Text Analysis for Social Sciences in Python

Week 5: Text as Data – Pre-processing Data (cont) & Obtaining Corpora

WINTER SEMESTER 21/22

HUYEN NGUYEN

HTTP://HUYENTTNGUYEN.COM

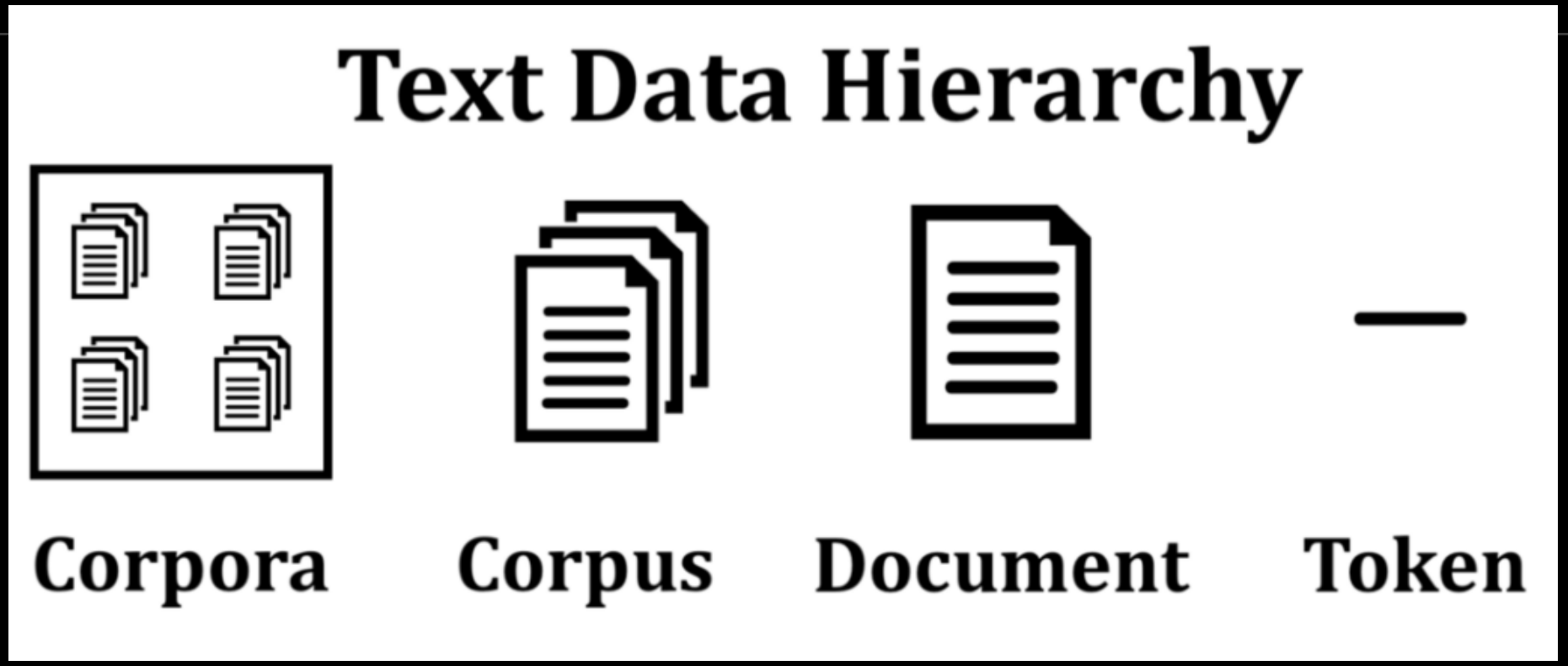
Last week…

Pre-processing data

Basic dimension reduction techniques
(stopwords, stemming, lemmatization)

…any questions with the materials? Home exercises?

Text as Data?



https://medium.com/@raufulazam95218/basics-of-natural-language-processing-nlp-e2b75d2e1dfe

Today's agenda

1. Feature selection/dimension reduction
   - tf-idf
   - N-grams
   - Other features
2. Obtaining corpora
   - Webscraping
   - Existing data sets

1. Dimension reduction: tf - idf

"Term frequency – inverse document frequency" (tf-idf)

$$tf_{ij} \times idf_{ij}$$

tf_{ij} = the count of occurrences of word/feature j in document i.

idf_{ij} = the log of one over the share of documents containing j $\log(\frac{n}{d_j})$

where $d_j = \sum_i \mathbb{I}_{|c_{ij}>0|}$, and n = total number of documents.

→ What happen with very rare/common words after $\boldsymbol{tf_{ij} \times idf_{ij}}$?

Dimension reduction: tf - idf

Very rare words will have low tf–idf scores because of low tf_{ij}.

Very common words that appear in most/all documents will have low tf–idf scores because of low idf_{ij}.

→ This improves on simply excluding words that occur frequently because it will keep words that occur frequently in some documents but do not appear in others

→ Common practice = keep only the words within each document i with $tf_{ij} \times idf_{ij}$

scores above some rank or cutoff.

n-grams/bag-of-words

The simplest and most common way to represent a document.

A phrase of length n is referred to as an n-gram.

The order of words is ignored altogether, where…

c_i = a vector whose length is equal to the number of words in the vocabulary with elements …

c_{ij} = the number of times word j occurs in document i.

This scheme can be extended to encode a limited amount of dependence by counting unique phrases, rather than unique words.

n-grams pitfalls

The dimension of c_i increases exponentially quickly with the order n of the phrases tracked.

The majority of text analyses consider n-grams up to 2 to 3 at most.

Why? The return to richer n-gram modeling is usually small relative to the cost.

<u>Best practice:</u>

1) begin analysis by focusing on single words.
2) Given the accuracy obtained with words alone, evaluate if it is worth the extra time to move on to 2-grams or 3-grams.

Other feature reduction levels?

Sentence/clause/syntax-level parsing?

Ordered sequence of transition between words? A single sentence of length s (i.e., containing s words) = $\mathrm{p} \times s$ matrix S, where…

the nonzero elements of S = indicate occurrence of the row-word in the column-position within the sentence

p = length of the vocabulary

Other feature reduction levels?

(!) Massive ↑↑ in the dimensions of the data to be modeled.

$\Rightarrow$ Analysis via word embedding = mapping of words to a location in $\mathbb{R}^K$ for some $K \ll \mathrm{p} \Rightarrow$ sentences = sequences of points in K dimensional space.

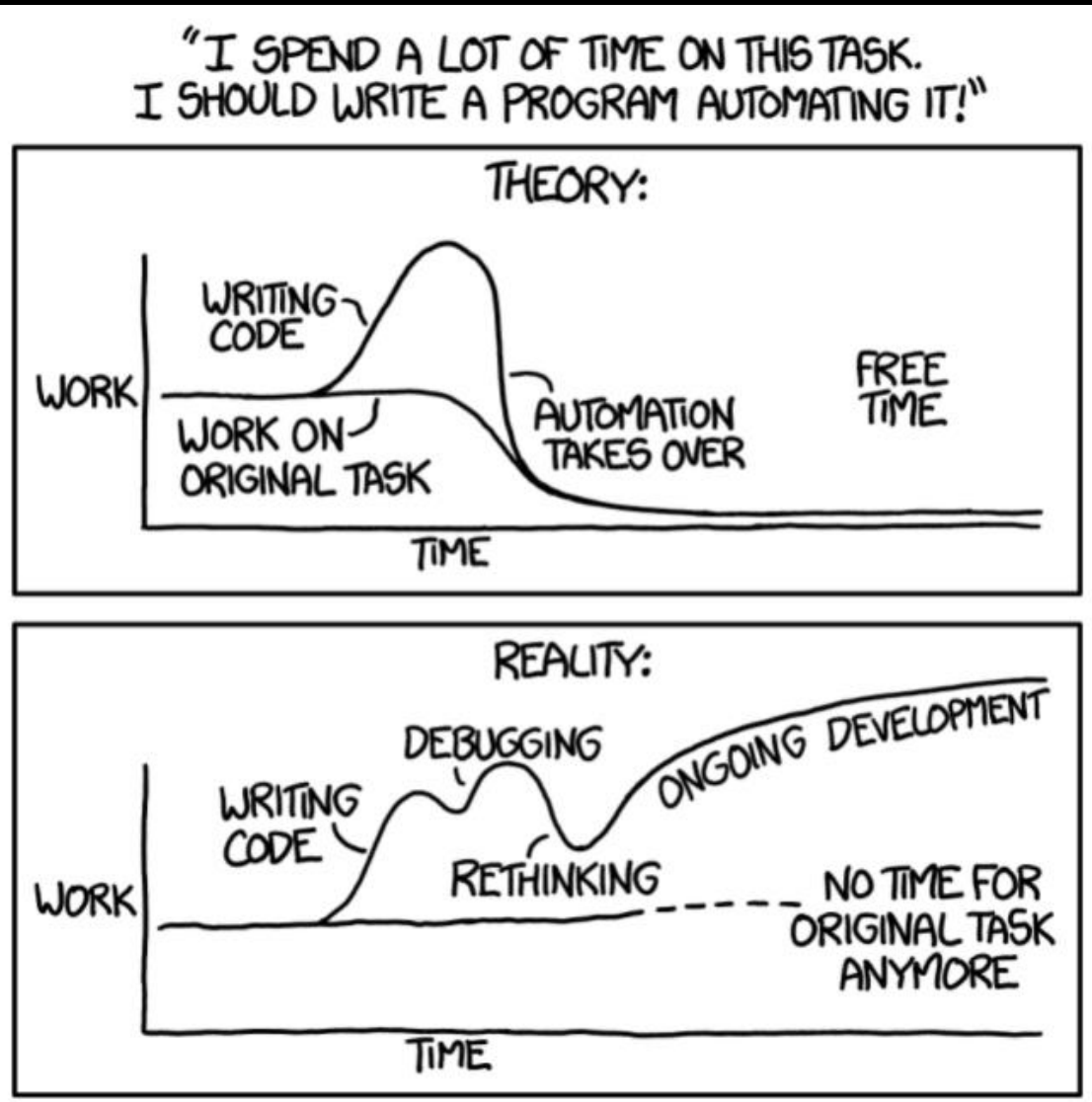
Data-driven dimension reduction techniques

IDEA: Form a small number of linear combinations of predictors → use these derived indices as variables in an otherwise standard predictive regression.

Principal Component Analysis (PCA)

Partial Least Squares (PLS)

Data Collection: Web scraping?



HTML elements of websites?

Useful web scraping packages?

Practice time ☺

Open your Google Colab/ Jupyter Notebook

Practice time ☺

https://github.com/httn21uhh/Text-Analysis-for-Social-Sciences-in-Python

There are .ipynb files for your reference

- Download it
- Run it on your own laptop
- This is NOT meant for passive scrolling!

This week…

Open your Google Colab/ Jupyter Notebook. Two exercises today:

Pre-processing data: tf-idf, n-grams

Webscraping with Selenium

- Follow closely the illustrated examples and replicate yourself as the session proceeds.
- Raise your hands to ask questions at any point, including when you think things go too fast/slow for you.