UPPSALA
UNIVERSITET

Text analysis I: A conceptual background

Introduction to R for Social Sciences

Josef Ginnerskov
Department of Sociology
2022-08-12

Today's outline

1. **A conceptual background to (computational) text analysis in R**

2. Basic text analysis tasks performed on vector strings following the logic of the tm package

3. More advance text analysis tasks conducted on digitized books based on the tidytext package

4. Individually solving the problem set by building your own Gutenberg corpus

Computational text analysis?

1. New expression for quantitative text analysis (see also text mining)

2. Draws heavily on NLP (natural language processing), which is increasingly drawing on machine learning (a branch of AI)

3. Social scientists leveraging on tools generated in computer science, data science and linguistics (see also computational social science)

4. Most common tasks are data exploration and data classification (e.g. what topics are 2000 sociology abstracts writing about based on their most frequent words or are these 362 IMBD reviews of "Sex and the City 2" mostly negative or positive based on their sentiments)

Call a spade a spade - or a shovel (1/3)

Documents

1. Book, article, post, message ... = **document**
2. Document + document = **corpus** (or data set)
3. Corpus + corpus = **corpora** (or data sets)

Call a spade a spade - or a shovel (2/3)

Words

1. Combinations of letters = **words** or **terms** or **tokens** or **features** or **n-grams**

2. N-grams include **unigrams** (e.g. horse), **bigrams** (e.g. cultural capital) and **trigrams** (e.g. social media trolls)

3. Common "set of words" types: **vocabulary** (i.e. all words occurring in your corpus), **dictionary** (e.g. a set of words of interest for your analysis), **stop words** (insignificant words to exclude from the data), and **lexicons** ([pre-made] word lists for categorizing your vocabulary needed for particular tasks)

Call a spade a spade - or a shovel (3/3)

Representations

1a. The **bag-of-words model** or **one-hot vectors** (words are stripped of their placement in sentences so that each document is reduced to a "bag" containing only word frequencies)

1b. The **document-term matrix** aka **document-feature matrix** (each row represents a document and each column a word) or the **tidy text tibble** (each row has one column for the word and another for each document containing the word)

1c. **Term frequency-inverse document frequency** (a weighted matrix that intends to show how important words are to a document in relation to the other documents in a corpus)

1d. **Topic modeling** (an unsupervised machine learning technique to explore what topics are most relevant in a corpus; based on the idea that distinctly co-occurring words make up a documents are)

2a. **Word embeddings** (include statistics on each words' neighbors in the text, i.e. the words are embedded in the text)

Turning text into data (1/2)

Access

1. Data from traditional methods (open answers in questionnaires, interview scripts, field notes ...)
2. Digitize physical texts (scan → OCR → encode → load into R)
3. Download from digital archives (**Gutenberg**, Runeberg, Scopus, Web of science ...)
4. Scraping websites (e.g. social media platforms like Twitter and Reddit)

Turning text into data (2/2)

Preprocess

Generate data set

1. Decide what documents make up your corpus and load the data
2. Add metadata (variables like author, publication year ...)

Tokenization

1. Remove what is not a word (remove: upper case, numbers, punctiation, white space...)
2. Decide what is a word (stemming, lemmatization, removing stop words)
3. Categorize words (Part-of-speech (POS) tagging, sentiment analysis...)

Some text analysis task solvable with R

1. Global and local word occurrence counts (e.g. wordcloud)
2. Words specific for each text in relation to the corpus (e.g. tf-idf)
3. Relations between words (e.g. cluster analysis or co-occurrence network analysis)
4. Themes associated with each text (e.g. topic modeling)
5. Emotions in texts (sentiment analysis)
6. The "meaning" of words (word embeddings)
7. The style of authors (stylometry)

Some popular R packages for text analysis

- **tm** (general text mining, corpus-based)
- **tidytext** (R specific leveraging the tidy format)
- **quanteda** (all-encompassing quantitative text analysis)
- **koRpus** (preprocessing and general statistics for single texts)
- **udpipe** and **opennlp** (natural language processing toolkit)
- **text2vec** (advance machine learning tasks like word embeddedings)
- **topicmodels**, **stm** and **lsa** (packages for finding latent topics)
- **wordcloud** (...wordclouds)
- **stylo** (stylometric tasks like author attribution)

Thank you for your time!

Do not hesitate to contact me


josef.ginnerskov@soc.uu.se


@doeparen


@doeparen