Computational Analysis of Classical Texts

Marc E. Canby

Lati 318 — Cicero: *De Re Publica*
Rice University

Outline

1. Getting and Cleaning the Data

2. Exploratory Text Analysis

3. Keyword Extraction: Frequency Count and TextRank

4. Predicting Missing Text: LSTM Neural Networks

Outline

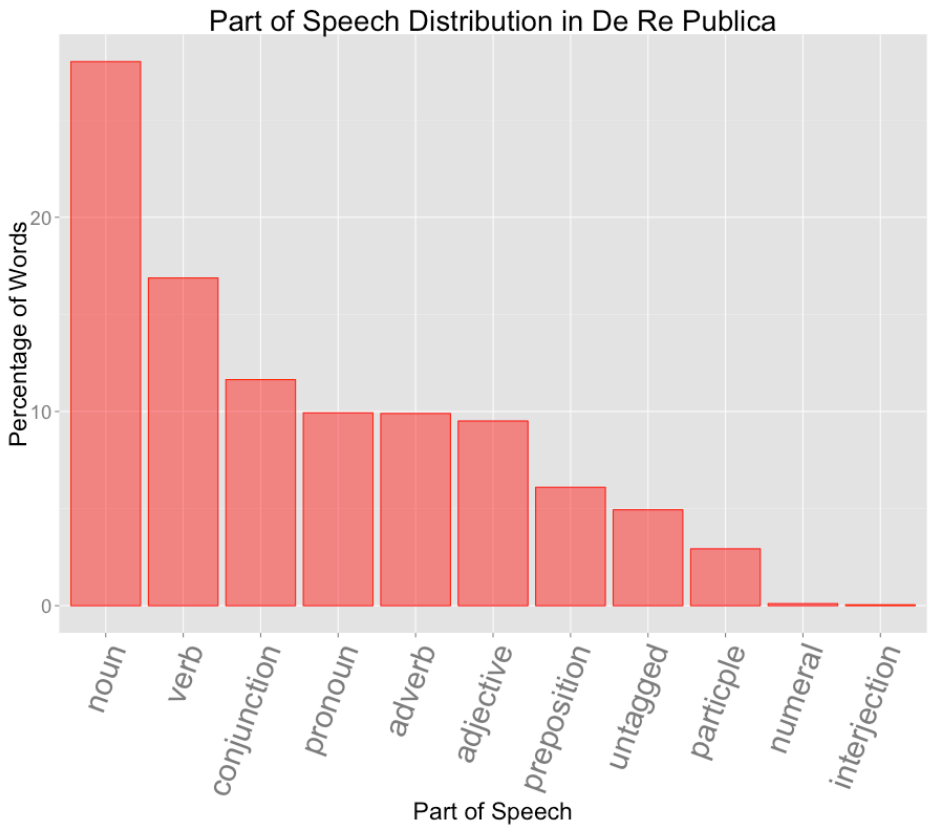
Getting and Cleaning the Data

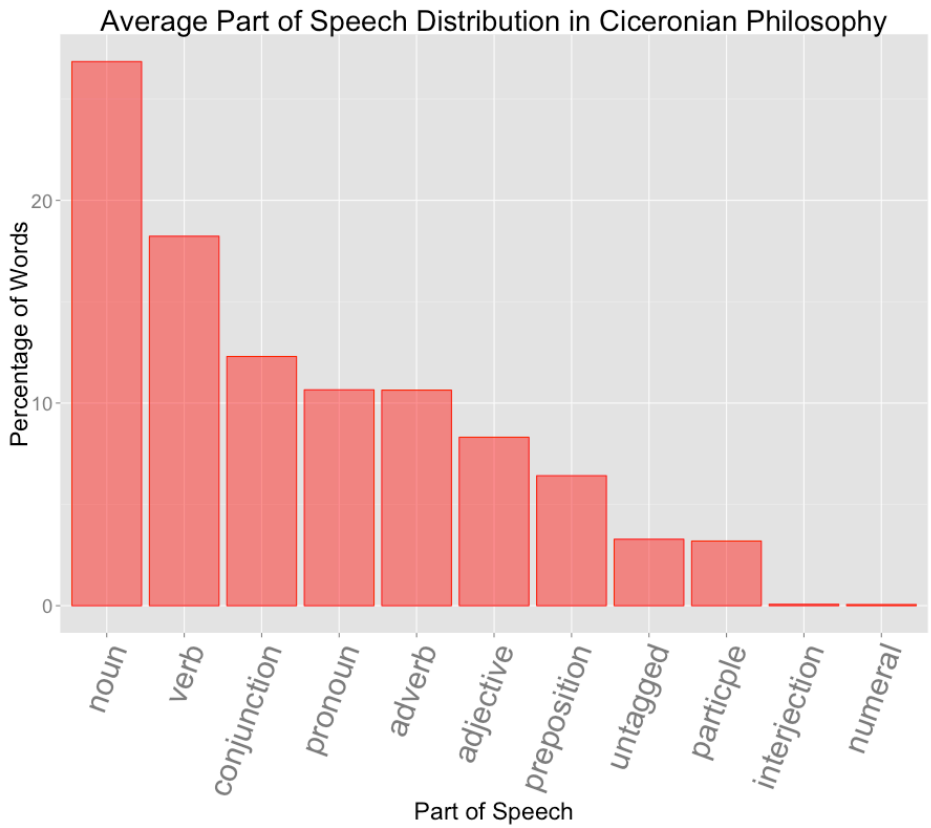
- Text obtained from *The Latin Library* at the sentence level:
  - `['nempe', 'ab', 'iis', 'qui', 'haec', 'disciplinis', 'informata', 'alia', 'moribus', 'confirmarunt', ',', 'sanxerunt', 'autem', 'alia', 'legibus', '.']`
- Cleaned up messy elements of data:
  - Line numbers: `[1,2,...,71]`
  - Angle brackets: `['&', 'lt', ';', 'im&gt', ';', 'petu', 'liberavissent', ',', 'nec',...]`
    - `<` should be < `>` should be >
  - Hyphens encoded as `&#`
  - English words: `['Cicero', 'The', 'Latin', 'Library', 'The', 'Classics', 'Page']`

Outline

Exploratory Text Analysis

- Number of characters: 109,777 (*average:* 136,893)
- Number of words: 20,067 (*average:* 24,924)
- Number of sentences: 820 (*average:* 1,059)





Outline

Keyword Extraction: Frequency Count

- Goal: Determine a set of keywords that summarizes the text
- Naive approach: Take words in text with highest frequency:
- Problem: Does not account for structure of text and relationships between words

Outline