

# Text Analysis and Visualization with Python<sup>1</sup>

Arieda Muço

Central European University

October 22, 2020

---

<sup>1</sup> Based on slides from Molly Roberts and Bijoyan Das.

# Text as Data

- Classify an email message as either a legitimate email or spam
- Learn about the opinion of a politician on the topic of immigration
- The content of the text will certainly contain important information for the task
- Text data is usually represented as concatenation of characters. In any of the examples just given, the length of the text data will vary
- This feature is clearly very different from the numeric features, and we will need to process the data before we can apply algorithms to it

# Preprocessing

- Simplify and make it useful for our purposes
- Lower dimensionality

# Document -Term

$$X = \begin{pmatrix} 1 & 0 & 0 & \dots & 3 \\ 0 & 2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 5 \end{pmatrix}$$

$X = N \times K$  matrix

- $N$  = Number of documents
- $K$  = Number of features

# Preprocessing for Quantitative Text Analysis

Recipe for preprocessing: retain useful information

- Remove capitalization, punctuation
- Discard stop words
- Discard Word Order (Bag of Words Assumption)
- Create Equivalence Class: Stem, Lemmatize, or synonym
- Discard less useful features (depends on application)
- Other reduction, specialization

Output: Count vector, each element counts occurrence of stems

# Stop Words

Stop Words: English Language place holding words

- the, it, if, a, able, at, be, because...
- Add “noise” to documents (without conveying much information)
- Discard stop words: focus on substantive words
- **Caution:** Exercise caution when discarding stop words. You may need to customize your stop word list.

# Creating an Equivalence Class of Words

Reduce dimensionality further (create equivalence class between words)

- Words used to refer to same basic concept.
  - ▶ family, families, familial → famili
- Stemming/Lemmatizing algorithms: Many-to-one mapping from words to stem/lemma

# Stemming vs Lemmatization

- Stemming algorithm:
  - ▶ Consists of chopping off end of word
  - ▶ Porter stemmer, Lancaster stemmer, Snowball stemmer
- Lemmatizing algorithm:
  - ▶ Condition on part of speech (noun, verb, etc)
  - ▶ Verify result is a word



# Stemming vs Lemmatization

- Stemming algorithm:
  - ▶ Word representations may not have any meaning
  - ▶ Takes less time
  - ▶ Use stemming when meaning of words is not important for analysis. Example: Spam detection.
- Lemmatizing algorithm:
  - ▶ Word representations have meaning
  - ▶ Takes more time than Stemming
  - ▶ Use lemmatization when meaning of words is important for analysis. Example: question answering application.

## Additional read

Stemming and Lemmatization – Stanford NLP

<https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html>

Preprocessing reduces dimensionality where it causes problems for inference (stopwords, stemming) and sometimes increases dimensionality when it makes our inferences better (bigram, ngrams)

# A Complete Example

“Political power grows out of the barrel of a gun” - Mao

# A Complete Example

“Political power grows out of the **barrel of a gun**” - Mao

- **ngram**: An analyst may want to combine words into a single term that can be analyzed.

# A Complete Example

[Political], [power], [grows], [out], [of], [the], [barrel of a gun] -  
Mao

- **ngram:** An analyst may want to combine words into a single term that can be analyzed.

# A Complete Example

[Political], [power], [grows], [out], [of], [the], [barrel of a gun] -  
Mao

- **Remove Stopwords:** Removing terms that do not convey important information

# A Complete Example

[Political], [power], [grows], [out], [barrel of a gun] - Mao

- **Stemming:** Takes the ends of conjugated verbs or plural nouns, leaving just the stem.



# A Complete Example

Finally, we can turn tokens and documents into a “document-term matrix.”

- Imagine we have a second document in addition to the Mao quote, which tokenizes as follows

Document 1: [polit], [power], [grow], [out], [barrel of a gun]

Document 2: [compar], [polit], [chicago], [polit]

# Document-Term-Matrix

	<i>Doc1</i>	<i>Doc2</i>
<i>power</i>	1	0
<i>grow</i>	1	0
<i>out</i>	1	0
<i>barrel of a gun</i>	1	0
<i>compar</i>	0	1
<i>polit</i>	1	2
<i>chicago</i>	0	1

# All steps together

1. Remove capitalization and punctuation
2. Discard word order (Bag of Words)
3. Remove stop words
4. Applying Stemming Algorithm
5. Create count vector

# How Could This Possibly Work?

- Speech may contain sarcasm:
  - ▶ The Star Wars prequels were amazing because everyone loves a good discussion about trade policy
- Subtle Negation
  - ▶ They have not succeeded, and will never succeed, in breaking the will of this valiant people
- Order Dependence
  - ▶ Peace, no more war
  - ▶ War, no more peace

# How Could This Possibly Work?

1. It might not: Validation is critical (task specific)
2. Central Tendency in Text: Words often imply what a text is about war, civil, union or tone consecrate, dead, died, lives. Likely to be used repeatedly: create a theme for an article
3. Human supervision: Inject human judgement (coders): helps methods identify subtle relationships between words and outcomes of interest

It is easier to capture some things than others

Time to code!!!