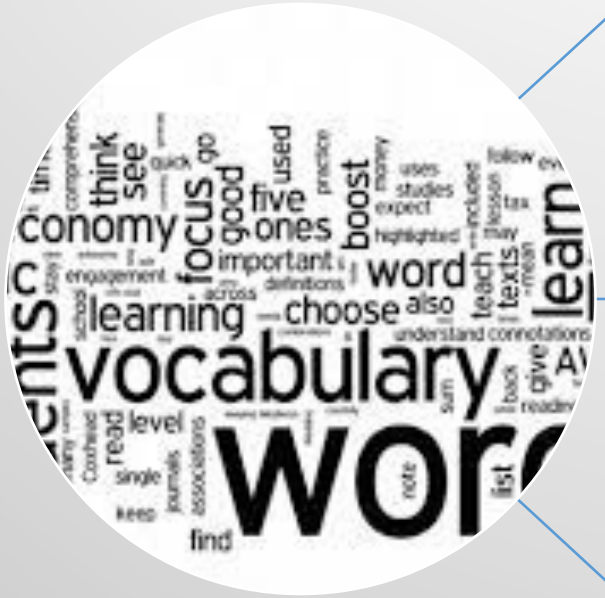
Natural Language Processing

Dr. Karen Mazidi

Part Two: Words



Topics

- how do we learn language?
- what can we learn about a text just from the words?

Quizzes

- Q Chp 5: Words

Homework

- Portfolio Chp 5: Word Guess Game

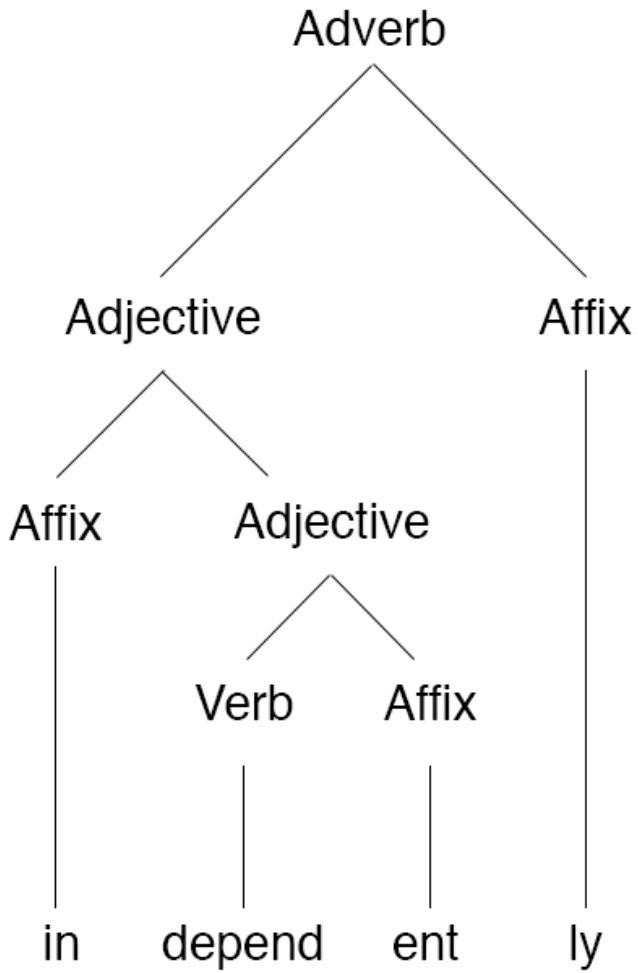
Language development time-table

Stage	Age	Developmental Language and Communication
1	0–3 months	Reflexive communication
2	3–8 months	Reflexive communication; interest in others
3	8–12 months	Intentional communication; sociability
4	12–18 months	First words
5	18–24 months	Simple sentences of two words
6	2–3 years	Sentences of three or more words
7	3–5 years	Complex sentences; has conversations

- Source: https://courses.lumenlearning.com/edpsy/chapter/language-development/

gives cues to meaning

Word morphology



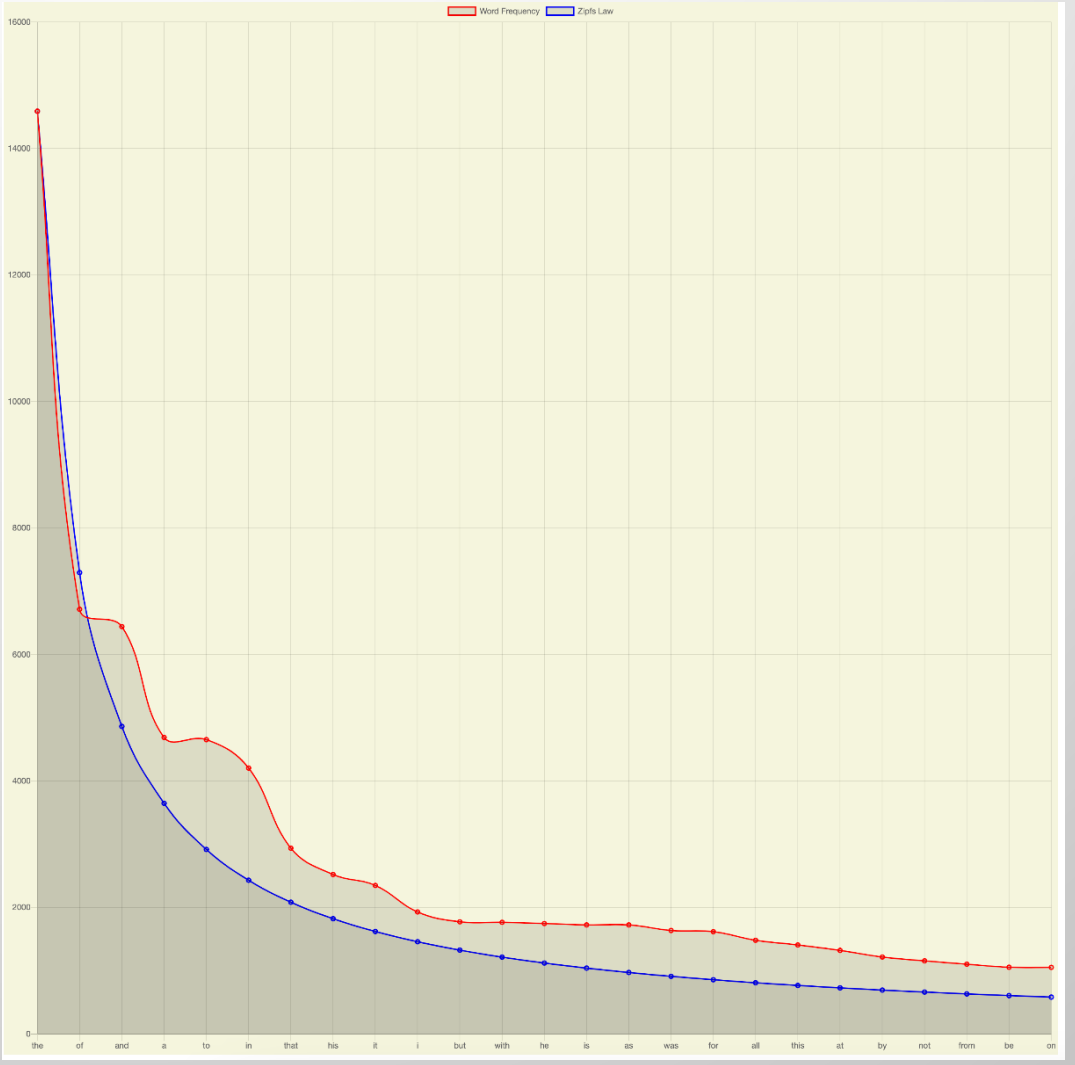
Python Code Examples
In-class coding

- GitHub:
  - 5.1 Words and Counting
  - 5.2 NLTK Text Object and Methods

More about Zipf's law and Heap's law

Experimenting with word frequencies

- Word Count Visualizer: https://coding-blocks-archives.github.io/jQuery_ChartJS_Wordcount_Visualiser/
- Try out some texts from project Gutenberg
- Word frequencies are shown in red, Zipf's law in blue for Moby Dick



the	14590
of	6715
and	6442
a	4689
to	4654
in	4205
that	2938
his	2522
it	2349
i	1930
but	1774
with	1766
he	1747
is	1725
as	1725
was	1637
for	1618
all	1482
this	1408
at	1322
by	1217
not	1157
from	1104
be	1056
on	1054

Zipf's law

- The frequency of a word in a sufficiently large text is inversely proportional to its rank in the frequency table
- If a word has rank N in the frequency table, it's frequency will be proportional to 1/N
- George Zipf, American linguist 1902-1950
- Holds across languages
- Similar phenomenon observed in:
  - Ranks of notes in music
  - Income rankings
  - Cities by population

Zipf's law

- Originally proposed as an empirical (observed) law
- Still no proof for why this holds true, but some theories:
  - Principle of least effort, so that easier/shorter words get used more often
  - Preferential treatment theorem, (ex: the rich get richer), in terms of words, the words that are more commonly out there get used more often
  - Zipf's brevity law, which observes that the small the word, the more frequently it is used

Zipf’s law

- Some text compression algorithms use Zipf’s law to determine which word should be compressed, that is, don’t bother compressing infrequently used words
- Also used in text generation systems to ensure that the generated text follows Zipf’s law

- Results can be different depending on preprocessing: text normalization particularly

Heaps’ law

- Originally credited to Harold Stanley Heaps, but was actually discovered earlier by Gustav Herdan

Heaps' law (or Herdan's law)

- As the length of a document grows, the number of new words encountered slows down
- Also an empirical law

$$V_R(n) = Kn^{\beta}$$

- V is the set of vocabulary words
- K and beta are determined empirically, typically in English:
  - 10 < K < 100
  - 0.4 < beta < 0.6

Heaps' law

- Useful in predicting the size of NLP models based on the training text for language models like GPT-3
- The GPT-3 models has 175 BN parameters on this training corpus:

Datasets	Quantity (Tokens)	Weight in Training Mix	Epochs elapsed when training for 300 BN tokens
Common Crawl (filtered)	410 BN	60%	0.44
WebText2	19 BN	22%	2.90
Books1	12 BN	8%	1.90
Books2	55 BN	8%	0.43
Wikipedia	3 BN	3%	3.40

Heaps' law

- Paper verifying Heaps' law using Ngram data
- https://arxiv.org/pdf/1612.09213.pdf
- Found that the exponent varied significantly with time intervals

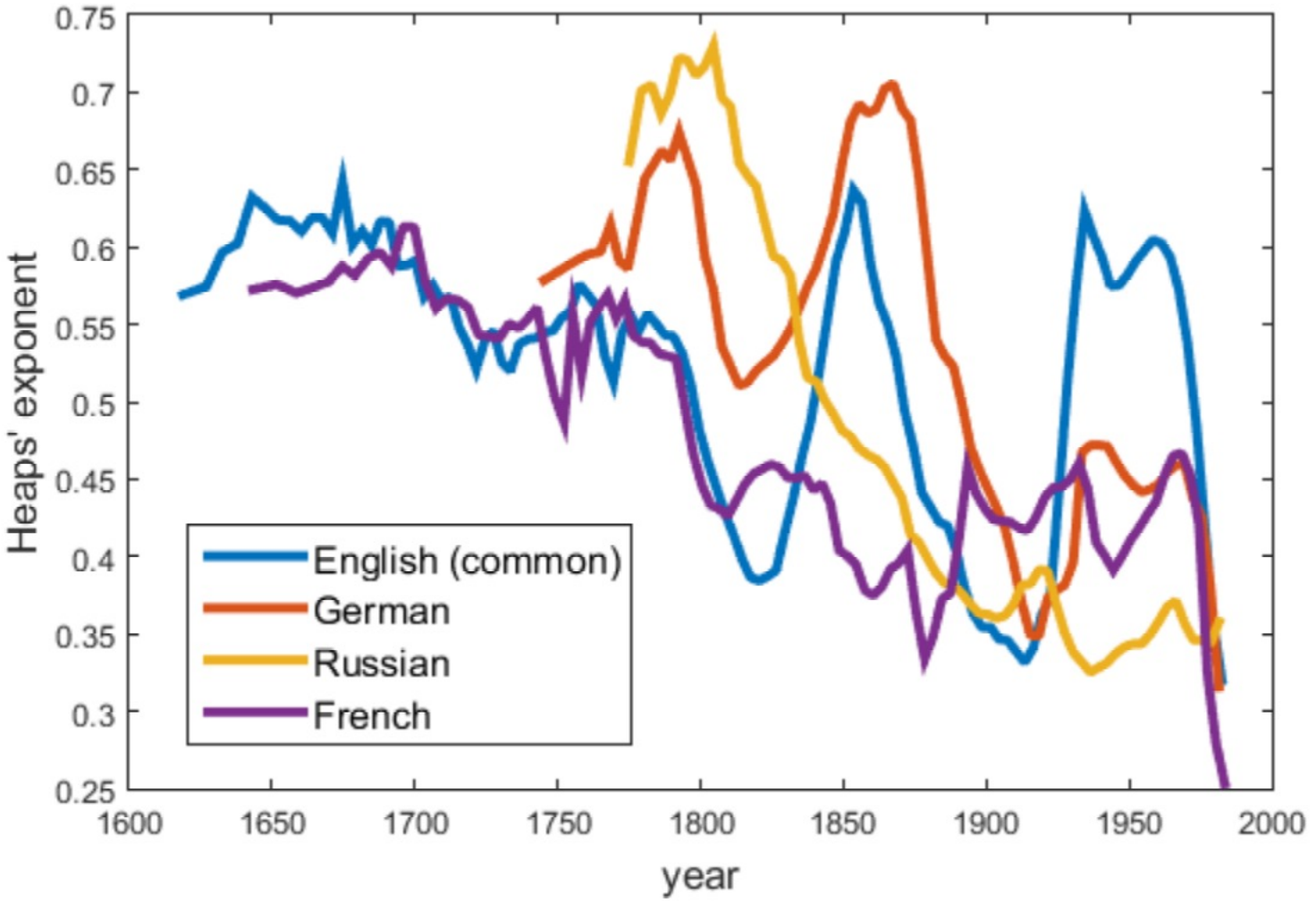


Figure 4. Change of the Heaps exponent with the time for the four European languages

Essential points to note

- How to use NLTK functions for text analysis

To Do

- Quiz Chp 5 Words
- Portfolio Chp 5: Word Guess Game

Next topic

POS Part of Speech Tagging