify tally for “Valor” and  
“valor”

Depending on task, may not  
want to do this... caps are  
useful for detecting “named  
entities.”



# A feasible sequence...

lower case

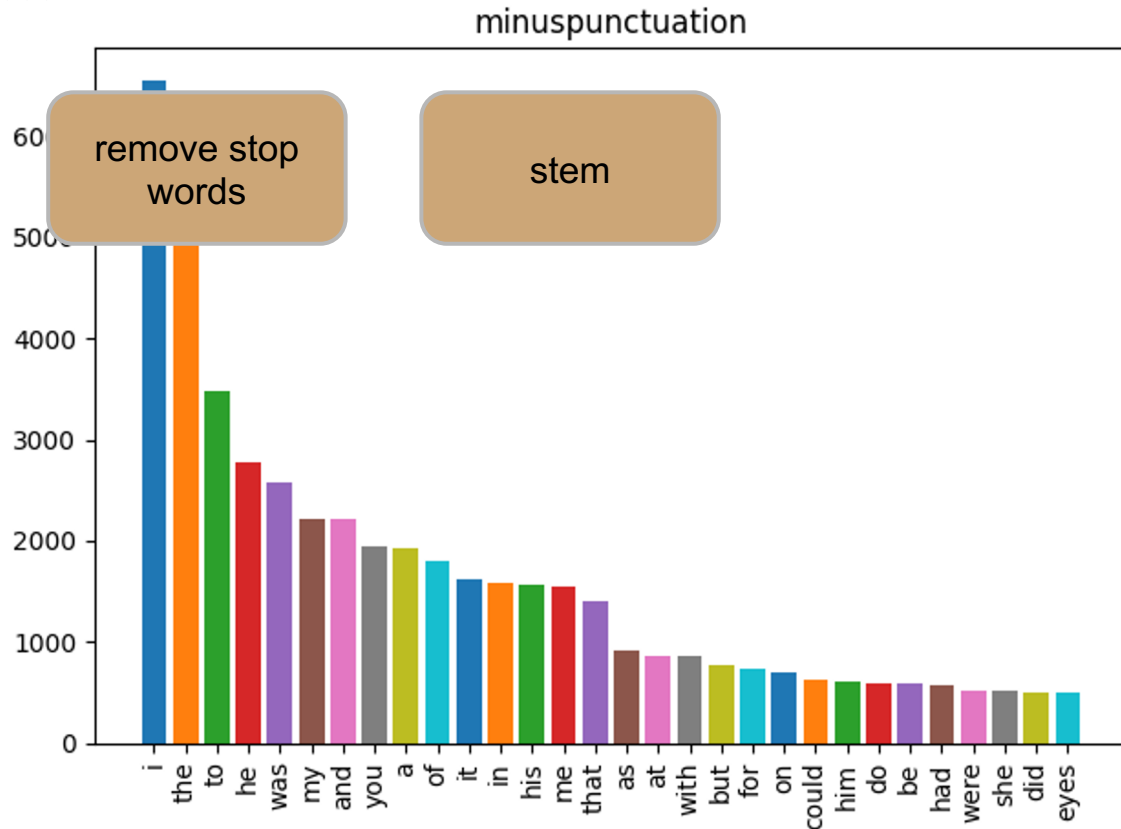
eliminate  
punctuation

remove stop  
words

stem

Punct tokenizer leaves periods  
at end of sentences: “father.”

amazingly, it works fine for  
“Dr.”, “\$3.50”, “!”



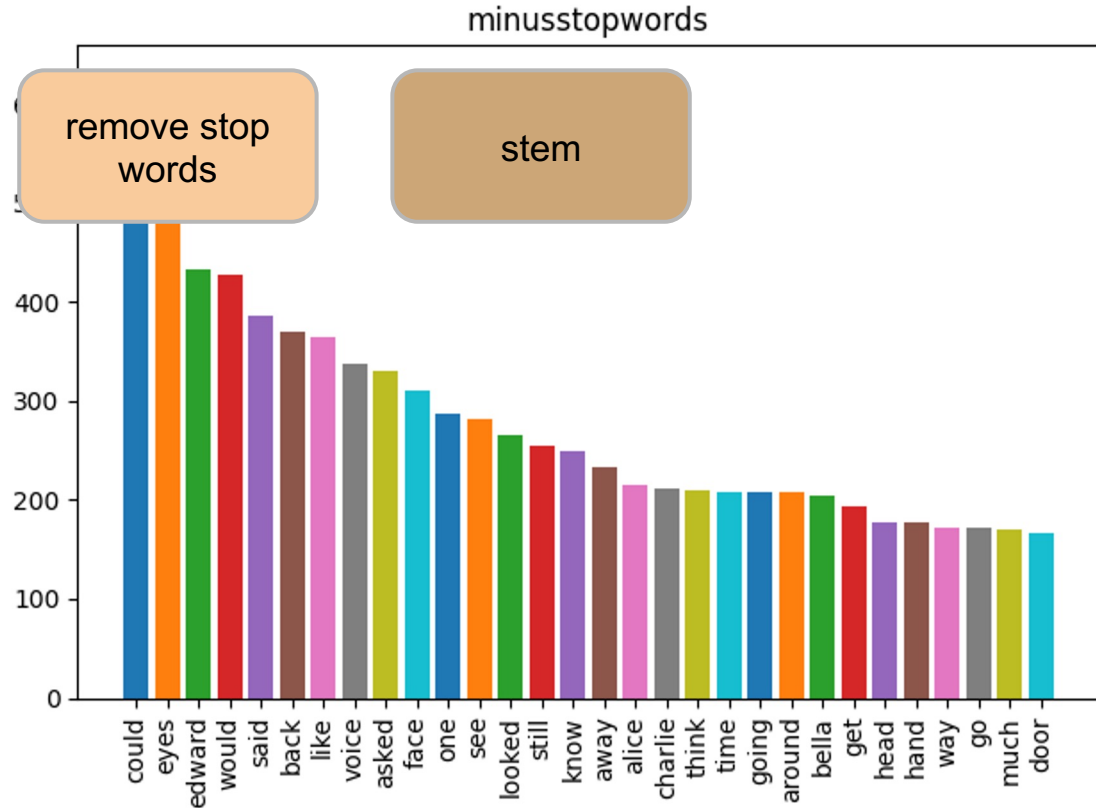
# A feasible sequence...

lower case

eliminate  
punctuation

List of common, unhelpful words compiled by nltk from large corpora. We keep words that aren't in that list.

More sophisticated approach is called tf-idf...



# A feasible sequence...

lower case

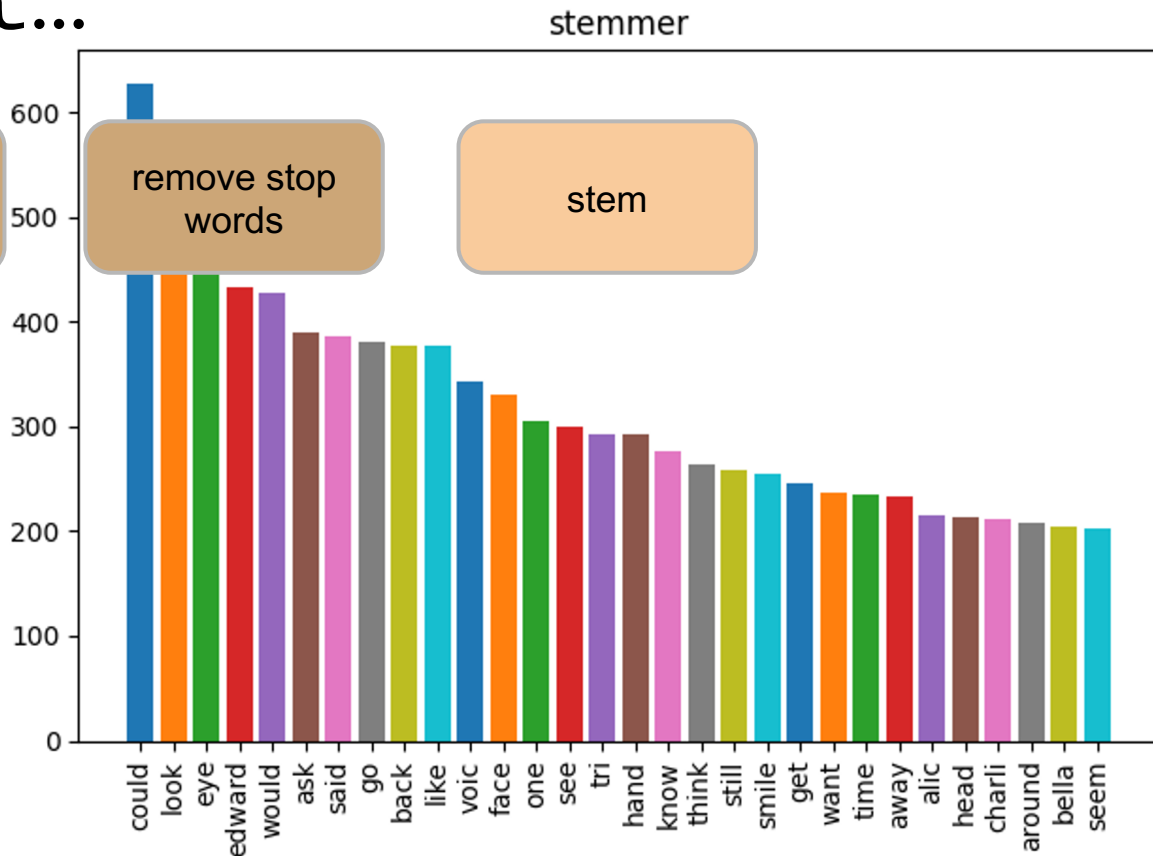
eliminate  
punctuation

remove stop  
words

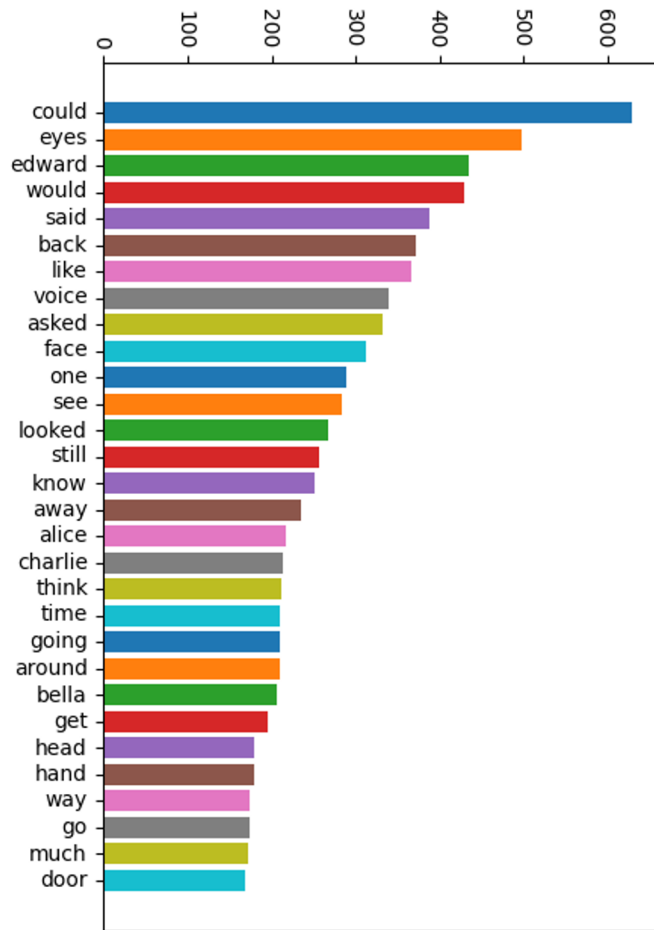
stem

“goes” -> “go”  
“running” -> “run”  
“eaten” -> “eat”

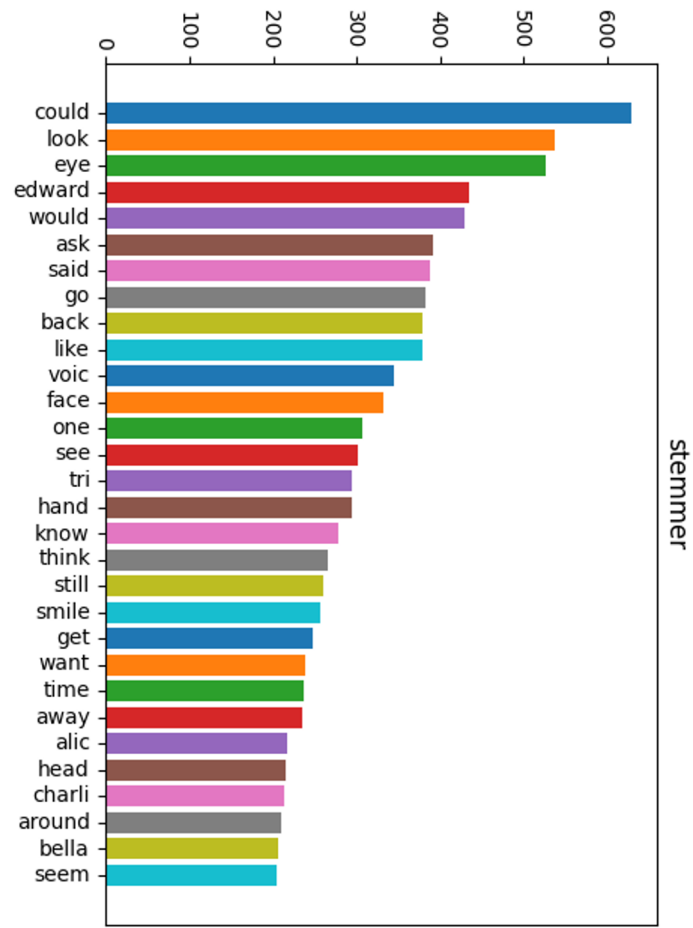
NLTK provides the stemmer



# Curious?



minusstopwords



stemmer

# Resources...

[https://classroom.github.com/a/QOTyz\\_JX](https://classroom.github.com/a/QOTyz_JX)

<https://www.nltk.org/book/>

<https://historynewsnetwork.org/article/33359>