Day 2 – Text to Bag of Words data

Introduction to Text Analysis in Python
TRIADS Training Series

Ishita Gopal

February 10, 2024

Last Workshop:

- Overview of text analysis pipelines
- Examples of what *dictionaries, supervised and unsupervised models* can do and what their advantages/disadvantages are
- Learnt to manipulate the basic unit of text – “strings” - in Python.

Today

1. Understand the document term matrix (DTM)
2. Get familiar with CountVectorizer in Python.
3. Learn to access and manipulate elements of a DTM.
4. Do word usage analysis using simple operations on DTMs.

Corpus, Tokens & Vocabulary

Corpus: a digitized collection of text

Tokenization: is the division of text into **tokens**. This may mean "words" split on spaces and punctuation. But…

- we may not wish to throw out punctuation (e.g.,"!!!"or "$" or #blessed)
- not split on punctuation (e.g., D.C. or gopali@wustl.edu)
- or not split on spaces (District of Columbia or Supreme Court)
- Some languages don't have spaces.

Chinese: 我开始写小说 = 我 开始 写 小说
I start(ed) writing novel(s)

姚明进入总决赛 “Yao Ming reaches the finals”

3 words?
姚明 进入 总决赛
YaoMing reaches finals

5 words?
姚 明 进入 总 决赛
Yao Ming reaches overall finals

7 characters? (don't use words at all):
姚 明 进 入 总 决 赛
Yao Ming enter enter overall decision game

Source: Burt L. Monroe, TADA slides

- **Tokens** and **types** are different.

- **Types** are the unique tokens - constitute the **vocabulary**, V.

Text normalization ("Pre-processing")

The process of putting tokens / textual features in a standard form. For example:

Case folding: typically converting all upper-case characters to lower case.

- **This** and **this** become the same thing.
- But..it may overdo it, conflating for example **Trump** and **trump** or **US** and **us**

Another text normalization thing...

Reducing words to their root form to handle variations.

Stemming involves removing prefixes or suffixes from words

- Probably okay to stem **running** and **runs** becomes **run**
- But...Porter Stemmer algorithm stems **universal**, **university**, and **universe** all to **univers**.

Lemmatization could be better but is computationally expensive. Reduces words to their base or dictionary form. Takes into consideration the larger context surrounding the word.

Part-of-speech tagging:

- Assigning a part-of-speech label to a given token.

- These are word-level classification problems, typically referred to as **tagging** or **sequence labeling** or **annotation** tasks.

- Word-level tasks like part-of-speech tagging interact with sentence-level tasks for identifying grammatical / syntactical structure that ties words together, which are typically referred to as **parsing**.

Document Term Matrices

Workflow:

Step 1: Take the entire Corpus

Workflow:

Step 1: Take our entire Corpus

Step 2: Identify all the all the tokens (eg: words) in the corpus

Workflow:

Step 1: Take the entire Corpus

Step 2: Identify all the tokens (eg: words) in the corpus

Step 3: Do some preprocessing (lower case, remove stop words etc.)

Workflow:

Step 1: Take the entire Corpus

Step 2: Identify all the tokens (eg: words) in the corpus

Step 3: Do some preprocessing (lower case, remove stop words)

Step 4: Transform it into a mathematical matrix

- **each column** *represents a unique word that exists in the entire Corpus across all text documents*
- **each row** *represents a unique text document*

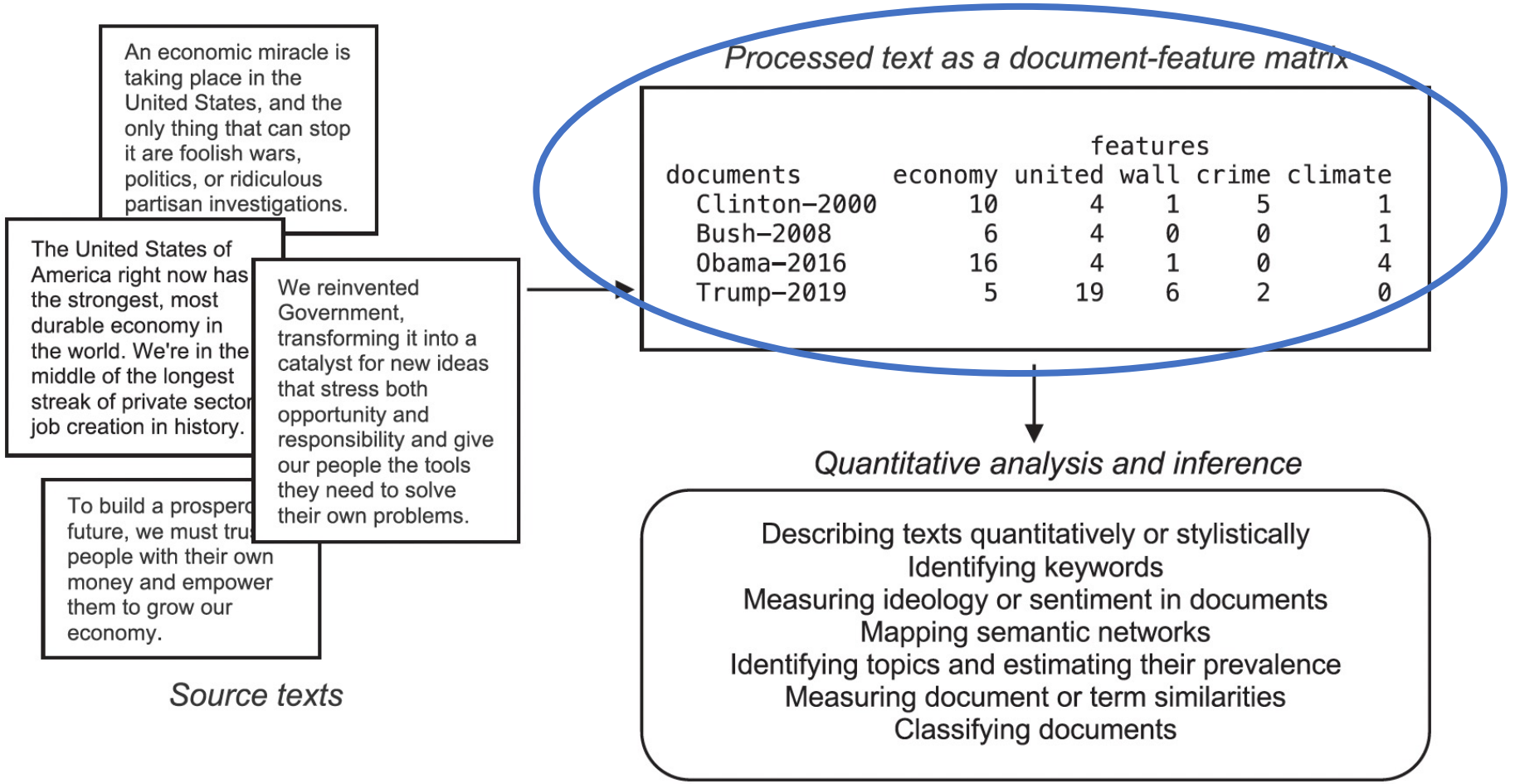

Figure 26.1 From text to data to data analysis

Source: Benoit. 2020

- Python Lab Notebook:

https://colab.research.google.com/drive/1Ku1NDJ5dlnyeB_xcIaPPz8Wl4r-gp_7E?usp=sharing