Text Analysis Glossary

Katie Ireland Kuiper,

PhD Candidate

katherine.kuiper25@uga.edu

This document provides a full glossary of all things text analysis.

Anaconda: python data science platform. Useful for hosting Jupyter notebooks and other environments for testing and running your code.

Artificial Intelligence/AI: interdisciplinary subfield of computer science; focuses on training computers and computational systems to perform a variety of tasks including decision-making, speech recognition, and translation tasks. Overlaps with some areas of Machine Learning and Natural Language Processing.

Brown corpus: first text corpus of general American English, compiled Henry Kučera and W. Nelson Francis at Brown University, in Rhode Island. It is one million tokens and contains a representative sample of different genres of American English. 1961. It is also available for analysis through the UGA Corpus Server!

collocations: a collocation is a group of words or tokens that co-occur. The node is the center word and collocations can involve a span on either the right or left (or both) of the node word. Different statistical measures are used to define the strength of the collocations, like the Mutual Information (MI) score and log-likelihood. See Evert 2010 and Gablasova et al. 2017.

concordance: a method in corpus linguistics and text analysis; allows the researcher to look more closely line by line at the output of a specific word or multi-word corpus search; the individual concordance line provides an example of a specific instance of use. See also KWIC/keyword in context.

corpus, corpora (plural): computerized database of text or multiple texts. corpus linguistics, corpus-based analysis

orcid.org/0000-0002-4725-9525

CQP (Corpus Query Processor): Corpus Query Processor/CQP provides excellent options for fast corpus searching, including regex, and interactive sessions in the terminal window. Available here at UGA with many, many awesome corpora!

distant reading: often applies to computational literary analysis; more generally refers to computational, computer-based methods applied to corpora or databases of text.

document/text classification: also sometimes called text categorization; generally refers to process of grouping texts or documents based on a specific reason or predefined terms. Can also refer to tagging texts or labeling groups of texts according to predefined terms. Example: classifying social media posts, classifying texts by polarity or sentiment.

DTM/document-term matrix: a mathematical matrix object that describes the frequency of terms in a document or collection or documents. It stores this information in an efficient manner so that it is easy for the researcher to run different types of analyses on the DTM data. It is also possible to transpose this into a term-document matrix, which stores the same data information.

formulaic language: refers broadly to different levels or units of text analysis, including collocations, ngrams, multiword units, or phrases. See collocations, ngrams, and Gablasova et al. 2017 for more.

frequency analysis: method from corpus linguistics in which frequencies of different word tokens, collocations, phrases, sentences, etc. can be counted and compared. A frequency distribution contains all tokens, generally sorted in order of occurrences.

HathiTrust: the HathiTrust Digital Library is a wonderful resource for data and contains digitized content from Google books as well as research libraries and the Internet Archive.

Jupyter notebook: open-source web app for running and sharing python code. Includes many useful options for sharing, commenting, organizing, and debugging your code.

Keyword analysis: method from corpus linguistics, specifically refers to comparing word token frequencies in a target corpus with a reference corpus and is defined statistically using the keyness measure.

KWIC (keyword in context): Most generally used format for concordance lines and word frequency searches. Most corpus programs include this as an option for researchers to view results of the searches. see also concordance.

Lemma, lemmatization: a lemma refers to the canonical form of a word (example: swimming, swam, and swim all have the same lemma swim); lemmatization refers to the process of extracting lemmas from texts, a useful processing step for working with text data. Lemmatization also takes into account the word meanings, but stemming only takes into consideration word forms.

Machine Learning: application of artificial intelligence to text analysis; often utilizing statistical algorithms. Different types include supervised and unsupervised machine learning. Supervised refers to the algorithms and programming involved in

orcid.org/0000-0002-4725-9525

getting an expected output, while unsupervised does not have a known outcome or output. As applied to text analysis supervised machine learning examples include: part of speech tagging, organizing, reformatting, and compiling text data. Unsupervised machine learning utilizes algorithms to extract meaning from text data. See also AI.

Montreal French Project: First transcribed corpus of spoken language was created in 1971, 1 million words (Sankoff & Sankoff 1973).

natural language processing (NLP): natural language processing is an interdisciplinary field used in computer science, data science, linguistics, and others to analyze, categorize, and work with computerized text.

n-grams: generally refers to sequences of tokens or words; in computer science and computational linguistics ngrams have different probability applications. In corpus linguistics, refer to n-number of words in a phrase, ie trigrams include three tokens and bigrams include two. See also formulaic language.

Part of speech tagging (PoS): Identifying and tagging each word for part of speech e.g. noun, verb, etc., the Penn Treebank is a tagset used for English data. See also lemmatizing/lemmas.

Perl: general purpose programming language; often used for text analysis. Supports very powerful regular expressions.

Project Gutenberg: excellent resource for digitized content, library containing over 60,000 ebooks that are out of copyright in the US. R package GutenbergR provides easy method to extract and analyze these texts!

Python: very useful programming language; recommended libraries to check out for text analysis include:

1. spaCy: pos tagging, tokenization, dependency parsing, etc.
2. CoreNLP: lemmatization, pos tagging, tokenization, named entity recognition
3. NLTK: Natural Language ToolKit; contains over 50 corpora, includes options for tokenization, tagging, parsing, document classification
4. Gensim: useful for various types of topic modeling
5. PyNLPI: open-source NLP library; great for of tasks ranging from building simplistic models and extraction of n-grams and frequency lists, with support for complex data types and algorithms
6. Pattern: useful for web-crawling (webscraping) for creating your own corpora; includes options for tokenizing, pos tagging, etc
7. Polyglot: very useful library for other languages than English
8. TextBlob: includes options for pos-tagging, noun phrase extraction, classification, translation and sentiment analysis

Regex/Regular expression: string or sequence of characters utilized to search for specific patterns; very useful and widely used in text analysis and often different programming languages will have different sequences for supporting Regex.

R: incredibly useful programming language; originally developed for statistical analyses. Recommended packages for text analysis include:

orcid.org/0000-0002-4725-9525

1. Tidytext: helpful for data formatting and visualization; works well with other packages in the Tidyverse (Silge & Robinson 2016)
2. Textmining/tm: includes options for data processing, metadata management, and creation of term-document matrices (Feinerer 2020; Feinerer et al. 2008)
3. Syuzhet: package created specifically for sentiment analysis and for extracting sentiment-derived plot arcs by Jockers.
4. **Text2vec**: useful for creating dtm, vectorizing data; it also supports topic modeling and collocational analysis.
5. StringR**:** supports regex, pattern matching, and useful options for string manipulation
6. spacyR**:** NLP package originally created for Python; useful for lemmatization, tokenization, and works well with quanteda and tidytext pacakges.
7. Quanteda**:** incredibly useful package; includes preprocessing abilities, dtm function, as well as statistical analyses options like document classification and topic modeling.
8. **Ggplot2:** great way to visualize your data; part of the Tidyverse collection of R packages.
9. Udpipe**:** useful for tokenization, part of speech tagging, etc. in a variety of languages including Arabic, Greek, German, Spanish, French, Dutch, English, and many, many more!
10. **polmineR:** excellent package for corpus analysis; works in conjunction with CQP and includes options for kwic concordance lines, collocational and ngram analysis, and dispersion. Works with many different languages, too.
11. **janeaustenr:** excellent R package that plugs-in well with tidytext for analyzing Jane Austen's complete works!
12. **Gutenbergr:** fantastic package for obtaining and utilizing texts from Project Gutenberg (see above) for corpus and text analysis.

R Studio: development environment for the R programming language. Most software demonstrations in R on the Digi website use R Studio to write, execute, and test the code.

Semantic analysis: refers to different types of methods depending on subfield; in corpus linguistics this often involves use of a tagset that tags words based on meaning, two prominent examples are the URCEL tagset developed by Rayson et al. and the Historical Thesaurus Semantic Tagger (Alexander et al. 2015) developed at the University of Glasgow.

Sentiment analysis: Natural language processing (NLP) technique; used to derive polarity, sentiment, and/or subjective opinions from text. Generally obtained with utilization of a sentiment dictionary.

string: specific type of data to represent text and is often implemented as an array of bytes (or words) that stores a sequence of elements, typically characters, using some character encoding (like Latin1 or UTF8). Strings may also be present in other data types such as lists, depending on the programming language.

term-document matrix (TDM): see DTM.

Tf-idf: term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is in a corpus or group of corpora.

topic modeling: Machine Learning technique that uses statistical modeling to output features (text tokens) in different

orcid.org/0000-0002-4725-9525

topic groups. It enables probabilistic modeling of term frequency occurrences in documents and is used to estimate the similarity between documents and variables (topics). Interpretation is still necessary of the statistical output and resulting topic groups; specific methods within this include Latent Dirichlet Allocation (LDA) and others.

tokenization: refers to the process of splitting up text into separate tokens/words, sentences, paragraphs, etc. These settings will most likely be different or contain different options depending on the program you use to tokenize your strings/text data.

Zipf's law: named after George Zipf, refers to the fact that in any given text collection, the frequency of a word is inversely proportional to its rank.

Works Cited

Alexander, M., Baron, A., Dallachy, F., Piao, S., Rayson, P., and Wattam, S. 2015. The Historical Thesaurus Semantic Tagger. http://eprints.gla.ac.uk/115024/

Al-Rfou, Rami. 2015. Polyglot. https://polyglot.readthedocs.io/en/latest/

Benoit K, Matsuo A, European Research Council. 2020. SpacyR package. https://cran.r-project.org/web/packages/spacyr/spacyr.pdf

Benoit K, Watanabe K, Wang H, Nulty P, Obeng A, Müller S, Matsuo A. 2018. quanteda: An R package for the quantitative analysis of textual data. https://quanteda.io. Journal of Open Source Software, 3(30), 774. doi: 10.21105/joss.00774

Bing, Liu. 2015. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Baker, Paul. 2011.

Bird, Steven, Ewan Klein, and Edward Loper. 2019. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit

Blaette, Andreas. 2020. Introducing the 'polmineR'-package. https://cran.r-project.org/web/packages/polmineR/vignettes/vignette.html.

Brezina, Vaclav. 2018. Statistics in Corpus Linguistics.

Brown, Simon. 2016. Tips for Computational Text Analysis. https://matrix.berkeley.edu/research/tips-computational-text-analysis

Bussiere, Kirsten. 2018. Digital Humanities - A Primer.

Clarke, M. 2018. An Introduction to Text Analysis and Processing with R. https://m-clark.github.io/text-analysis-with-R/

Evert, Stefan and Andrew Hardie. 2011. Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium. In Proceedings of the Corpus Linguistics 2011 conference, University of Birmingham, UK.

Evert, Stefan. 2010. Association Measures. http://www.collocations.de/AM/index.html

Evert, Stefan. 2003. The CQP Query Language Tutorial. Evert, Stefan. 2007. Corpora and collocations. http://www.stefan-evert.de/PUB/Evert2007HSK_extended_manuscript.pdf Feinerer et al. 2008.

Feinerer, Ingo. 2020. Introduction to the tm Package: Text Mining in R. https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf

orcid.org/0000-0002-4725-9525

Feinerer, Ingo, Hornik, Kurt, Meyer, David. 2008. Text Mining Infrastructre in R. *Journal of Statistical Software.*

Firth, JR. 1957. Papers in Linguistics. London: OUP.

Gablasova, Dana, Brezina, Vaclav, and McEnery, Tony. 2017. Collocations in Corpus-Based Language Learning Research: Identifying, Comparing, and Interpreting the Evidence. https://onlinelibrary.wiley.com/doi/10.1111/lang.12225

Grün, Bettina & Kurt Hornik. topicmodels: An R Package for Fitting Topic Models. https://cran.rproject.org/web/packages/topicmodels/vignettes/topicmodels.pdf

Han, Na-Rae. Python 3 tutorials. http://www.pitt.edu/~naraehan/python3/.

HathiTrust. https://www.hathitrust.org/about.

Honnibal, Matthew and Montani, Ines and Van Landeghem, Sofie and Boyd, Adriane. 2020. spaCy: Industrial-strength Natural Language Processing in Python. https://doi.org/10.5281/zenodo.1212303

Hvitfeldt, Emil & Julia Sigle. 2020. textdata: download and load various text datasets.

Jockers, M. 2015. Syuzhet: Extract sentiment and plot arcs from text. https://github.com/mjockers/syuzhet

Jockers, Matthew. 2020. Introduction to the Syuzhet Package. https://cran.r-project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html

Kretzschmar, William, C. Darwin, C. Brown, D. Rubin, D. Biber. Looking for the Smoking Gun: Principled Sampling in Creating the Tobacco Industry Documents Corpus. Journal of English Linguistics. 32:1.

Laudun, John. Text Analytics 101. https://johnlaudun.org/20130221-text-analytics-101/

Loria, Steven. 2020. TextBlob: Simplified Text Processing. https://textblob.readthedocs.io/en/dev/

Mullen, Lincoln. 2018. Introduction to the tokenizers package.

Mullen, Lincoln, Keyes, Os, Selivanoc, Dmitriy, Arnold, Jeffrey, Kenneth, Benoit. 2018. tokenizers R package.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal Dependency Parsing from Scratch In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 160-170.

Project Gutenberg. https://www.gutenberg.org

Rayson, Paul, Archer, Dawn, Piao, Scott, and McEnery, Tony. The UCREL Semantic Analysis System. University of Lancaster.

Rehurek, Radim, and Sojka, Peter. 2010. Software Framework for Topic Modelling with Large Corpora. Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.

Roberts, Margaret, Brandon Stewart, and Dustin Tingley. Stm: R Package for Structural Topic Models.

Robinson, David. 2020. gutenbergr: Download and Process Public Domain Works from Project Gutenberg. https://CRAN.R-project.org/package=gutenbergr

orcid.org/0000-0002-4725-9525

Selivanov, Dmitriy, Bickel, M., Wang, Q. 2020. Text2vec: Modern Text Mining Framework for R. https://CRAN.R-project.org/package=text2vec

Silge, Julia, and David Robinson. 2016. tidytext R package.

Silge, Julia, and David Robinson. 2020. Text Mining with R: A Tidy Approach.

Wickam, Hadley. 2019. stringr: Simple, Consistent Wrappers for Common String Operations. https://CRAN.R-project.org/package=stringr

Wijffels, Jan. 2020. UDPipe Natural Language Processing – Text Annotation. https://cran.r-project.org/web/packages/udpipe/vignettes/udpipe-annotation.html

Last updated 26 March 2021, by Katie Ireland Kuiper.

orcid.org/0000-0002-4725-9525