**Useful recent references on text mining (in alphabetical order)**

Aggarawal CC: Machine Learning for Text. Springer 2018

The emphasis is on machine learning methods for textual analysis. Our book discusses the very same methods and assesses their importance, but does so at a somewhat lower mathematical level. Book is recommended for supplemental reading.

Ignatow RF, Mihalcea RF: Text Mining: A Guidebook for the Social Sciences. SAGE Publications 2016

Sketches many useful ideas, but without preparing the reader on how to actually carry out the analysis. Authors identify the right topics and the appropriate programs, but their text is not meant to be a “how to” book.

Jockers ML: Text Analysis with R for Students of Literature (Quantitative Methods in the Humanities and Social Sciences). Springer 2014

Includes quite a bit of basic material on R and R-Studio: How to install R, a review of basic R functions such as which, paste, strsplit, grep, for loops, bind, and apply, and a discussion of how to read text into R (using scan, XML files). The book uses Melville’s novel Moby Dick for illustration. Discusses concepts such as TTR (type-token-ratio), Hapax richness, KWIC (key word in context), clustering, and topics models. The text analysis in this book relies more on basic R functions than on the more powerful and easier-to-use R libraries used in our text.

Kwartler T: Text Mining in Practice with R. Wiley 2017

Good discussion of text mining with R, with many examples of R code. This book has a business focus, emphasizing the importance of text mining to improve the bottom line of a business. Views text mining as the process of distilling actionable insights from text in order to increase the competitive advantage.

Rockwell G, Sinclair S: Hermeneutica, Computer-assisted Interpretation in the Humanities. MIT Press, 2016

Thoughtful discussion of text analysis, and what it is and what it is not. Ample hands-on illustrations using their excellent and very flexible software Voyant.

Shawn G, Milligan I, Weingart S: Exploring Big Historical Data: The Historian's Macroscope. London: Imperial College Press, 2016

Excellent discussion, without getting too much into the technical details on how to carry out the analyses. Reliance on the software packages Voyant and Mallet. Recommended for further reading, especially for historians.

Weiss SM, Indurkhya N, Zhang T: Fundamentals of Predictive Text Mining (Texts in Computer Science). Springer 2015 (second edition)

Excellent illustration of big-data and machine learning tools for text analysis. While their main focus is not on the practical implementation of the discussed techniques, this text is highly recommended for additional reading.

**Useful corpora for possible study**

**United States Congressional Record: Speeches in the U.S. Congress from 1873 to 2017 (43rd to 114th Congress)**

Gentzkow, Shapiro and Taddy (2019) have made available all speeches in the U.S. Congress from 1873 to 2017 (43rd to 114th Congress) as transcribed in the United States Congressional Record. They obtained the digital text from HeinOnline who performed Optical Character Recognition (OCR) on scanned print volumes. The digitized text of speeches as well as the meta information for the speeches (name of the speaker, together with chamber/state/party affiliation, and the date of the speech) for each of the 72 consecutive sessions of Congress (spanning an uninterrupted period of 144 years) are available on their Stanford University website. Our Chapter 2 provides information on how to access this information.

Gentzkow, Matthew, Jesse M. Shapiro, and Matt Taddy. Congressional Record for the 43rd-114th Congresses: Parsed Speeches and Phrase Counts. Palo Alto, CA: Stanford Libraries [distributor], 2018-01-16. <https://data.stanford.edu/congress_text>

**Proceedings of the Old Bailey, 1674-1913**

A fully searchable edition of a large body of texts detailing the lives of common people, containing the records of 197,745 criminal trials held at London's central criminal court. Transcripts cover more than 200 years of trials in London (from 1674 to 1913) and contain 127 million words. Information was transcribed from pamphlets and published proceedings. Transcription errors were reduced by using two independent typists. <https://www.oldbaileyonline.org/>

**Index Thomisticus compiled by Father Busa**

In 1946, Father Busa started to construct the Index Thomisticus as a tool for performing text searches within the massive works of St. Thomas Aquinas (written in Latin). His plan was to conduct a comprehensive concordance study of this very large corpus. He wanted to construct an alphabetical list of words with citations of the passages in which each word was found. He wanted to know where a given word appears and describe the context of its appearance. It certainly was a daunting task to collect such information on every word in this very large corpus. In 1949, Father Busa met with Thomas J. Watson, the founder of IBM, and was able to persuade him to sponsor the Index Thomisticus. The resulting text analysis relied on punch cards and punch card readers (which were limited to 80 characters each). The work involved the transcription of phrases found in the text onto punch cards; duplication of the number of punch cards by the number of words on each card; lemmatization of words (that is, to break words down into their roots); alphabetizing the cards; and publishing the final product in printed form. The output provided, among others, words and their frequencies; words set out under hand-lemmatization; and keyword-in-context concordance.

The project lasted about 30 years, leading to the publication of 56 printed volumes of the Index Thomisticus in the 1970s. In 1989, a CD-ROM version was produced. In 2005, a web-based version made its debut. In 2006, the Index Thomisticus Treebank project started the syntactic annotation of the entire corpus. The work includes the work of Thomas Aquinas (118 texts) as well as 61 texts by other authors related to Thomas Aquinas. It contains approximately 11 million words, with each word morphologically tagged and lemmatized by hand. <https://itreebank.marginalia.it/view/download.php>

**The Brown corpus**

The Brown Corpus was compiled in the 1960s by Henry Kučera and W. Nelson Francis at Brown University as a general corpus (text collection) in the field of corpus linguistics. It contains 500 samples of English-language text, totaling roughly one million words, compiled from works published in the United States in 1961. It can be accessed through <https://github.com/ekinhbayar/brown-corpus>

**Project Gutenberg**

Project Gutenberg is a library of over 60,000 free eBooks. You will find the world's great literature here, with focus on older works for which U.S. copyright has expired. Thousands of volunteers digitized and diligently proofread the eBooks, for enjoyment and education. <http://www.gutenberg.org/>

**Reuters Corpora (RCV1, RCV2, TRC2)**

In 2000, Reuters Ltd made available a large collection of Reuters News stories to be used in research and development of natural language processing, information retrieval, and machine learning systems. Reuters Corpus Volume 1 (RCV1) contains 810,000 Reuters (English Language News stories) spanning the period from 1996-08-20 to 1997-08-19. <https://trec.nist.gov/data/reuters/reuters.html>

**MEDLINE Document Collection**

The Medline data set from the National Institutes of Health contains a large number of abstracts on medical subjects. Text data have been extensively reviewed and are of good quality. <https://www.nlm.nih.gov/databases/download/pubmed_medline.html>

**Tweets of President Donald Trump**

We have downloaded the tweets made by President Donald Trump between May 2009 and December 2019. The data, given in the text file **TrumpTweets.txt**, contains 43,918 tweets covering the period from May 2009 to December 2019. The tweets include 1734 retweets. The R program file (**RProgramTrumpTweets.txt**) processes the .txt data and generates an R worksheet (**TrumpTweets.RData**) that can be used for the analysis. The tweets include several emoji's. The R program illustrates how emoji's can be translated into descriptive text. The R worksheet includes information on the date of the tweets, an important meta variable for the text analysis. The three files can be found on the website.

**Further comments on computing and software**

We use for our text analysis **R**, the free software environment for statistical computing and graphics. However, we are eclectic in using whatever helps with textual analysis. The **Google Ngram Viewer** is helpful if one wants to learn how often certain phrases are used among all books in the Google collection and how their use changes over time. The packages **AntConc** and **Voyant** are especially useful for their excellent concordance and collocate tools. Concordance tools tell you about all text snippets around a specified key word in context (KWIC). You may want to see all text snippets in a corpus around a word such as “republic” or around phrases such as “republican form of government” or “post office” as this information will tell you about the context in which the word or phrase is used. Collocate tools allow you to further quantify the concordance information. For example, they can be used to obtain counts on how often the word “black” or “indian” occurs within a window of specified length around a specified KWIC word such as “slave”. Both software packages are flexible, allowing you to change the length of the window and focus on words before or after the given KWIC word of interest. **Mallet** is useful stand-alone Java-based package for machine learning applications in text analysis, including information extraction, natural language processing, document classification, clustering, and topic modeling.

There is no single gold standard procedure for text mining. You have to choose the method which is most convenient for your particular text mining application. In this book we have used the software language **R**, because it includes many useful packages for text analysis and visualization and because we were already familiar with it from our prior statistical work. Another excellent choice is **Python**, a similar free general-purpose programming language. Python requires a somewhat stronger computer programming background. Python may be a better choice if one looks for natural language processing libraries and if one deals with very large corpora. Of course, if the corpora are extremely large, one needs to think about distributed computing. There the work tasks are located on different networked computers which communicate and coordinate their actions by passing messages to one another.