Programming, Problem Solving, and Algorithms

CPSC203, 2019 W1

Announcements

“Problem of the Day” continues!

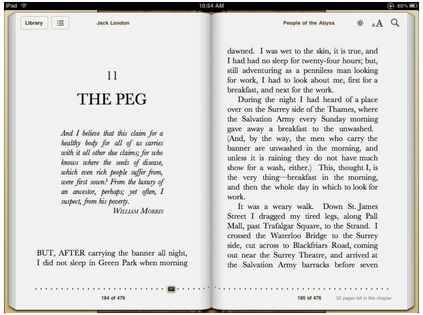
Today:

Visualizing Literature

Natural Language Processing

Named Entity Recognition

Visualizing Literature



11

THE PEG

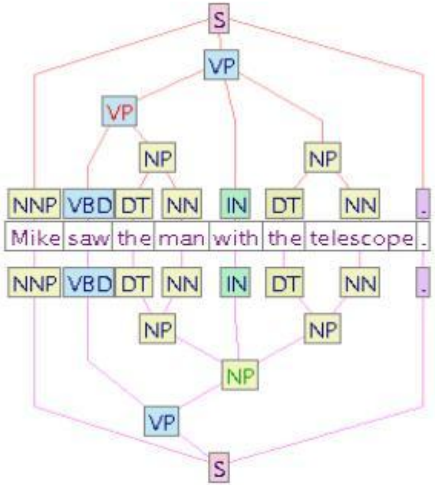
And I believe that this claim for a healthy body for all of us carries with it all other due claims; for who knows where the seeds of disease, which even rich people suffer from, were first sown? From the luxury of an ancestor, perhaps; yet often, I suspect, from his poverty.
William Morris

BUT, AFTER carrying the banner all night, I did not sleep in Green Park when morning

dawned. I was wet to the skin, it is true, and I had had no sleep for twenty-four hours; but, still adventuring as a penniless man looking for work, I had to look about me, first for a breakfast, and next for the work.

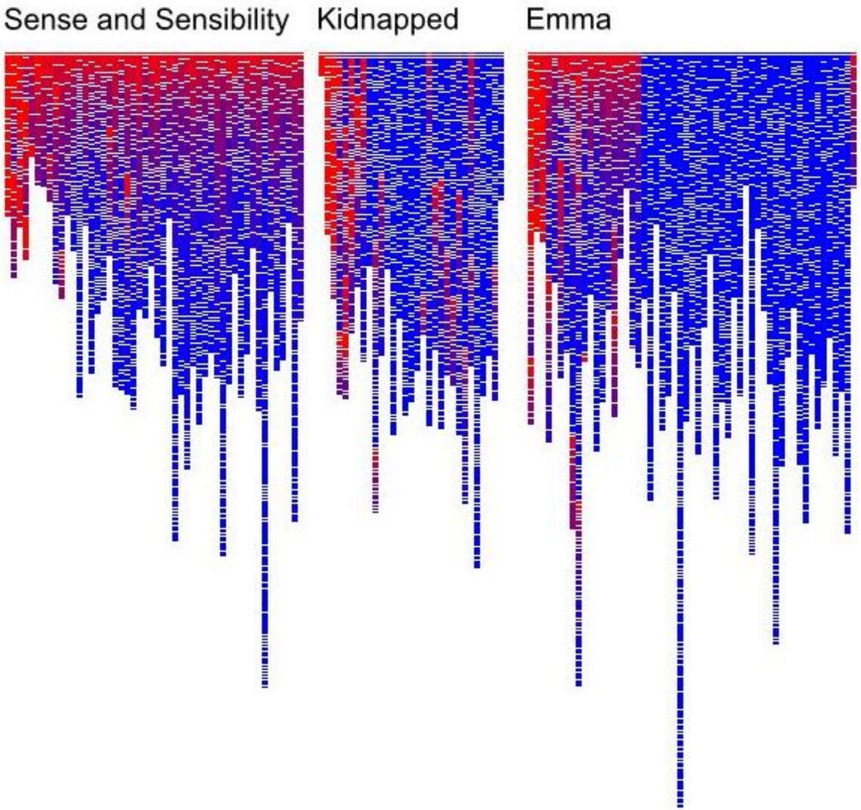
During the night I had heard of a place over on the Surrey side of the Thames, where the Salvation Army every Sunday morning gave away a breakfast to the unwashed. (And, by the way, the men who carry the banner are unwashed in the morning, and unless it is raining they do not have much show for a wash, either.) This, thought I, is the very thing—breakfast in the morning, and then the whole day in which to look for work.

It was a weary walk. Down St. James Street I dragged my tired legs, along Pall Mall, past Trafalgar Square, to the Strand. I crossed the Waterloo Bridge to the Surrey side, cut across to Blackfriars Road, coming out near the Surrey Theatre, and arrived at the Salvation Army barracks before seven



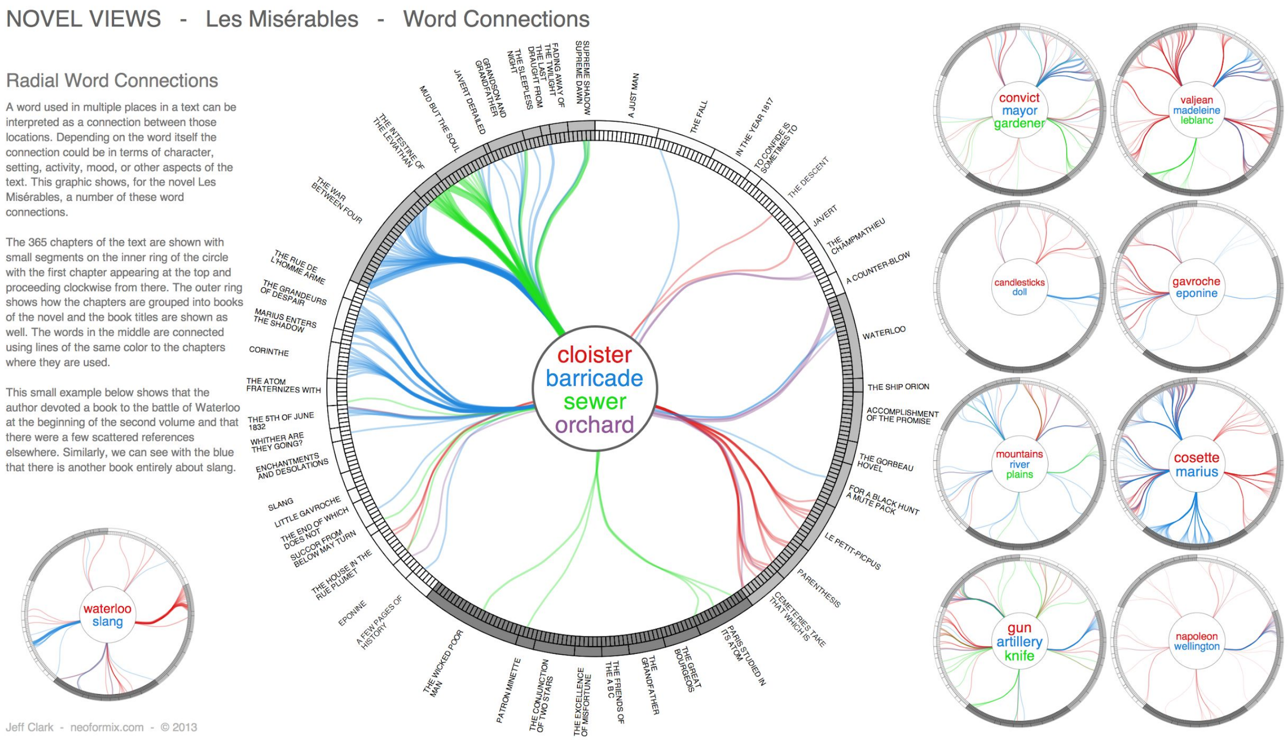
Collision Detection
Collapsible Force Layout
Force-Directed States
Azimuthal Projections
Box Plots
Bubble Chart
Bullet Charts
Calendar View
Choropleth
Collapsible Tree Layout
Zoomable Treemap
Zoomable Partition Layout
Non-contiguous Cartogram
Chord Diagram
Dendrogram
Force-Directed Graph
Zoomable Area Chart
Calendar View
Rotating Cluster Layout
Sankey Diagram
Circle Packing
Population Pyramid
2000
Stacked Bars
Streamgraph
Fisheye Distortion
Hive Plot
Co-occurrence Matrix
Motion Chart
Sunburst
Node-Link Tree
Treemap
Voronoi Diagram
Chord Diagram
Animated Béziers
Zoomable Sunburst
Collatz Graph
Hierarchical Edge Bundling
Voronoi Diagram
Symbol Map
Parallel Coordinates
Parallel Sets
Word Cloud
Cloud
Obama's Budget Proposal
Facebook IPO
Scatterplot Matrix
Zoomable Pack Layout
Hierarchical Bars
Epicyclical Gears

Example



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.

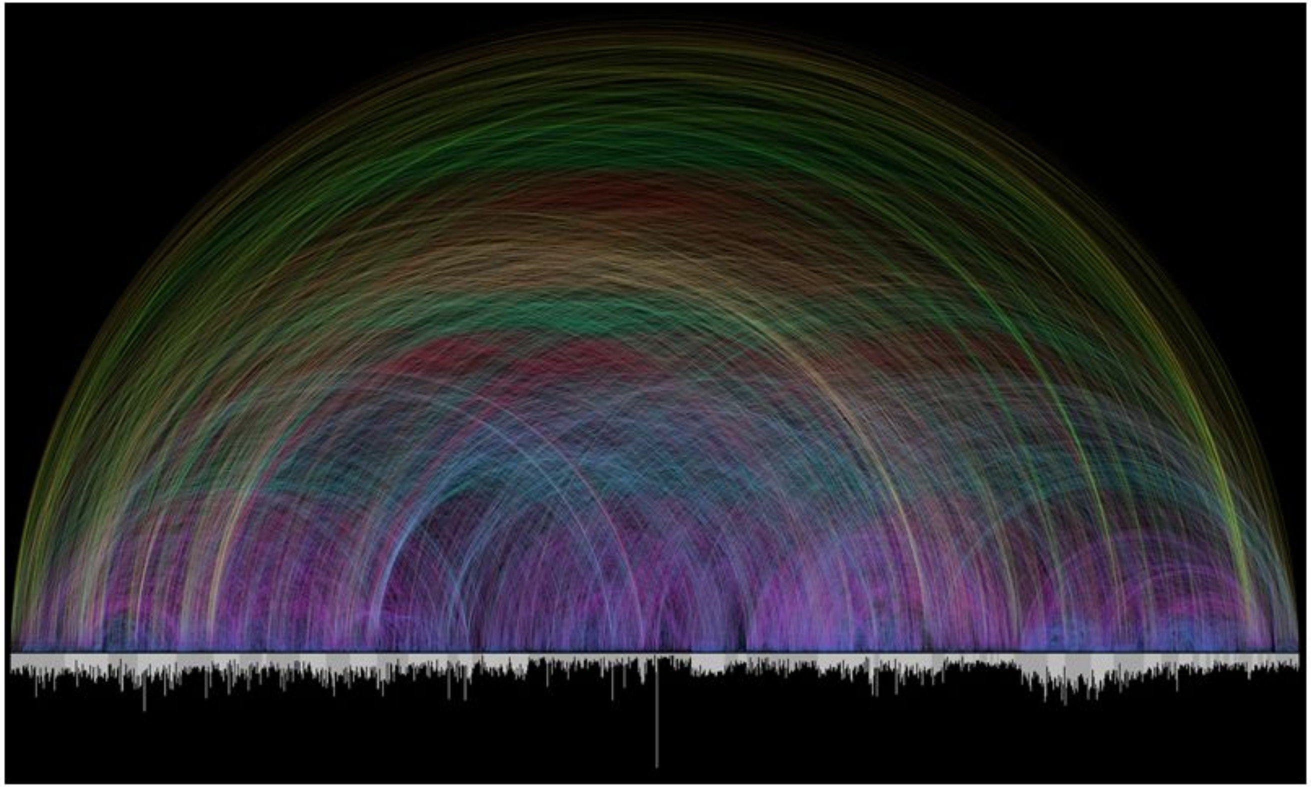
The 365 chapters of the text are shown with small segments on the inner ring of the circle with the first chapter appearing at the top and proceeding clockwise from there. The outer ring shows how the chapters are grouped into books of the novel and the book titles are shown as well. The words in the 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 at the beginning of the second volume 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

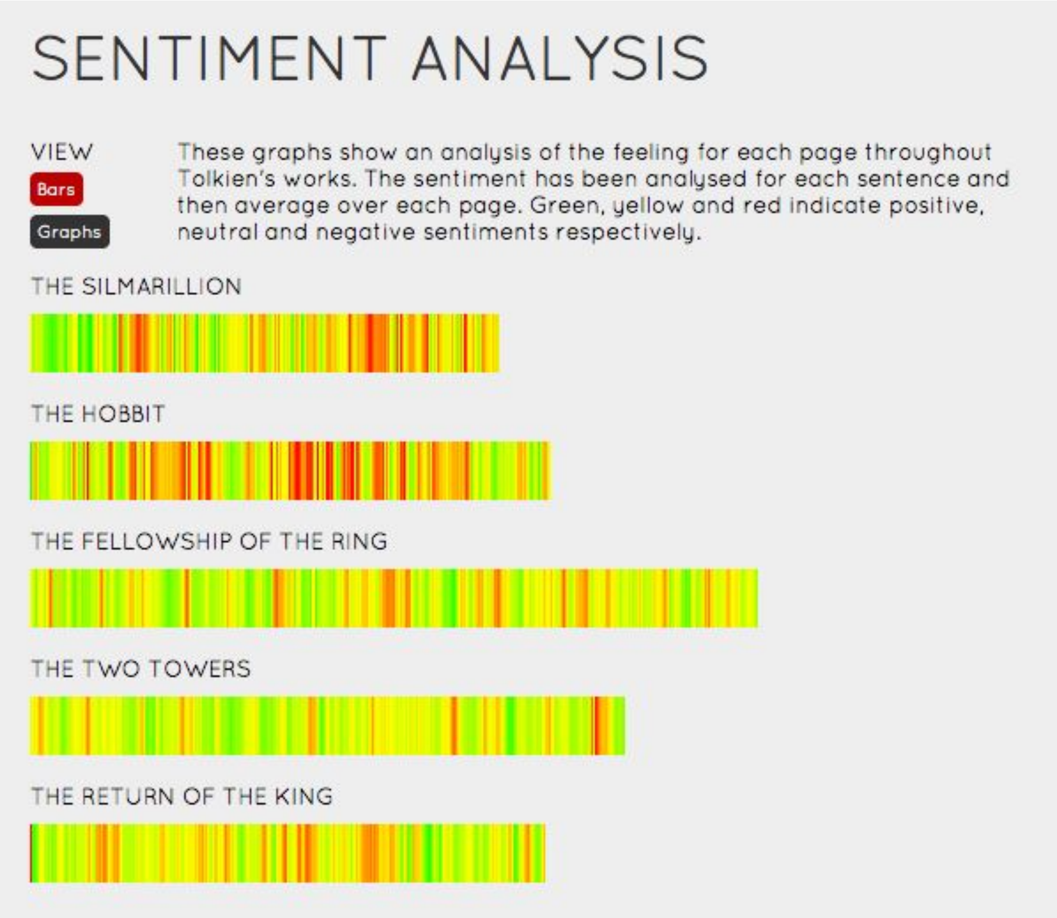
http://neoformix.com/2013/NovelViews.html

Example



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

Example

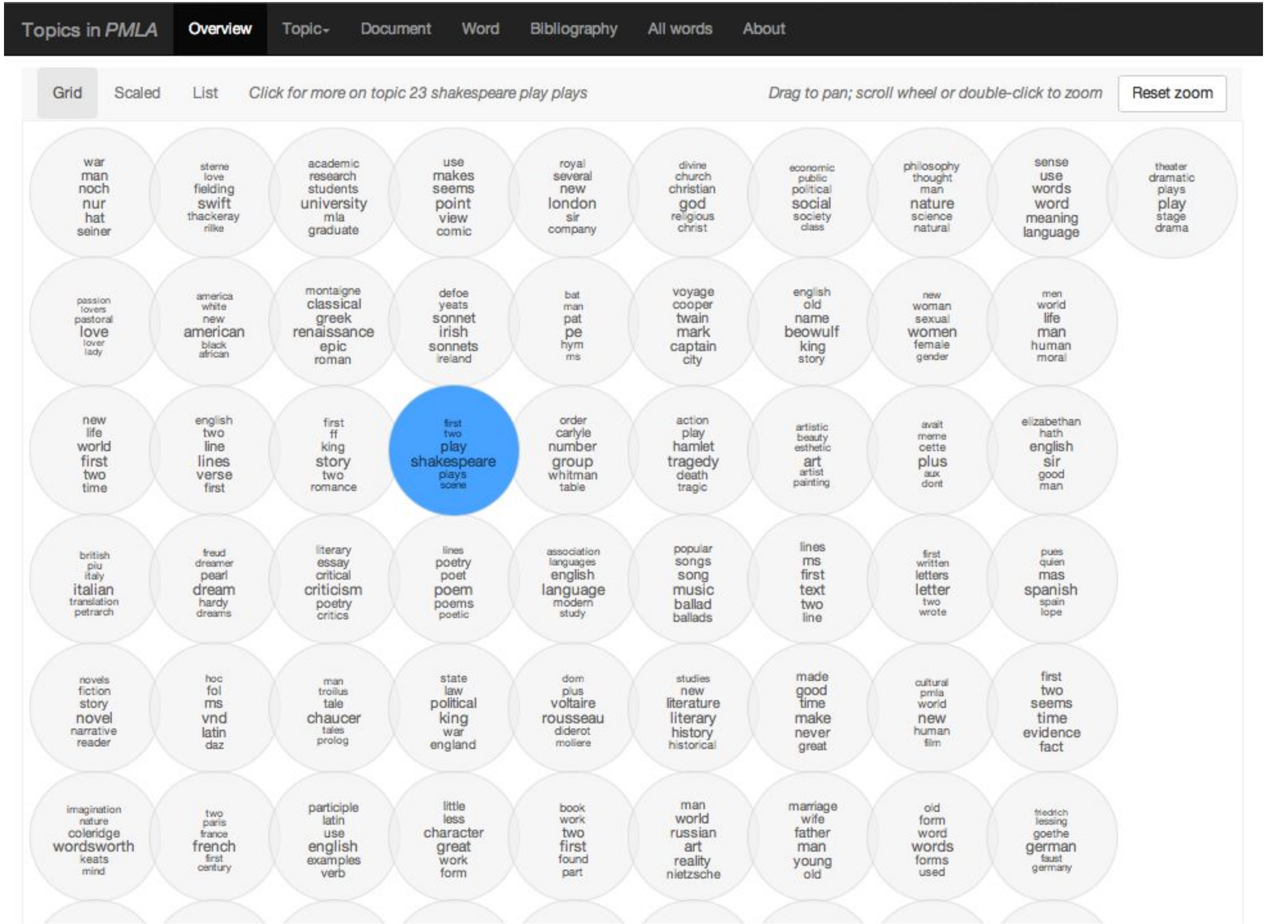


SENTIMENT ANALYSIS

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.

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

Example



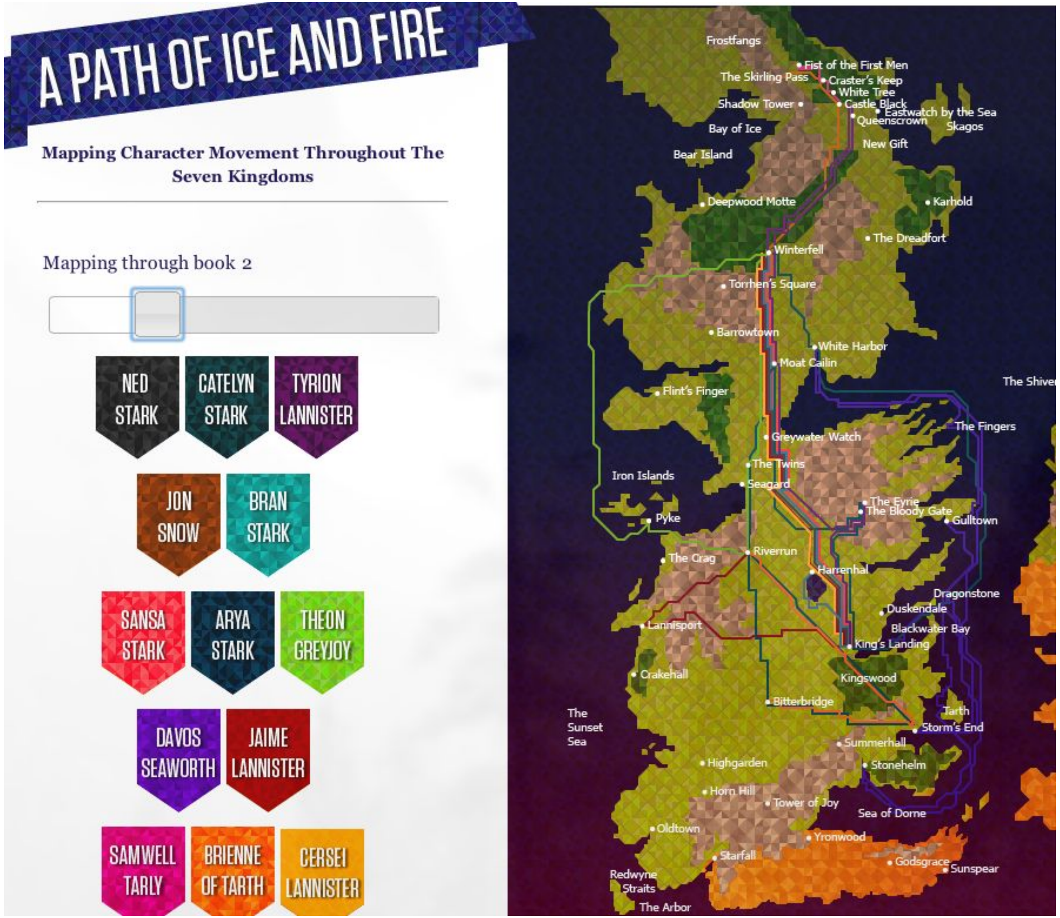
Click for more on topic 23 shakespeare play plays

Drag to pan; scroll wheel or double-click to zoom

war man noch nur hat seiner	sterne love fielding swift thackeray rilke	academic research students university mla 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 play stage drama
passion lovers 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	men 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	avait meme cette plus aux dont	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 quien mas spanish spain lope	
novels fiction story novel narrative reader	hoc fol ms vnd latin daz	man troilus tale chaucer tales prolog	state law political king war england	dom plus voltaire rousseau diderot moliere	studies new literature literary history historical	made good time make never great	cultural pmla 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 words forms used	friedrich lessing goethe german faust germany	

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

Example



A PATH OF ICE AND FIRE

Mapping Character Movement Throughout The Seven Kingdoms

Mapping through book 2

http://www.emelynbaker.com/westeros.html

Example

The New York Times

http://www.nytimes.com/newsgraphics/2013/12/30/year-in-interactive-storytelling/

How do we begin?

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

returns a string.

We want to analyze the data by word or by ________ or by ________ or by ________...

can do this using nltk's "tokenizer"

Tokenization

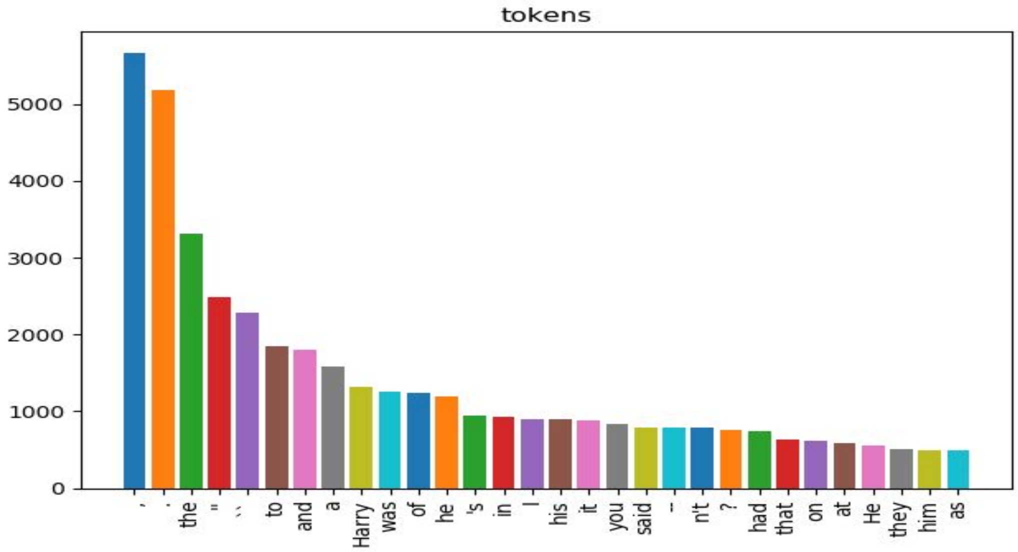
Translate: “Astrology. The governess was always\ngetting muddled with her astrolabe, and when she got specially muddled she would take it out\nof the Wart by rapping his knuckles. She did not rap Kay's knuckles, because when Kay grew\nolder”

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']

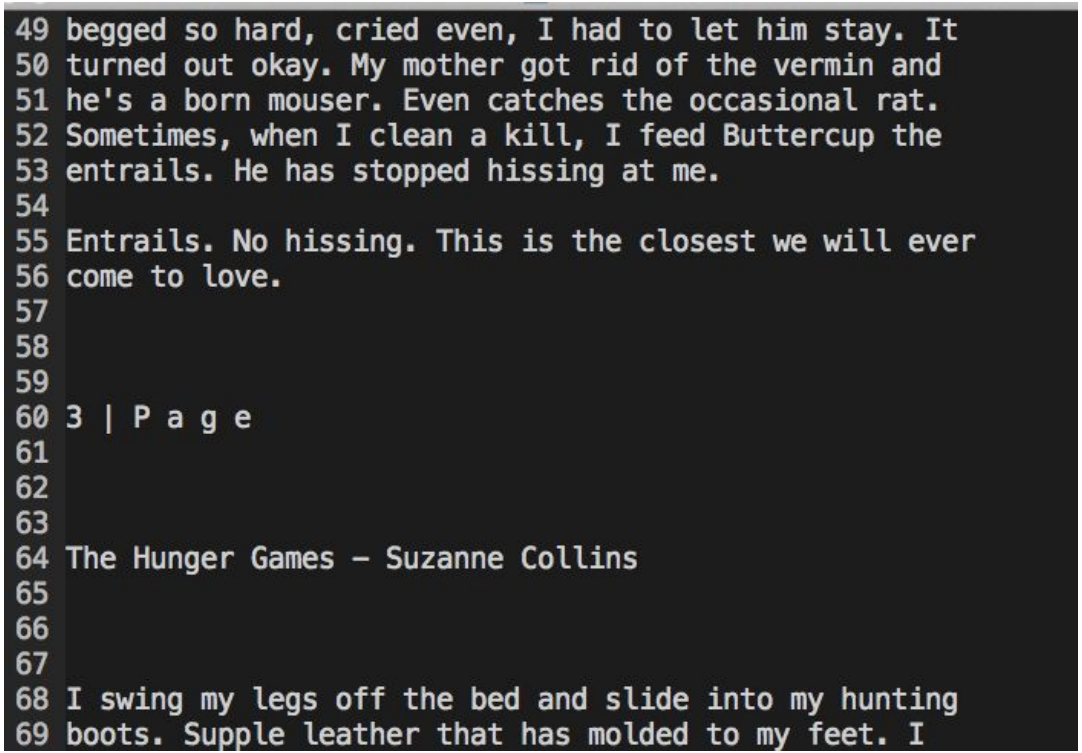
http://text-processing.com/demo/tokenize/

Python Demo

The python script in “LecHP” was assembled from examples in Ch1-3 of the NLTK book. http://www.nltk.org/book/



Pre-processing



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

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



3 | P a g e



The Hunger Games – Suzanne Collins



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

A feasible sequence…

lower case

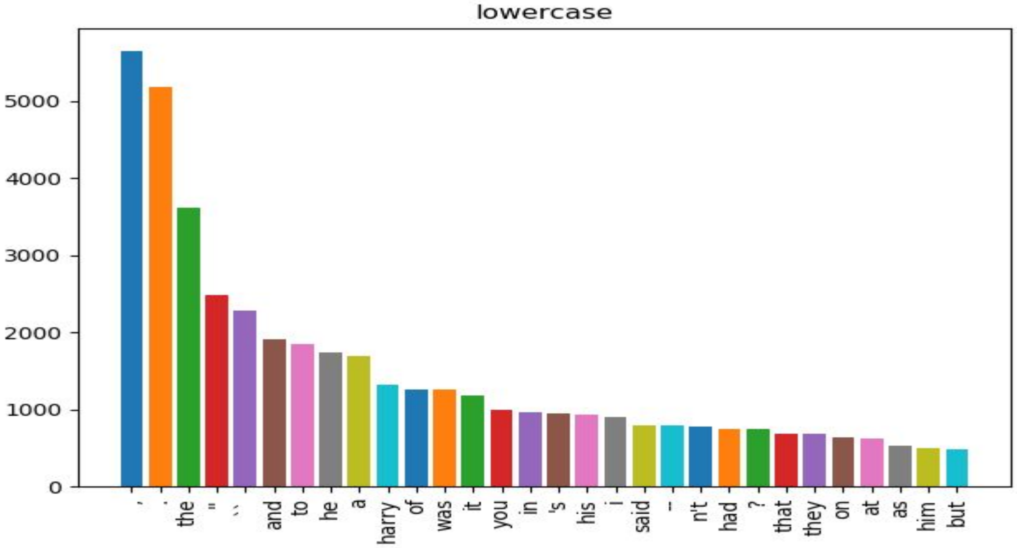
eliminate punctuation

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

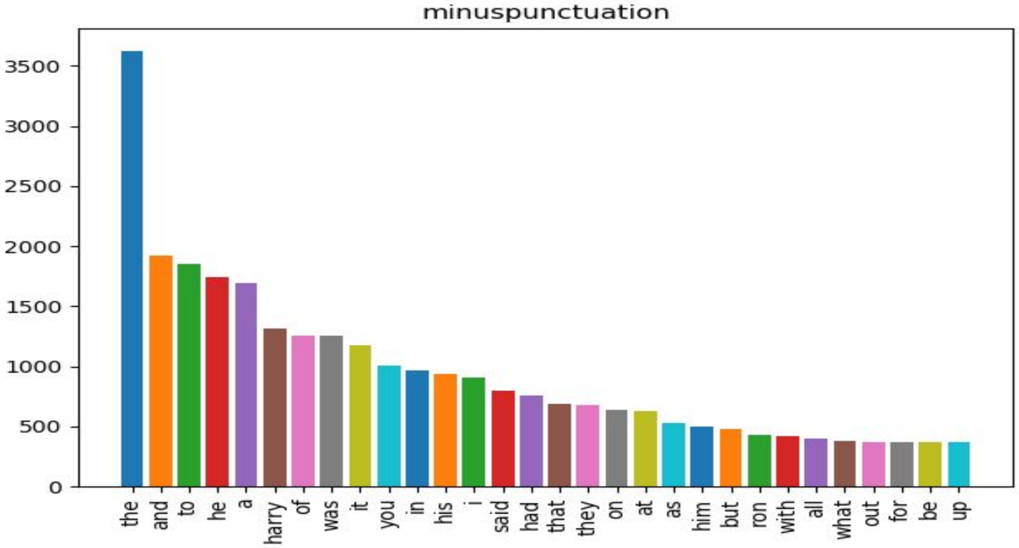
eliminate punctuation

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

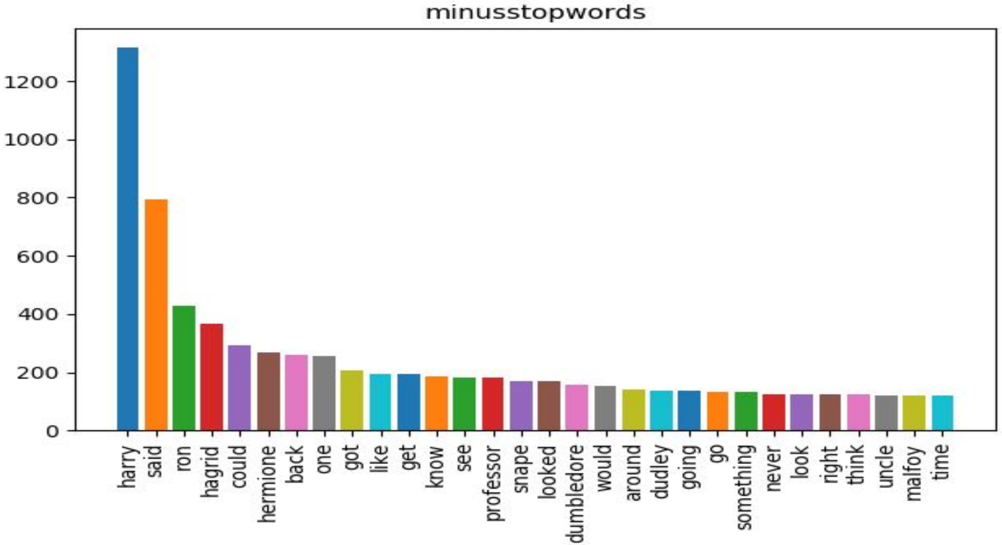
eliminate punctuation

remove stop words

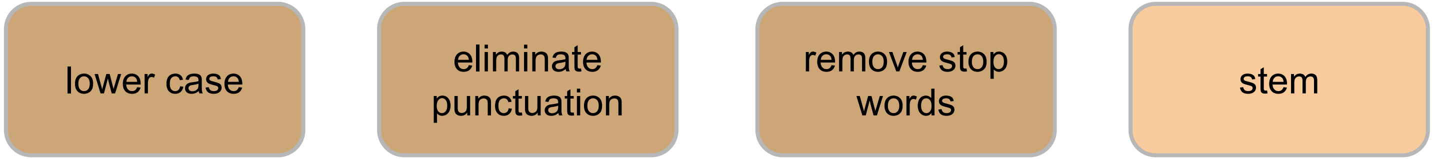
stem

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

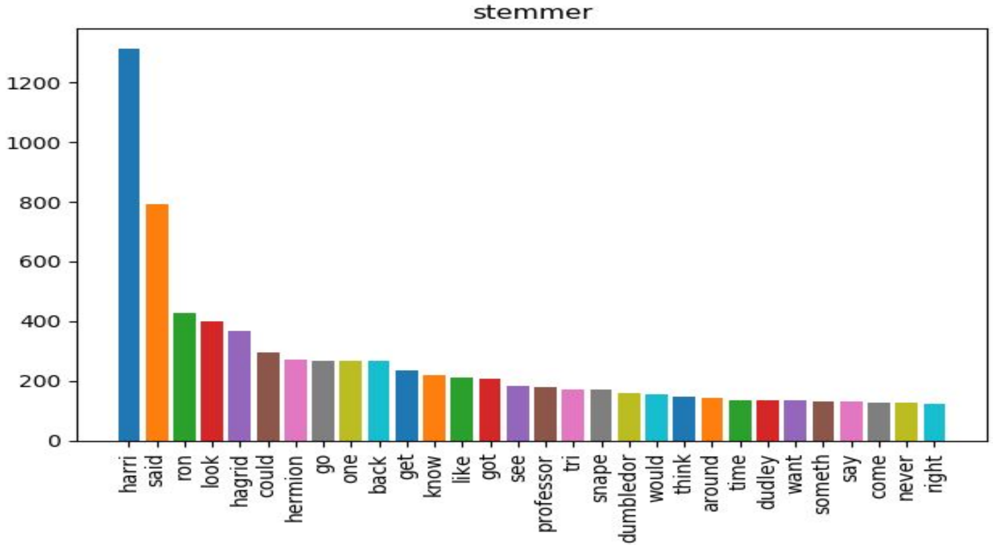
eliminate punctuation

remove stop words

stem

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

NLTK provides the stemmer



Parts of Speech…

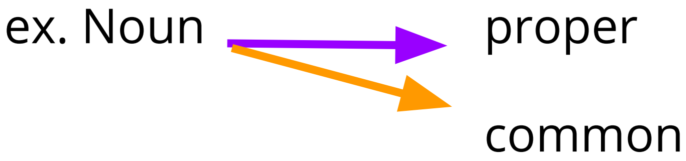
Middle School Grammar:

Noun, Verb, Adjective, Adverb, Preposition, Conjunction, Pronoun, Interjection

open

closed

Subdivide classes!



ex. Noun

proper

common

UPenn POS

POS inference is hard!

Plug in the curling iron. JJ

I want to learn to play curling. NN

The ribbon is curling around the Maypole. VBG

Search space

ex.	Plays	well	with	others.
	NNS	UH	IN	NNS
	VBZ	JJ		
		NN		
		RB		

Word senses come from lexicon (like wordNet)

Search space

Only 11% of English words have more than one POS, but…

they tend to be very common words.

I know that he is honest.

That movie was fantastic!

I wouldn’t go that far.

Accuracy and expectations...

- modern POS taggers achieve 97% accuracy. w00t!

- tagging w most frequent POS gets ______ .

- humans achieve ______ agreement.

One last example:

Bill	saw	that	man	yesterday.
NNP	NN	DT	NN	NN
VB	VB	IN	VB	RB
NN				

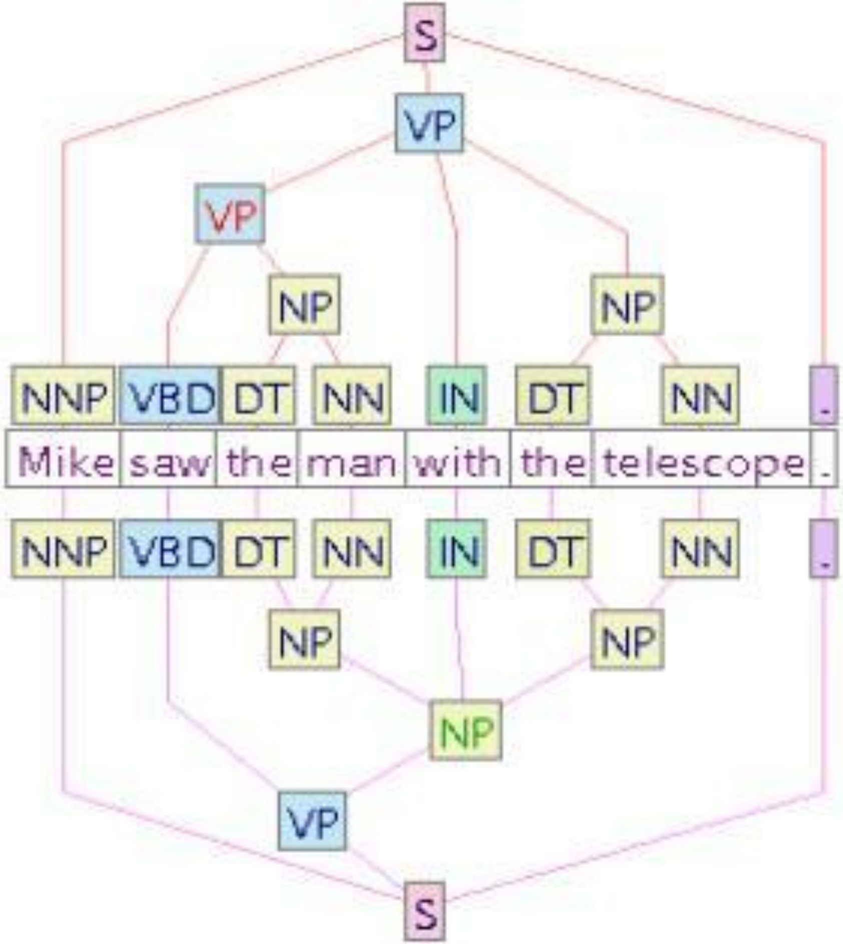
- man is rarely a verb
- VB never follows DT

96.6% on known words
86.8% on unknown words

POS tagging in NLTK

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

Ch5: Categorizing and Tagging Words



Named Entity Recognition (NER)

1. Underline all of the proper nouns (named entities) in this text..

Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense.

Mr. Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large mustache. Mrs. Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbors. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere.

Typical categories of entities are PERSON, LOCATION, ORGANIZATION. Think about how you might discover each of the entities using a program.

NLTK NER discovery...

2. Repo LecHP contains a file called test.py. Modify and execute this file to answer the following questions. In each case, sketch an example of the output, and explain it briefly in English.

a. if `textRaw` is the string above, what is the result of

```
sents = sent_tokenize(text)
```

b. if `sents` is the result of part a, what is the result of

```
sentWords = [word_tokenize(s) for s in sents if s]
```

NLTK NER discovery...

c. if `sentWords` is the result of part b, what is the result of

```
sentWordsPOS = [pos_tag(s) for s in sentWords]
```

d. if `sentWordsPOS` is the result of part c, what is the result of

```
sentWordsNER = [ne_chunk(s) for s in sentWordsPOS]
```

e. if `sentWordsNER` is the result of part d, what is the result of

```
subtrees = [chunk.subtrees() for chunk in sentWordsNER]
```

NLTK NER discovery...

f. if `subtrees` is the result of part e, what is the result of

```
entities = [[ s for s in st if s.label() == "PERSON"] for st in subtrees]
```

g. if `entities` is the result of part f, what is the result of

```
entities = [[ ' '.join(c[0] for c in s.leaves()) for s in st] for st in entities]
```

3. Write python code that would extract all the verbs from the text above. The answer to problem 2c will help you!

4. (challenge) Write a function `personVerbs(person, text)`, that returns a list of all the verbs that occur in sentences that also contain `person`.

POTD #38 Tue

https://github.students.cs.ubc.ca/cpsc203-2019w-t1/potd36

Describe any snags you run into:

1. Line ___: ____
2. Line ___: ____
3. Line ___: ____
4. Line ___: ____
5. Line ___: ____

ToDo for next class...

POTD: Continue every weekday! Submit to repo.

Reading: TLACS Ch 10 & 12 (lists and dictionaries)

References:

https://www.youtube.com/watch?v=wsSEKm-rU6U

https://github.com/gboeing/osmnx-examples/tree/master/notebooks

https://gist.github.com/psychemedia/b49c49da365666ba9199d2e27d002d07