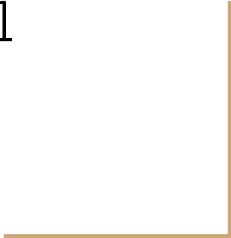


Programming, Problem Solving, and Algorithms

CPSC203, 2019 W1



Announcements

~~"Problem of the Day" continues!~~

I think canvas is live. POTD 1-27 recorded + MT.

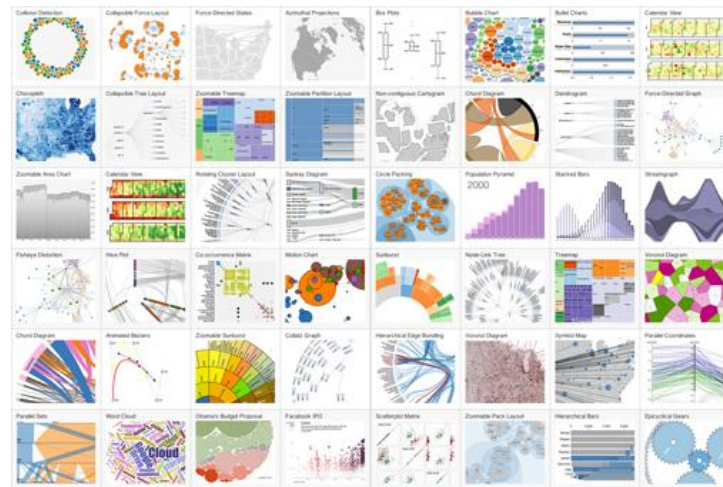
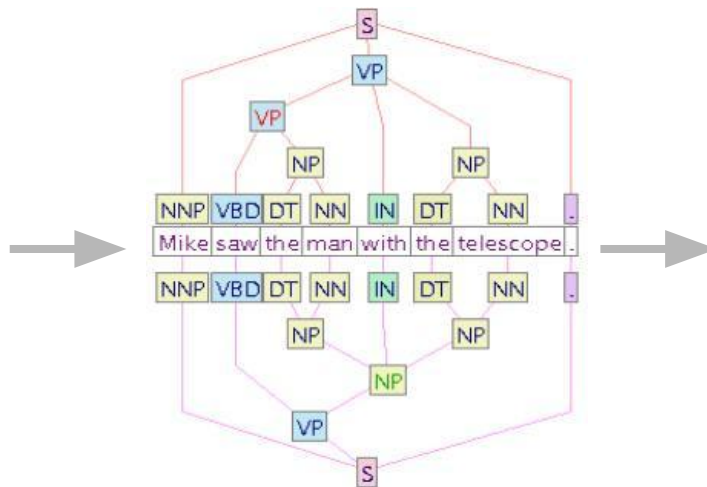
Final exam: 12/11 noon, here.

Today:

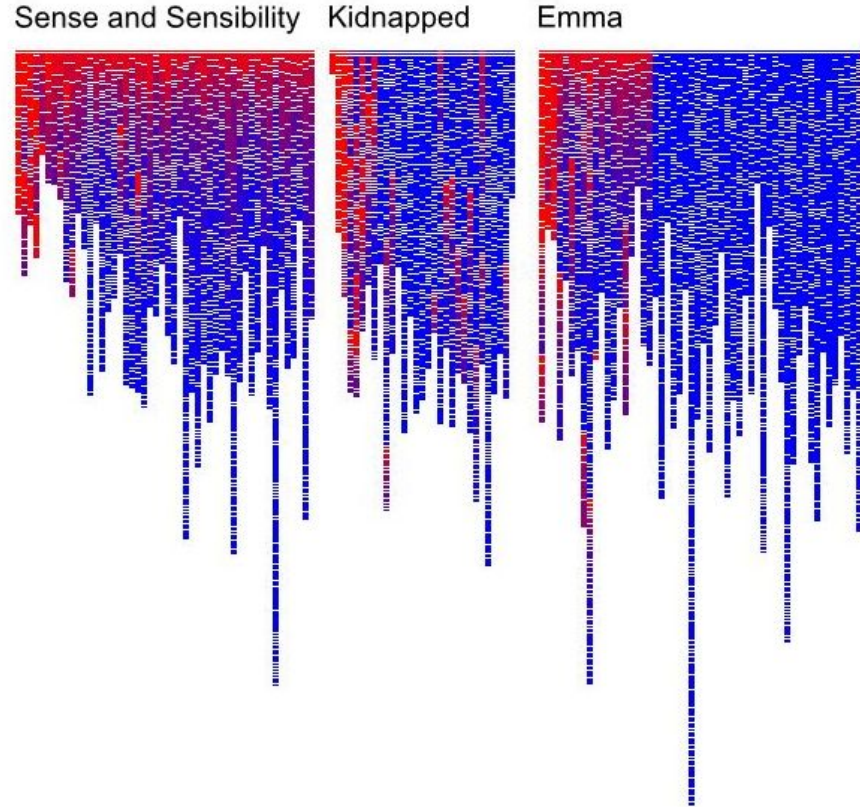
Visualizing Literature

Natural Language Processing

Named Entity Recognition



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

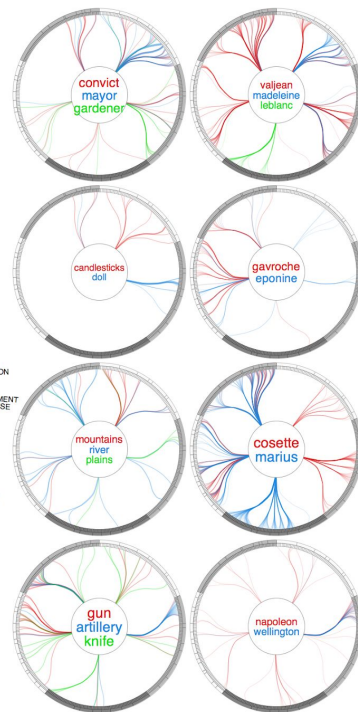
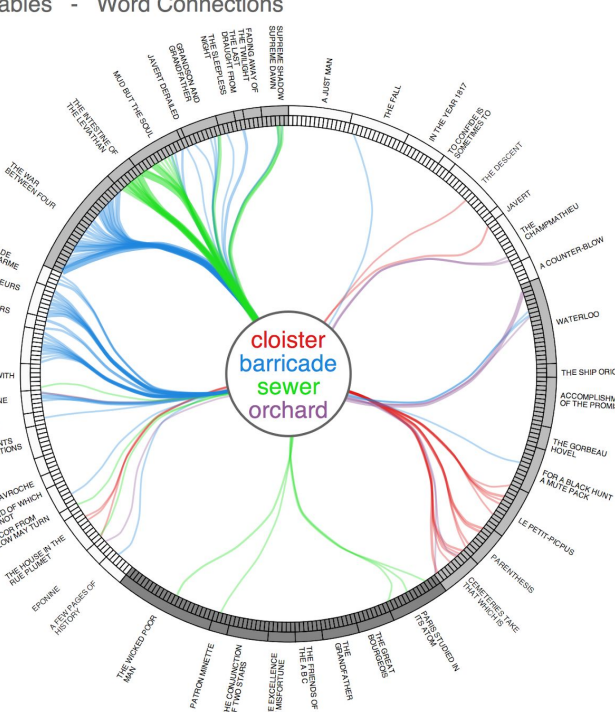
A word used in multiple places in a text can be interpreted as a connection between those locations. Depending on the word itself the connection could be in terms of character, setting, activity, mood, or other aspects of the text. This graphic shows, for the novel Les Misérables, a number of these word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

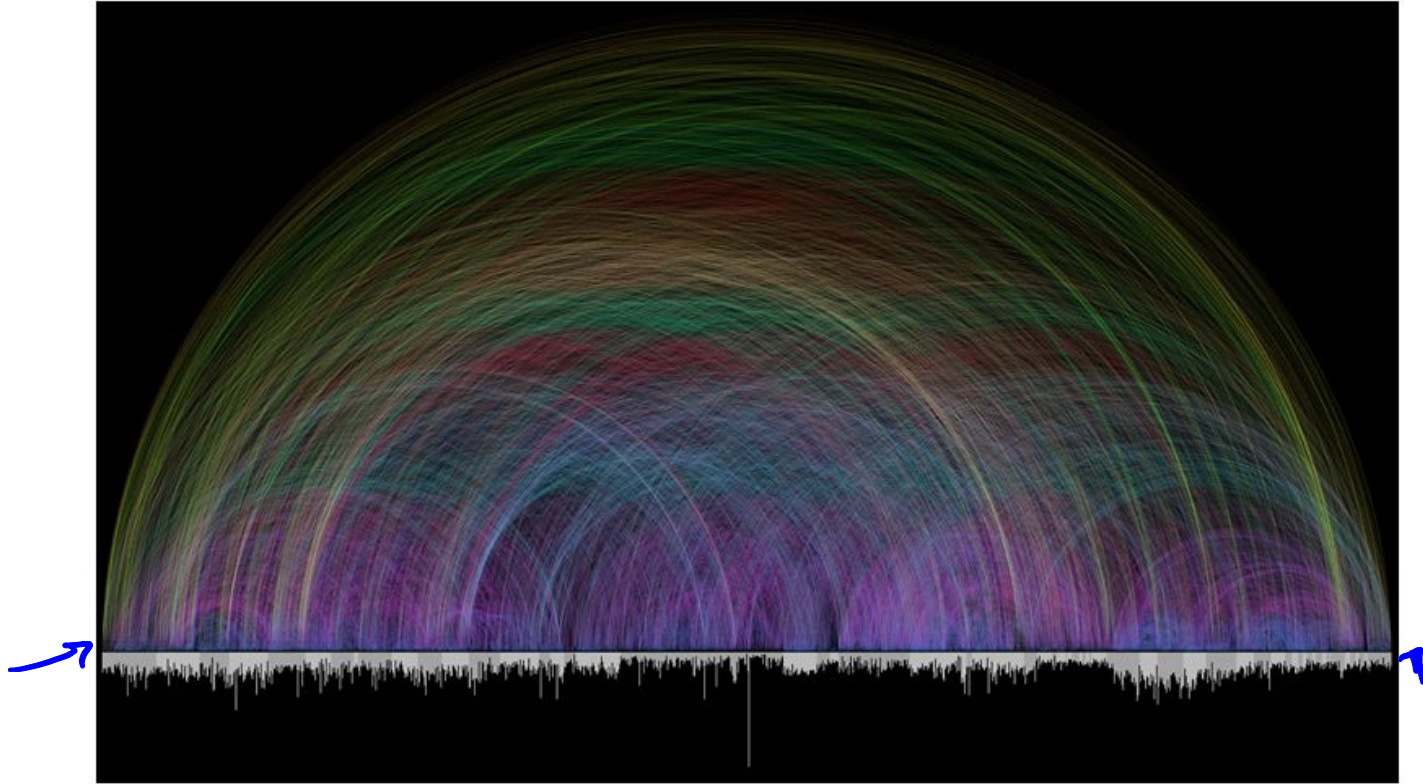


Jeff Clark - neoformix.com - © 2013



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

Example



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

Example

SENTIMENT ANALYSIS

VIEW

Bars

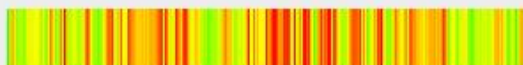
Graphs

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.

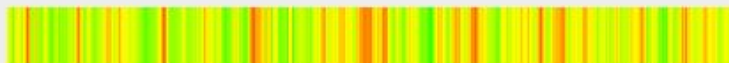
THE SILMARILLION



THE HOBBIT



THE FELLOWSHIP OF THE RING



THE TWO TOWERS

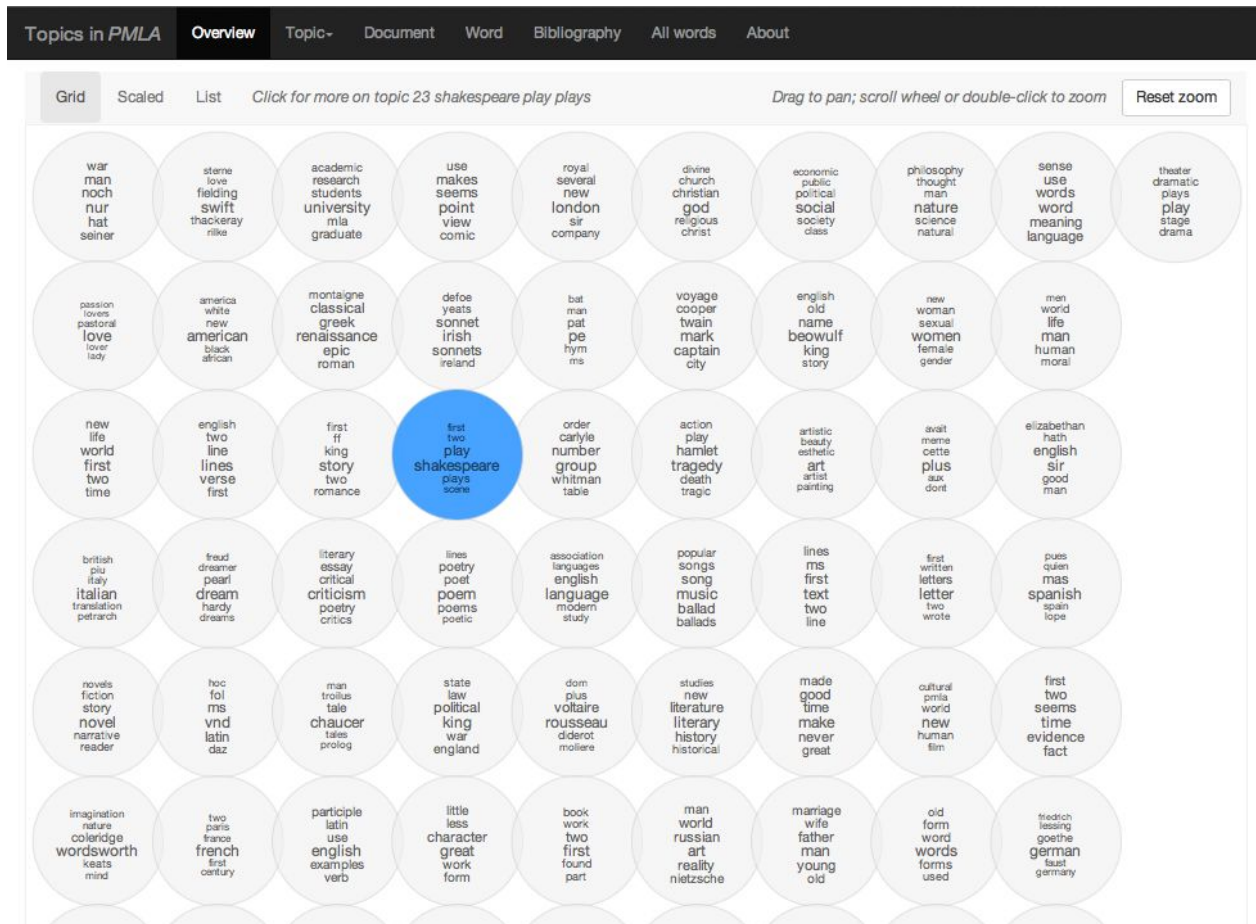


THE RETURN OF THE KING



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

Example



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

Example



Example

The New York Times

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

How do we begin?

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

returns a string.

We want to analyze the data by word or by sentences or by paragraphs or by chapter books

can do this using nlk's "tokenizer"

natural language toolkit

A handwritten diagram in blue ink. At the bottom, the words 'natural language toolkit' are written. Above this, the word 'tokenizer' is underlined and enclosed in a blue oval. Two blue arrows originate from the oval: one points diagonally up and to the left towards the word 'string' in the text 'returns a string.', and the other points diagonally up and to the right towards the word 'sentences' in the text 'We want to analyze the data by word or by sentences or by paragraphs or by chapter books'.

Tokenization

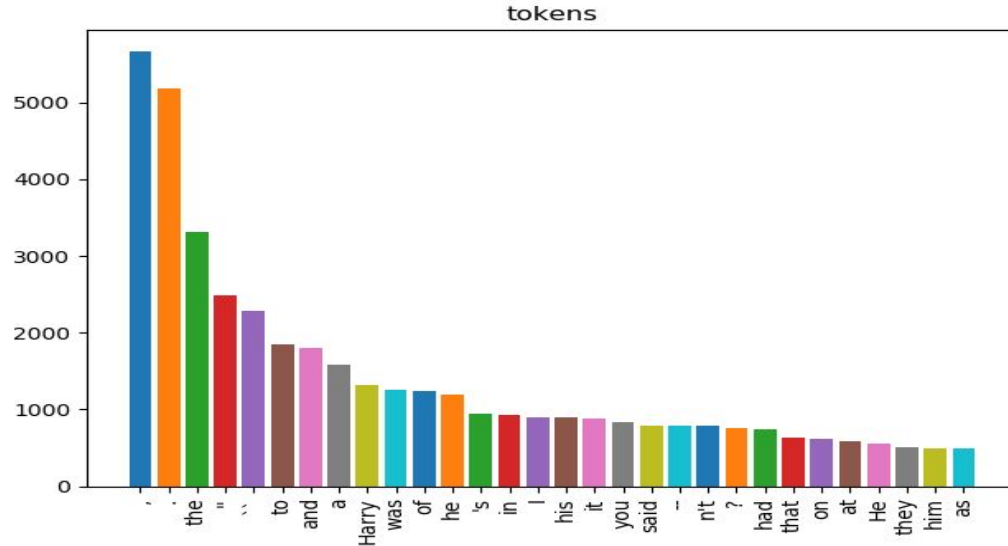
Once + Future King

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 "LecHP" was assembled from examples in Ch1-3 of the NLTK book. <http://www.nltk.org/book/> ←



Pre-processing

*a necessary evil... addressed by
getting to know
your data.*

```
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
```

gutenberg.org

A feasible sequence... *for data cleaning*

lower case

eliminate
punctuation

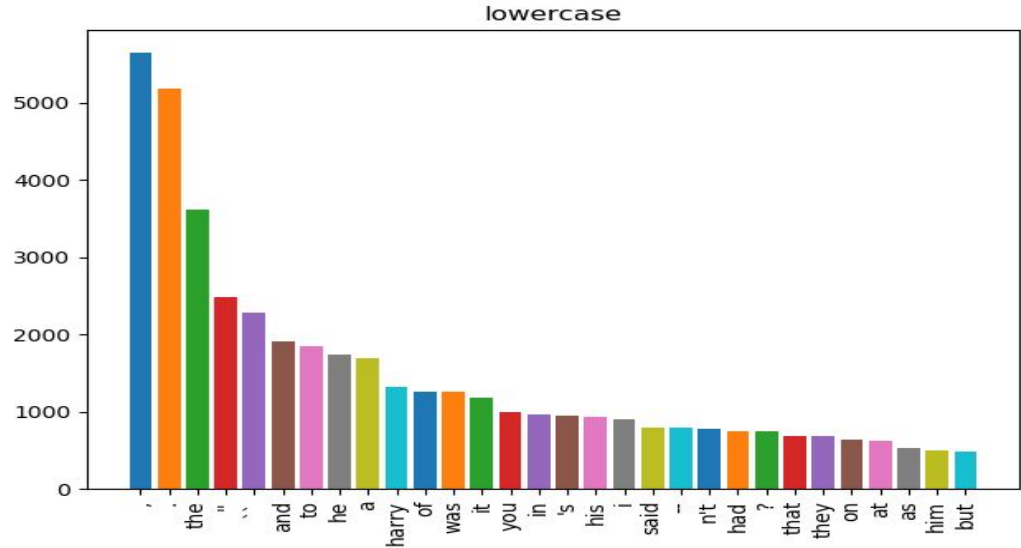
remove stop
words

stem

Unify tally for “Valor” and
“valor”

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

The != the



A feasible sequence...

lower case

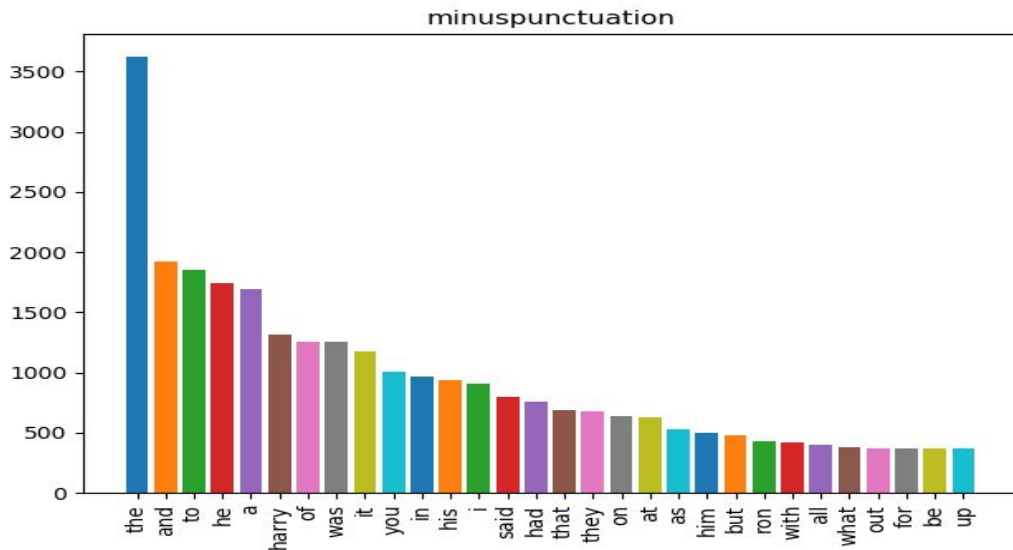
eliminate
punctuation

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

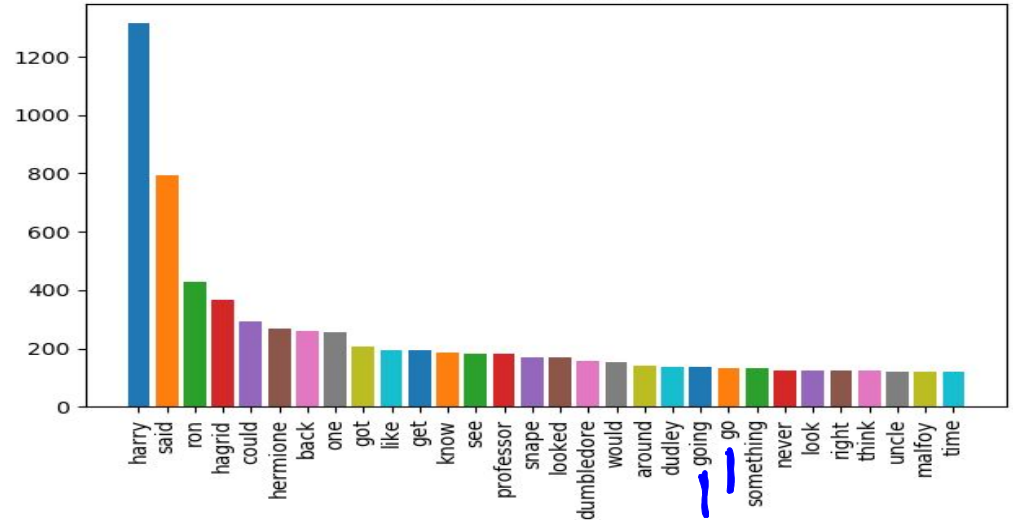
remove stop
words

stem

minusstopwords

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

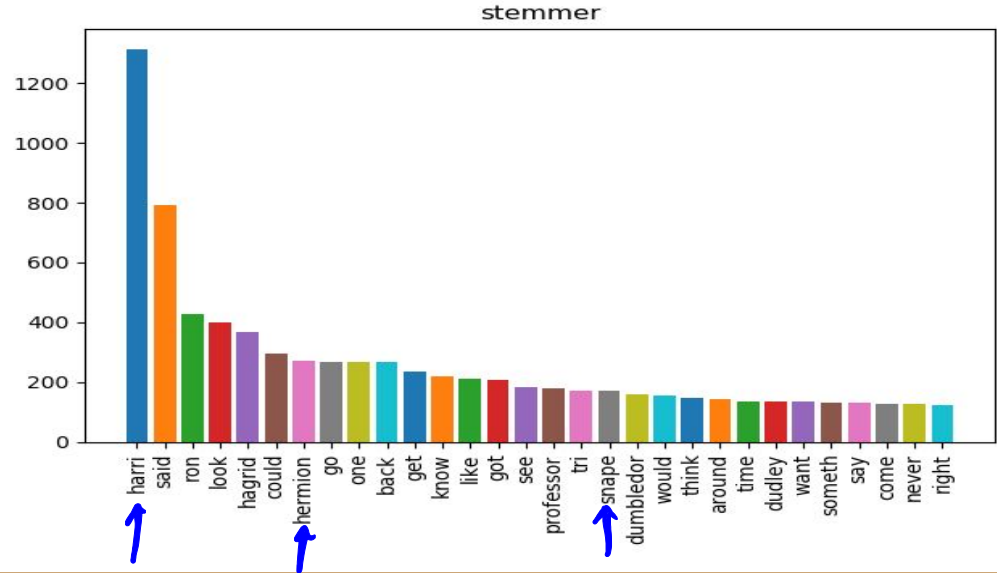
stem

“goes” -> “go”

“running” -> “run”

“eaten” -> “eat”

NLTK provides the stemmer

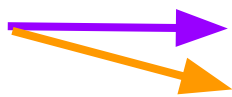


Parts of Speech...

Middle School Grammar:

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

Subdivide classes!

ex. Noun  proper
common

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

wordNet is
resource for
defn + pos
given any
word.

Search space

ex. Plays well with others. $2.4 = 8$

{ NNS
VBZ

UH.
JJ.
NN.
RB

IN

NNS

of different
labellings is
product of
possible word
senses.

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.

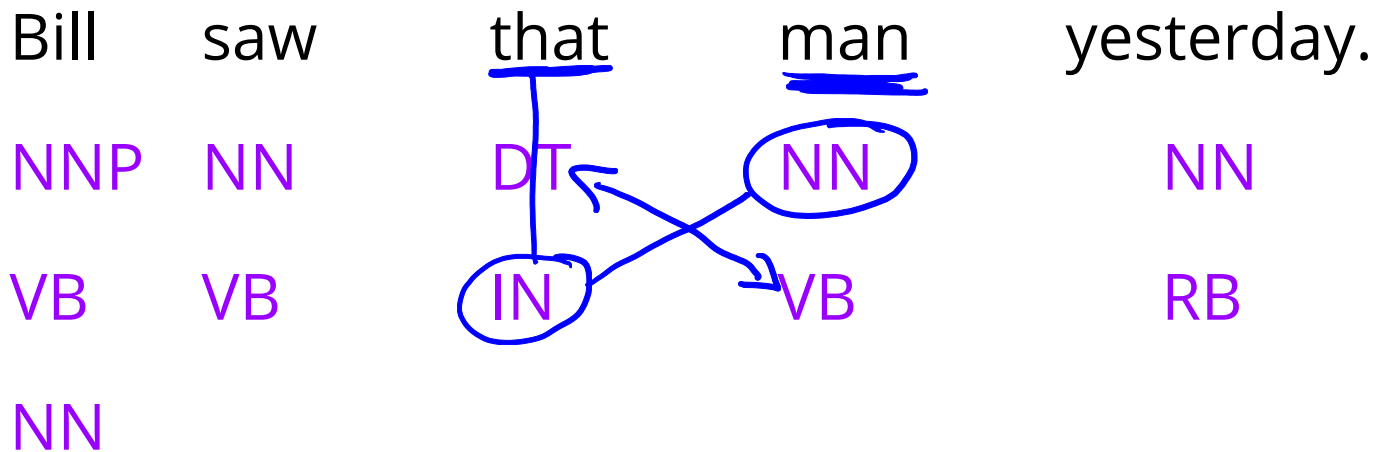
That movie was fantastic!

I wouldn't go that far.

Accuracy and expectations...

- modern POS taggers achieve 97% accuracy. w00t!
- tagging w most frequent POS gets 90%.
- humans achieve 98% agreement.

One last example:



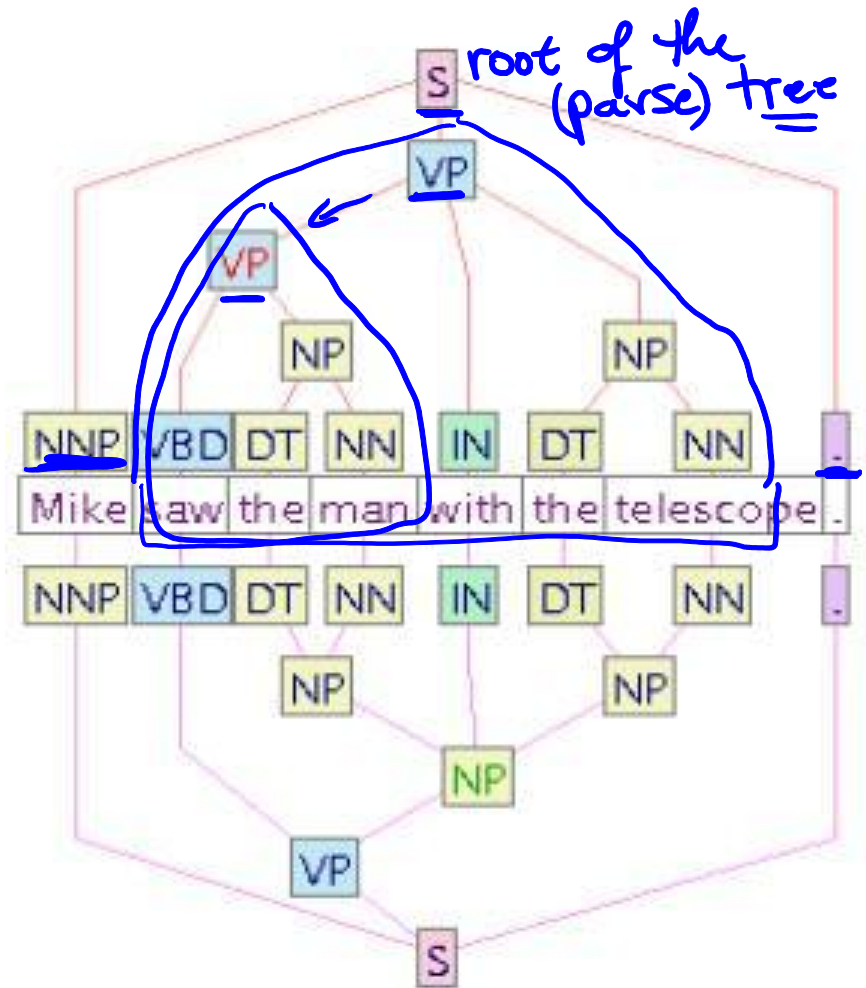
- man is rarely a verb
- VB never follows DT

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>