Computational Sermon Analysis

Theo Pleizier

2023-12-15

# 1 Introduction

Computational sermon analysis (CSA) locates sermon analysis within a wider family of digital research methods. In digital humanities (DH) or computational humanities (CH) these methods have different names, such as ‘text mining’(Jockers and Underwood 2015), ‘corpus linguistics’ (Gries 2016; Stefanowitsch 2020), ‘corpus assisted discourse analysis’ (Gillings, Mautner, and Baker 2023) or ‘natural language processing’(McGillivray, Poibeau, and Fabo 2020). These methods have their origins in a variety of disciplinary traditions and represent different approaches, however, they all point to the interaction of the study of human discourse and computer science.

Increasingly, computational methods are used to study sermons (Agersnap et al. 2020; Agersnap and Nielbo 2022; Center, Washington, and Inquiries 2019). Immediately, the interdisciplinary nature of these studies comes to the fore. To mention only a few examples: Hans Malmström used computational methods to study linguistic features such as meta-discours of three denominational types of sermons (Malmström 2015b, 2015a). Robert Boss applies several digital humanities methods and visualisation techniques to a corpus with sermons by Jonathan Edwards (Boss 2020) and Michael Keller studied rhetoric and imagination in Edwards’ sermons (Keller 2018). Constantine Boussalis and others studied the nature of political speech in a dataset of over 110,000 US sermons (Boussalis, Coan, and Holman 2020). Anne Agersnap created a corpus of over 11,000 Danish sermons (Agersnap et al. 2020) and locates the computational study of sermon in the interdisciplinary domains of sociology of religion (lived religion), homiletics (the empirical turn) and digital humanities (computational methods) (Agersnap 2021, 14–16, 36–37).

Though preaching is primarily an oral event (Ong 2002), a sermon can also be understood as a ‘textual object’. This object is constructed by the preacher (‘sermon preparation’), it is performed within the liturgical community (‘the preaching event’) and it is processed by listeners (‘sermon reception’). Wilfried Engemann distinghuises between these types of ‘texts’ as manuscript (the product of sermon preparation) and auredit (the reception by the listener) (Engemann 2019, 11); while ‘transcript’ refers to the spoken text during a worship service. As a *textual object*, the sermon is a complex whole of intertextual references, speech acts, and a mixture of religious and everyday language. Three features of the sermon are important to mention here. First, a sermon is *structured language* (Grözinger 2008). Second, a sermon is *religious speech*, it communicates religion (Pleizier 2022). Thirdly, a sermon is *personalised speech*, it reflects the personal style of preachers. Sermons are speeches and as speeches they have considerable length. Analysis of a longer speech is often very time consuming. Further, a sermon is an example of a *weekly* speech. In the Netherlands only the amount of weekly sermons can be estimated between 3,000-5,000, a number based upon the sites of worship that are connected to the major streaming services for online streaming of worship.[[1]](#footnote-20) Only computers can process these enormous amount of textual data (‘big data’) and the field of *natural language processing* consists of methods to use computers to ‘understand’ natural languages (Khurana et al. 2023).

This chapter presents some basic techniques from the computational toolkit. To illustrate the complexity and the possibilities of computational techniques, I rely on a dataset that has been constructed from sermons that are available in the public domain, the sermons from John Henry Newman (1801-1890) and Charles Haddon Spurgeon (1834-1892). Both preachers belong to the Victorian era (Old 2007), the ‘golden age of sermon culture in Britain’ (Gibson 2012).

In CSA two strands of research converge: sermon analysis and digital theology (section 2). Next, the case study of ‘Victorian preaching’ introduces a type of research questions that can only be answered by relying to CSA (section 3). Though Newman’s and Spurgeon’s sermons can be downloaded freely for research purposes, the construction of a dataset (or ‘corpus’ as it is called in corpus linguistics) requires a specific type of workflow, common to the fields of data science and digital humanities (section 4). The major part of the chapter illustrates various computational analyses. First, basic operations in corpus linguistics are introduced to answer relative simple and straightforward research questions on the use of language in sermons. This section introduces explorative data analysis, statistical methods (keyness and lexical diversity) and sentiment analysis (section 5). Secondly, two advanced techniques demonstrate the application of artificial intelligence in sermon analysis. Following the typology of Van Attenveld and others, these techniques include inductive methods (unsupervised machine learning) and deductive methods (supervised machine learning) (Atteveldt, Trilling, and Calderon 2022) (section 6). The chapter closes with a brief reflection on the implications and limitations of CSA for the study and theory of preaching (section 7).

# 2 Digital theology and sermon analysis

Sermon analysis has been oriented towards mixed methods from the start (Hendrik J. C. Pieterse 2020; Moehn 2001; H. J. C. Pieterse 1995). Word frequencies (quantitative approach) and thematic coding (qualitative approach) were combined in an effort to apply the principles of empirical research into sermon analysis. The first approaches to the study of word frequencies in sermons can be traced back to 1973, when the Homiletical Workgroup (‘Homiletische Arbeitsgruppe’) applied content analysis to the study of sermons (Homiletischen Arbeitsgruppe 1973), ‘a large-scale project that could only be carried out with the aid of a computer’ (Engemann 2019, 384). At the turn of the century the developments in computer science, the availability of powerful personal computers and the accessibility of specialised software (including programming languages) opened up new possibilities for research. For example, Georg and Matthias Ballod analysed a corpus of 12 Pentecost sermons using software for calculating lexical diversity, word frequencies and key words. They performed various statistical calculations to study the readability of the sermons, the themes related to Pentecost, and the language that enables the listeners to be involved in the sermon (Ballod and Ballod 2000). Due to the technically improved methods in the field of artificial intelligence, the first quarter of the Twentyfirst century showed an increase in machine learning applications. The results of the study of Pew Research demonstrated the relationship between denominations and the length and content of sermons in a wide variety of US churches based upon a dataset of almost 50,000 sermons. Its use of computational methods reflects the current state of research and includes the application of machine learning (Center, Washington, and Inquiries 2019). The researchers wrote a so-called ‘webscraper’ to scrape the websites of churches to find the sermons to include in the dataset:

The custom-built program used a machine learning model to identify pages likely to contain sermons and a set of specially designed algorithms to collect media files with dates from those pages, identify the files containing sermons and transcribe those files for further analysis. (Center, Washington, and Inquiries 2019, 21)

Following the lead of these two examples, this chapter introduces basic (section 5) and advanced (section 6) computational methods for sermon analysis. These methods are presented in an order that follows a common workflow in data science: from exploratory data analysis and standarised methods of analysis to more sophisticated and custom-built approaches that apply existing algorithms or develop new models. The latter methods include machine learning. There is a range of existing software programs that can perform standard calculations, computational approaches however often involve computer programming (section 4).

The basic unit of computation is a number. Texts, therefore, need to be transformed in numbers. Document-term-matrices and word-frequency-tables are examples of text-turned-into-numbers.

First, document-term-matrices (or: document-feature-matrices) are a special kind of frequency tables. They turn text into numbers by counting how often terms (or: tokens) appear in documents. A document-term-matrix (DTM) ‘represents a *corpus* (or set of documents) as a matrix or table, where each row represents a document, each column represents a term (word), and the numbers in each cell show how often that word occurs in that document.’ (Atteveldt, Trilling, and Calderon 2022, 10.1) These matrices are often used to apply machine learning tasks, both unsupervised and supervised (section 6). An example of a DTM of 5 randomly sampled documents from the balanced corpus of Spurgeon’s and Newman’s sermons presents the documents (left column) in relation to the times the words occur in the document (other columns):

## Document-feature matrix of: 5 documents, 10 features (16.00% sparse) and 13 docvars.  
## features  
## docs holy spirit convey a chief truth of religion few words  
## 07\_16.txt 1 5 2 54 0 3 118 2 0 3  
## 07\_07.txt 2 1 0 76 1 1 134 3 1 2  
## 05\_11.txt 19 36 0 65 0 4 242 0 1 16  
## 09\_25.txt 7 4 0 43 2 4 161 6 1 10  
## 09\_22.txt 4 2 1 76 0 5 178 4 1 3

Compared to document-term-matrices, frequency tables are more straightforward representations of word counts in a corpus. After creating a balanced corpus of 56 sermons of Spurgeon and Newman[[2]](#footnote-24), frequency tables can be calculated. The example of a frequency table below presents 6 columns.

## all no\_stop spurgeon\_only spurgeon\_freq newman\_only newman\_freq  
## 1 the god jesus 853 world 594  
## 2 of us must 761 yet 452  
## 3 and shall never 741 let 384  
## 4 to christ see 725 life 374  
## 5 in man life 695 great 370  
## 6 is may spirit 689 though 365  
## 7 that one yet 661 st 349  
## 8 a lord let 606 time 349  
## 9 it upon go 600 faith 342  
## 10 he can sin 588 even 336  
## 11 you now faith 571 things 325  
## 12 not men said 558 see 324  
## 13 be say great 542 god's 318  
## 14 i know things 542 way 310  
## 15 we come like 537 first 298

The first column represents the raw, unfiltered data. The 15 most frequently used terms are listed in order of frequency, with ‘the’ being the most frequently used word in the corpus. This list, however, is not very insightful as the terms correspond with the most used terms in the English language (column 1). Therefore, data scientists apply a filter with so-called stopwords. This results in a list of words that already indicates that we might be dealing with a the specific genre ‘sermon’ (column 2). The final step is to identify the top 15 words in Spurgeon’s and Newman’s sermons respectively. The result provides a glance in the sermonic vocabulary of both preachers (column 3-6).

The table with word frequencies seems evident, yet it raises quite a few issues. First, the sermons are not of equal length. Spurgeon’s sermons are significantly longer than Newman’s, the numbers already suggest this. Second, the frequencies cover the sermons of each preacher taken from the 71 sermons that share the same biblical text. They provide an insight in the overall usage of words, not in relation to specific biblical texts. Third, there is something striking with two items in the list with Newman’s words: the ‘st’ (term 10) suggest a combination with other words (such as ‘St. Paul’) and ‘God’s’ (term 13) is interpreted as a single token. The field of *quantitative corpus linguistics* (Gries 2016) has solutions to deal with these issues. Compounds are a special kind of word-pairs (or bigrams). A frequency table of all compounds that start with ‘st’ shows indeed that Newman uses ‘st’ in combination with biblical saints, with St. Paul being the champion with 194 occurrences in 52 (out of 71) sermons.

## feature frequency rank docfreq group  
## 1 st\_paul 153 1 38 all  
## 2 st\_paul's 44 2 20 all  
## 3 st\_john 37 3 23 all  
## 4 st\_peter 31 4 21 all  
## 5 st\_james 14 5 8 all

Collocations are another central method in corpus linguistics: ‘Collocations are co-occurrences of words, which are then referred to as *collocates*’ (Gries 2016, 16). A collocation display with the term ‘God’s’ shows the most frequent combinations, starting with ‘God’s grace’ which occurs 32 times in the corpus of 71 sermons.

## collocation count count\_nested length lambda z  
## 102 god's grace 30 0 2 4.711581 23.21340  
## 115 god's mercy 22 0 2 5.524740 21.95441  
## 792 god's will 19 0 2 2.730243 11.56479  
## 223 god's sight 16 0 2 4.750889 17.48036  
## 245 god's service 13 0 2 5.355281 16.94981  
## 321 god's presence 13 0 2 4.591405 15.49319

Theologians are known with the phenomenon ‘concordance’. In Biblical studies concordances are used to list the biblical verses where Hebrew or Greek words occur. In quantitative linguistics, concordances provide tables with keywords in context. For example, a sample of 5 occurrences of the keyword ‘God’s grace’ in the corpus with Newman’s sermons demonstrates how the keyword-in-context function enables the researcher to study keywords in their context:

## Keyword-in-context with 5 matches.   
## [07\_16.txt, 2978:2979] plain pledge without reserve of | God's grace |  
## [05\_12.txt, 3746:3747] God which was in me | God's grace |  
## [07\_02.txt, 2348:2349] ourselves creating within us under | God's grace |  
## [09\_02.txt, 832:833] great but he may through | God's grace |  
## [08\_14.txt, 2585:2586] for the successive motions of | God's grace |  
##   
## given to the souls of   
## was sufficient for him It   
## those parts of the Christian  
## become a saint ever so   
## to excite them to action

In sum, frequency tables entail more complex techniques than simple word counts. Further, the examples of ‘st’ and ‘God’s’ also show that the term ‘word’ is not unambiguous. However, frequency tables are the starting point for other computations, such as keyness and lexical diversity (section 5).

Word frequencies, stop-words, tokens, machine learning, document-term-matrices and collocations. These are new terms in the area of sermon analysis. A computational approach adds new terminology to the field of practical theology, often derived from a conglomerate of disciplines and methods that together shaped digital humanities in the last decades: quantitative corpus lingustics, natural language processing, data science and text mining. These methods have in common that they provide the research community with new empirical approaches, often relying upon statistical techniques. The developments are still very recent. Hence, the distinction between disciplines remains rather fluid and fuzzy (Wymer 2021, 2). The digital and the computational are distinct according to this helpful description: ‘the term *digital* refers to information in binary form while *computational* refers to processes performed by algorithms’ (Drucker 2021, 1)

In the pioneering work of Peter Phillips and others in Durham, digital theology is described according to four ‘waves’ - a metaphor derived from Heidi Campbell’s work on digital religion. Phillips and others distinghuish between: 1. the use of digital theology in teaching; 2. theological research enabled by digitality; 3. reflexive theologically-resourced engagement with digitality; and 4. a prophetic re-appraisal of digitality (Phillips 2020, 37–40). Erkki Sutinen and Anthony-Paul Cooper understand digital theology as the dialogue between theology and computer science:

The particular viewpoint of Digital Theology focusses on expression of faith, as do the classic fields of Christian theology: exegetics, systematic theology, church history and practical theology when they explore and analyse the sources, methods, trends and practices of the intellectual exercise of expressing and conveying faith as various forms of information. The interrelatedness between faith as spiritual belief and its intellectual, tangible expression as creed, doctrine or concrete behaviour forms the basis for the dialogue of theology and computer science applied in IT, as Digital Theology. (Sutinen and Cooper 2021, 1)

Sutinen and Cooper specifically locate digital theology in practices. A similar understanding of the relationship between digital humanities and theology is proposed by Stefan Karcher who argues that practical theology is well-equipped to enage with digital humanities. Besides the study of online communication and the historical texts, Karcher specifically mentions sermon analysis as an example in which digital humanities and (practical) theology fruitfully interact (Karcher 2020, 141).

# 3 The Case Study: Preaching in the Victorian Era

In the year Friedrich Schleiermacher died, Charles Haddon Spurgeon (1834-1892) was born. It was the age of the Victorians. C.H. Spurgeon the popular Baptist preacher from London was a contemporary of John Henry Newman (1801-1890). Newman started as an Anglican parish priest and served as university chaplain in the city of Oxford, but in 1845 he converted to Roman Catholicism.[[3]](#footnote-26) Pope Leo XIII created him a Cardinal in 1879. Spurgeon and Newman are two famous preachers in the Victorian Age (Old 2007).

Spurgeon became one of the most known nineteenth century preachers. His sermons are widely distributed; publication of his sermons started shortly after they were delivered. Several historical and homiletical studies researched aspects of his preaching (Edwards 2004, 455–461;@old2007, 222–43). Richard Lischer characterises his preaching as a combination of ‘oratorical prowess and evangelical fervor with a deep concern for social issues’ (Lischer 2002, 316). Spurgeon’s sermons are included in many online repositories, such as *Christian Classics Ethereal Library* (CCEL) that offers 63 volumes with more than 3500 sermons in open access.[[4]](#footnote-27)

Equally important as a preacher, was John Henry Newman. His sermons have been published according to the two phases in his ecclesial life: his Anglican sermons and the sermons after he became Roman-Catholic in 1845. The literary legacy of Newman has been digitally processed, including the handwritten manuscripts of his sermons.[[5]](#footnote-29) The fulltext, open source availability of his sermons, the *Parochial and Plain Sermons* (1834-1843) and some of his occasional sermons can be found at the digital repository *Newman Reader*. This sermons only cover the final part of Newman’s Anglican period (217 sermons).

The sermons of these famous preachers have survived due to their intentional publication strategies. Early adapters of the internet have added digitized versions of their (published) sermons in online, full-text, open access repositories.[[6]](#footnote-32) From Spurgeon more than 3500 sermons are included[[7]](#footnote-34) ; from Newman more than 220 sermons were scraped[[8]](#footnote-35) from the online repository.[[9]](#footnote-36)

# 4 A Typical Workflow to Create a Corpus: Programming, Pipelines and Preparing Data

In their article ‘Sermons as Data’, Anne Agersnap and others describe the construction of a corpus of sermons (Agersnap et al. 2020). They describe the three stages of sampling, cleaning and annotating; stages that are common in a data science workflow. Like any empirical method, a computational approach has a structural logic and a toolbox. This logic in is expressed in so-called ‘pipelines’; a sequence of instructions that a computer can process, usually starting with ‘input’, followed by a sequence of operations, and resulting in ‘output’. The output becomes ‘input’ for a new pipeline. Pipelines structure a computational workflow. A simple example of a pipeline - or actually a stage in a larger pipeline - is the following:

# input: read a folder with textfiles  
sermon\_texts <- readtext("gen/newman/txt/\*")  
  
# apply the function 'corpus' from the quanteda-package  
sermon\_corpus <- quanteda::corpus(sermon\_texts)   
  
# output: save the corpus in a file  
save(sermon\_corpus, file = "gen/newman\_corpus")

These lines of code take a folder with a large collection of text-files as its input and stores it in an object called ‘sermon\_texts’. Next, a function is applied to the object, namely to store the entire collection of text-files in a special format, a ‘corpus’ format. The function is part of a small program, a package called ‘quanteda’. Finally, the corpus-object is saved to a file called ‘newman-corpus’. This file is the output. This object can be loaded as input for a new series of operations. This example illustrates two other aspects of a computational workflow. First, the primary toolbox for computational analysis is computer programming. Second, before any analysis takes place, data should be prepared in a format - a dataframe or a matrix - that a computer can process. Data preparation is actually a procedure that covers many different operations, such as sampling, scraping, cleaning (or wrangling), and annotating. Everything that is needed to have a dataset that can be used for further analysis. This section introduces programming, pipelines and preparation of data. By way of practical illustration, it presents the various steps to create a corpus of the sermons by Newman and Spurgeon, the dataset that will be used in the various analyses that are presented in the next sections.

The journey starts with acquiring the raw data. Next, the raw data is cleaned to create a consistent dataset. Finally, the data is annotated with so-called metadata. Each of these operations are done by a sequence of small computer programs (‘code’), a list of pipelines. First, we need to locate the data, the websites that contain the sermons. For Newman we need the website ‘newmanreader.org/works/index.html’ which contains 8 volumes of the *Parochial and Plain sermons*. Spurgeon’s sermons are freely available at ‘ccel.org/index/author/S’; this index contains links to 63 volumes with sermons.[[10]](#footnote-39) Now the first program needs to be written to download the two sets of sermons.

## Programming

Computational sermon analysis starts with selecting and learning a *programming language*. Instead of using existing software suites for textanalysis, such as LancsBox or AntConc[[11]](#footnote-41), computational analysts often use one - or a combination - of two major programming languages: Python[[12]](#footnote-44) or R[[13]](#footnote-46).

In the examples presented below I use R(R Core Team 2021), a language that was developed for statistical analyses and is currently a widely used in academic research. R is actively developed and because it is open source it has many community-developed libraries to add specific functionalities; its documentation is amply available, also with a specific interest in textmining (Jockers 2014), or advanced techniques like machine learning (Boehmke and Greenwell 2019) or deep-learning (Chollet and Allaire 2018). R can be used to read all sorts of database-structured files or to scrape data from websites.[[14]](#footnote-49)

After installing R we need to install the libraries[[15]](#footnote-51) that we need for the examples in this chapter, such as *readtext*, *rvest* and *quanteda*.[[16]](#footnote-54) Then data needs to be located and read into the program.

This is where data analysis starts: fetching data from a local repository or a source that is available online. For both preachers, J.H. Newman and C.H. Spurgeon, we have to prepare a script that downloads (or: scrapes) the data from the internet. This requires a little research on where the sermons can be found and how they are stored on the servers. It appears that Spurgeon’s sermons are stored at the website of *Christian Classics Ethereal Library* in 63 volumes that together contain more than 3,500 sermons; the volumes can be downloaded as separate textfiles. So we need a script that downloads all 63 textfiles in one flow. Newman’s sermons can be found at *Newman Reader*, the indexes of 8 volumes of the *Parochial and Plain Sermons* can be found on separate pages; a 9th volume with occasional sermons (*Sermons on Subjects of the Day*) contains an additional 26 sermons. In total 217 sermons by Newman are currently available in plaintext in the public domain. The scripts that downloads Spurgeon’s and Newman’s sermons consists of entering the references to the webpages (*urls*) into variables that form the input for a larger ‘loop’ that walks through all the pages, downloads the file or scrapes the webpage and stores the content into separate text-files.

The small R-program that does this for Newman’s sermons looks like this:

# load packages  
library(here)  
library(rvest)  
library(stringr)  
  
# input: construct urls for the volumes with sermons  
volumes <- 8  
newmanurl <- "https://www.newmanreader.org/works/parochial/volume"  
newmanurls <- paste0(newmanurl,1:volumes,"/")  
outputpath <-"gen/newman/txt/"   
if(!dir.exists(here(outputpath))) dir.create(here(outputpath), recursive = TRUE)  
  
# loop over the volumes to find urls for the sermons  
for (u in seq\_len(volumes)){  
 newman\_volume <- read\_html(newmanurls[u])  
   
 sermonlist <- newman\_volume %>% html\_nodes("a") %>% html\_attr("href")  
 sermonlist <- sermonlist[str\_detect(sermonlist,"sermon\\d")]  
 sermonlist <- sermonlist[!is.na(sermonlist)]  
   
 sermonurls <- paste0(newmanurls[u], sermonlist)  
  
 # loop over the urls of the sermons to scrape the sermon  
 for(s in seq\_along(sermonurls)){  
 newman\_sermon <- read\_html(sermonurls[s])  
 sermontext <- newman\_sermon %>% html\_text2()  
 sermonfile <- paste0(str\_pad(u,2, pad = "0"),"\_",  
 str\_pad(s,2, pad = "0"),".txt")  
   
 # output: store the sermon in a file on the computer  
 writeLines(sermontext, here(outputpath, sermonfile))  
 }  
}

Comments in the code (prefixed with a hash-sign, #) explain what the code is doing. For reasons of readability and brevity, the next examples will only present these comments in the form of pseudocode:[[17]](#footnote-55)

SET number\_of\_volumes to 8  
COMBINE the main-url with the number\_of\_volumes urls  
FOR each volume   
 READ content of the url  
 EXTRACT list of urls that contain individual sermons  
 FOR each sermon-url  
 READ content of the sermon-url  
 SET filename to volume\_sermon.txt  
 SAVE content of sermon-url to filename  
 ENDFOR  
ENDFOR

The output of the code is a list with 217 text-files, each containing one of Newman’s sermons. A similar program is written to download (‘scrape’) the sermons by Spurgeon. The resulting textfiles, however, need a lot of data cleaning. They contain information that not properly belongs to the sermon, such as a sermon number, a title, the biblical text, a copyright statement and footnotes. To analyse the text of the sermon, this information need to be removed from the files.[[18]](#footnote-57) Further, some of this extra information is worth keeping, such as the date when the sermon was preached, the biblical text or the number of the sermon and the volume where it came from. This is called ‘meta-data’. To extract the meta-data from the files and to clean the files from information that is not part of the actual sermon, we need a new part of our computational pipeline.

## Pipelines

The workflow in computional analysis is often described as a *pipeline*. A pipeline consists of a series of functions or scripts that are sequentially executed. Every next script uses the output of a previous script as its input. For instance, the output of data fetching is used as input for the next step in data preparation: the texts of all Luthers sermons are read into the computer (*input*) and subsequntly stored into a dataset (*output*); in the next part of the pipeline this dataset (*input*) is transformed in a database that consists of single sermons and each sermon is stored as a set of single lines (*output*).

Data preparation is that part of the pipeline in which the data becomes available for the various methods of analysis, such as textmining, topic modelling, sentiment analysis and word embeddings. The results of each of these methods (*input*) is finally piped into a script that produces a visualisation, a table or even a complete report (*output*). In their study of 11,955 sermons Anne Agersnap and Kristoffer Laigaard Nielbo describe three computational pipelines, a pipeline for wordcount, a pipeline for seeded semantic network analysis, and a pipeline for topic modelling. Each of these pipelines consists of several steps, like tokenization, frequency tables, word embeddings, concatenation and visualisations (Agersnap and Nielbo 2022).

A good practice in working with pipelines is a standard organisation of files in folders, for instance datasets in /data, scripts or code in /src, and the generated output by scripts in /gen. [[19]](#footnote-59) Hence, since the data for our example of Victorian preaching is all fetched through a script, the output is stored in /gen/newman and /gen/spurgeon respectively. The data-folder contains only one file: an index of all 3500+ Spurgeon-sermons was found on the internet, a separate file that was downloaded manually and used for automatic collection of meta-data.[[20]](#footnote-62)

For the Victorian-preaching project the following pipeline was created that prepares the data for analysis, the pipeline contains multiple scripts:

1. fetch data: scripts that downloads the volumes with sermons from Spurgeon and Newman
2. process data: the text of the sermons is extracted from the volumes, based upon headings and footers. Some information (such as dates, biblical texts) is stored as meta-data.
3. clean data: internal references such as footnotes and biblical texts are removed from the sermons; errors in the Spurgeon-files are corrected based upon the downloaded index-file (see above).
4. build corpus: two large corpus-files are built from the two sets of text-files. The Spurgeon-corpus consists of 3,516 sermons, the Newman-corpus contains 217 sermons. A balanced corpus-file is created based upon sermons by both preachers, based upon the same biblical text.

This pipeline is common for a computational project, but the individual scripts are adapted to the specific datasets, document variables of a corpus, analytic goals or reporting purposes. As with all research projects, the research questions will be leading in programming a series of pipelines to answer the research questions. Here, computational analysis does not differ from any methodology that includes the steps of data collection, data analysis and reporting data.

## Preparing data

After fetching the data from websites, the data should be prepared for analysis. Sermons are large texts, consisting of many words and sentences. Often the texts are not entirely consistent or contain words or sentences that do not properly belong to the sermon, such as a brief heading that explains the situation in which a sermon was preached or a few tail-lines that contain information on the publication of the sermon. Data preparation aims to create a coherent and balanced dataset. Finally, a corpus can be created.

In general, data preparation consists of three steps. First, the downloaded and scraped sermons that have been stored in single files (*fetch data*) is processed. Each file is automatically checked. Headings and footers are removed, according to rules that are defined after a manual inspection of a few single files. In the case of the sermons of Newman and Spurgeon, the headings contain relevant metadata. For instance, the raw data - one of the Spurgeon-files - starts with the following header:

Characteristics of Faith  
  
 A Sermon  
  
 (No. 317)  
  
 Delivered on Sabbath Morning, May 27th, 1860, by the  
  
 REV. C.H. SPURGEON  
  
 At Exeter Hall, Strand.  
  
 "Then said Jesus unto him, Except ye see signs and wonders, ye will not  
 believe."--John 4:48.  
  
 YOU WILL REMEMBER that Luke, in his letter to Theophilus, speaks of  
 things which Jesus began both to do and to teach, as if there was a  
 connection between his doings and his teachings. In fact, there was a

The actual sermon starts with the sentence: ‘You will remember’ Yet, the other lines contain relevant information: the title of the sermon (*Characteristics of Faith*), the date of preaching (*May 27th, 1860*), the place (*Exeter Hall*) and the Biblical text (*John 4:48*). A random check learns that this structure is similar for each sermon. It appears that the Biblical texts marks the end of the heading. A script determines the line where the heading ends and the sermon starts. The script defines a function to find the biblical reference at the start of each new sermon:

DEFINE pattern to extract Biblical text  
DEFINE function fetch\_volume  
 (code)  
DEFINE function fetch\_sermon  
 (code)  
DEFINE function find\_scripture   
 CALL fetch\_volume to select volume-file  
 SELECT sermon from volume  
 FOR each line of the sermon until the 90th line  
 IF line contains pattern THEN  
 biblical text found  
 STORE line number  
 break loop  
 ENDIF  
 ENDFOR

Another script extracts the metadata (title, date, place, text) from the heading and stores it in a table. For Newman, an additional table is needed to reconstruct the dates of the sermons. This table is available online, so the script contains code to connect the sermons with the dates that have been reconstructed by earlier researchers.[[21]](#footnote-65) For each sermon, two dates are added: the date when the sermon was preached (date of delivery) and the date when the sermon was published (date of print). In the case of Spurgeon’s sermons the dates of the sermons are a bit more complex. Quite a few sermons include an extra date of publication. This date refers to the date when the sermon was published in a magazine before it was added to the collection of printed sermons. Further, Spurgeon’s sermons often indicate whether the sermon was preached on a Sunday or a weekday (15%). Time indicators are missing from 583 sermons (16,5%).

Second, the processed files are cleaned. It appears that there is material in the files that should be removed, such as footnotes added by the publisher, biblical references between brackets (in Newman’s sermons) or - as it turned out - larger parts called ‘expositions’ (in Spurgeon’s sermons). Though those are found out by manual checks, it is undoable to check and clean over 3,000 files. Again, a script is written that attempts to remove these pieces from the text based upon textual markers. For instance, ‘expositions’ always start with the word ‘Exposition’ at the beginning of a new paragraph and footnotes are always surrounded by square brackets.

The third and final part of the pipeline ‘data preparation’ is the creation of single files with the sermons. Based upon the tables with metadata the sermons are extracted from the raw data. This results in 217 text-files with sermons by John Henry Newman and 3,516 text-files with sermons by Charles Haddon Spurgeon. Now we are ready to create a corpus, or actually, multiple corpora

A corpus is a dataset. Yet it is different in size from a dataset that is collected or generated for qualitative analysis:

The set of texts or *corpus* dealt with is usually of a size which defies analysis by hand and eye alone within any reasonable timeframe. It is the large scale of the data used that explains the use of machine-readable text.(Hardie and McEnery 2011)

In computational analysis, a corpus usually refers to a single object that the computer can read and analyse, rather than a folder full of files. Therefore, we need to combine all the single files into one object (a ‘corpus’). Usually, relevant metadata is added to the corpus. Metadata can be used to make specific selections from the corpus, such as all sermons by Newman, preached in the year 1843. The next figure presents the output for a summary overview of a sample of five sermons taken from both the Newman-corpus.

Based upon the metadata of both corpora, it is possible to compare the sermons by both preachers in relation to the Biblical texts. A comparison of both corpora demonstrates that both preachers have sermons on 56 similar biblical texts and on 148 similar biblical chapters. In the final stage of creating a corpus, three different balanced corpora are created:

1. a random sample of 150 sermons by each preacher;
2. 56 sermons on the same biblical texts by each preacher;
3. 148 sermons on the same biblical chapter by each preacher

For example, the randomly sampled balanced corpus is created according to the following instructions:

LOAD corpus-spurgeon  
LOAD corpus-newman  
ADD document variable with the name of the preacher to both corpora  
SET sample-length to 150  
DRAW sample from corpus-newman with size sample-length  
DRAW sample from corpus-spurgeon with size sample-length  
CREATE corpus-balanced by combining both samples

After data-preparation, pipelines are designed to select and to process data (data analysis) and to report data (creating graphs, tables and plots). Ideally, when running a script that produces a plot, the script checks if the previous steps in the pipeline and runs scripts that are needed to produce the plot. Hence, the next part of the chapter introduces a few analytic tools. Each of these analyses, such as *lexical diversity* or *topic modelling*, is done in a separate script and starts with reading the output generated in the previous pipelines followed by selecting and processing variables from the metadata or subsets from the corpus to perform the analysis. The plots that were included were also generated in the analytic pipelines.

Finally, the phase of analysis can start. The following two sections contain a selection of analytic techniques to study the data and answer the earlier formulated research questions.

# 5 Basic Operations in Corpus linguistics

Corpus linguistics is about frequencies (Gries 2016, 141). If words can be counted, than calculations and more complex computations can be done with words, including advanced statistical techniques.

Frequencies are often based upon single words (‘tokens’) or combinations of words (‘n-grams’). Tokens differ from words, as for tokens decisions have to be made what counts as the most basic linguistic unit. For example, do we treat ‘God’s’ - as it occurs relatively often in Newman’s sermons - similarly to the noun (God) or does the analyst take it as a separate word, the genetive, such as in ‘God’s grace’? Collocational and concordance analysis can shed more light on this (see tables XXX above). Hence, the four basic tools for computational analysis are: frequency analysis (‘tokens’), collocational analysis (the co-occurance of words), concordance analysis (‘keywords in context’) and keyword analysis (‘keyness’) (Gillings, Mautner, and Baker 2023, 13). These four tools function as basic techniques to perform more advanced statistical operations, such as *keyness* (significant key-terms in a corpus compared to a target corpus), *lexical dispersion* (distribution of tokens in a corpus), *lexical diversity* (measures for the lexical ‘richness’ of discourse) or *document similarity* (cluster analysis on document features).

Often tokenization - the breaking up of a corpus in single ‘words’ - is followed by automatic tagging. Part-of-speech (POS) tagging and Named-entity-recognition (NER) tagging provide additional features, such as grammatical functions (POS) or identification of proper names (NER). Tagging is often done by trained algorithms and is language-dependent. For sermon analysis, POS tagging makes it possible to study the use of nouns or verbs in sermons and NER tagging helps to identify proper names in sermons to study how biblical characters are portayed in sermons, how Trinitarian language is distributed in the sermon or how names of contemporary persons or places function in homiletic discourse. Tagging is a technique in natural language processing and based upon annotation specifications followed by implementing a machine learning algorithm (Pustejovsky and Stubbs 2013). These advanced statistical and machine learning techniques, however, are still based upon frequency analyses.

## Exploratory Data Analysis

Data science and computational analyses often start with exploratory data analysis (EDA), invented by John Tukey in 1977. EDA is a way of describing the dataset whether a deductive (hypothesis-testing) or an inductive (hypothesis-generating) research design is applied. The researcher needs to know the data before more complex analyses are performed. EDA often involve frequency tables (see section 2), descriptions of document variables of a corpus, and visualisations of aspects of the data. EDA can result in more complex statistical operations such as cluster analysis (Atteveldt, Trilling, and Calderon 2022, chap. 7.3). Grolemund and Hadley describe EDA as an iterative cycle from generating questions about the data, through searching for answers by visualising; transforming, and modelling data, to refine questions and/or generate new questions (Grolemund and Wickham 2017, chap. 7). This section presents five examples of EDA: a simple table that counts sermons, a visualisation of the length of sermons over time, a lexical dispersion plot, a visualisation of the different vocabularies of the preachers (keyness) and a measurement of lexical diversity. EDA techniques provide insights in the data that are impossible to acquire by manual analysis. For homiletical research this means that large datasets can be approached quickly, from simple counts to more complex visualisations.

### Counting sermons

A simple question in quantitative studies is: how many. Especially in the Spurgeon corpus, this question emerges in relation to 3,516 sermons. How many of these sermons are from the same biblical passages? The Spurgeon-corpus contains sermons from 1854 to 1891, 37 years of preaching ministry. Being a Baptist, Spurgeon didn’t necessary follow a lectionary. His choice of texts therefore gives an impression of his theology. So, we ask the question how often did Spureon preach from the same passages during this period of 37 years?

A small script is written to make the calculation:

LOAD the Spurgeon-corpus  
TRANSFORM the column with Scripture-references  
 REMOVE from the references everything after the verse indicator ":"   
 STORE the references with biblical book and chapter in book-chapter variable  
CREATE a table that counts how often chapters are referred to   
SORT the table in decreasing order (high values first)  
PRINT the top of the table (biblical chapters that are preached from 20 or more times)

The resulting table already is insightful. Nine biblical chapters have been preached from 20 or more times. Spurgeon preferred to preach from New Testament passages, with the book of Romans and the gospel of John being his favorites. Psalm 119 clearly was his most often used passage to preach from. Theologically, it is striking that two passages linked to the classic doctrine of atonement (Isaiah 53 and 2 Corinthians 5) are among the most preached passages.

## chapter freq  
## 1 Psalm 119 33  
## 2 Romans 8 32  
## 3 John 6 29  
## 4 John 14 28  
## 5 John 1 24  
## 6 Hebrews 11 23  
## 7 Isaiah 53 22  
## 8 Romans 5 22  
## 9 2 Corinthians 5 20

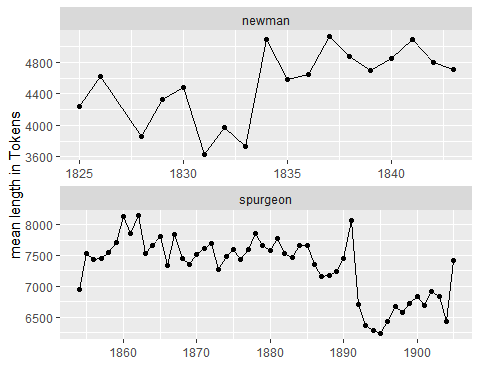
### Sermon length over time

The first comparison between the two preachers is the length of their sermons. Evangelical-baptist preaching may result in longer sermons than Anglican preaching. So the first hypothesis is that Spurgeon’s sermons are longer than Newmans. Further, given the amount of sermons (18 years of preaching by Newman, 37 years by Spurgeon), it is possible to gain insights in the developments of the sermons over time.

A corpus-object contains the number of Tokens in each document. An overview of the sermons by Newman provides insight in the number of sentences, tokens and different tokens (types) in the entire corpus.

## Corpus consisting of 5 documents, showing 5 documents:  
##   
## Text Types Tokens Sentences volume nr scripture year printed  
## 09\_16.txt 1254 6076 191 9 207 Isaiah 2:2 1842 1843  
## 02\_07.txt 862 3099 86 2 33 Matthew 3:15 1831 1835  
## 06\_18.txt 999 4977 121 6 148 2 Corinthians 10:5 1840 1842  
## 05\_05.txt 1094 4695 149 5 111 Philippians 4:4 1839 1840  
## 08\_07.txt 1175 5364 161 8 180 2 Kings 22:19, 20 1830 1843

An important technique in data science is visualisation of data. We quickly get an idea of the distribution of variables and patterns in the data. A small script calculates the mean length of all the sermons by one preacher grouped per year.

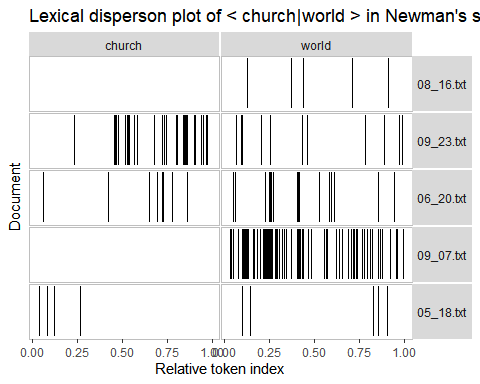


The first hypothesis is confirmed: over the years Spurgeon’s sermons are considerably longer than Newman’s sermons. Newman’s sermons have length between 3600-4600 tokens, while Spurgeon’s sermons range between 6250-8000 tokens. The plot also reveals that in Newman’s sermons, the sermons become slightly longer after 1832. The first sermon in 1833 was preached in July 1833, when Newman returned from a trip to Europe. After his return, it seems that he started to preach longer than before his trip. The plot of Spurgeon’s sermons shows a significant decrease of length in preaching around 1891. Spurgeon fell ill in 1891 and died in January 1892. His sermons continued to be read until 1905. Most of these sermons were also preached earlier, so it seems that in the published sermons by Spurgeon the sermons that were chosen to be read after his death were selected from the shorter sermons. This hypothesis, however, needs to be tested by an more detailed study. But this illustrates the significance of exploratory data analysis: it generates new questions based upon insights that are gained from the dataset as a whole.

### Lexical dispersion

Three other techniques from quantitative linguistics can be grouped under the heading of EDA. The first is called *lexical dispersion*, the distribution of a certain term in a corpus. Gries comments that it is ‘important not to rely only on the frequency of words, but also to consider how (un)evenly words are dispersed.’ (Gries 2016, 15). Gries also suggests a measure for dispersion (DP) which can be computed for each corpus file, depending upon the size of the file and the relative fequency of the word that is studied (Gries 2016, 179).

Given the list of frequent words in Newman’s sermons (see above), we may wonder how the terms ‘church’ and ‘world are distributed in Newman’s corpus. There are several ways of projecting lexical dispersion. For instance, concatenating all documents in chronological order, will give the dispersion of the terms from the first sermons (1825) to the last sermons (1843) in the dataset. To illustrate the technique of lexical dispersion, a sample of 5 random sermons are drawn from the corpus and the terms ’church’ and ‘world’ are indicated in each sermon. This gives an idea where in the sermons Newman uses the terms and how often the terms occur in individual sermons. This plot quickly shows that there is one particular sermon that has ‘world’ as its topic, namely corpus-file 09\_07.txt. This is a sermon on Proverbs 11:21. The sermon with more interest in ‘church’ is sermon 09\_23.txt, a sermon on John 4:42.



Lexical dispersion plots relate the frequency of a word to its location in the corpus. The researcher does not need to do all the calculations since the R-package *Quanteda* takes care of this. The plot above can be generated through a few simple lines of code

DEFINE search-terms  
CREATE sample corpus by drawing a sample of five documents from the corpus  
TOKENIZE the sample corpus  
APPLY keyword-in-context on the search-terms to the tokenized sample corpus  
PLOT the keyword-in-context with the function textplot\_xray()

Exploratory data analysis includes techniques that concern serious statistical calculations, though these calculations are often taken care of by predefined functions, one of these is ‘keyness’’.

Lexical dispersion plots relate the frequency of a word to its location in the corpus. The researcher does not need to do all the calculations since the R-package *Quanteda* takes care of this. The plot above can be generated through a few simple lines of code

DEFINE search-terms  
CREATE sample corpus by drawing a sample of five documents from the corpus  
TOKENIZE the sample corpus  
APPLY keyword-in-context on the search-terms to the tokenized sample corpus  
PLOT the keyword-in-context with the function textplot\_xray()

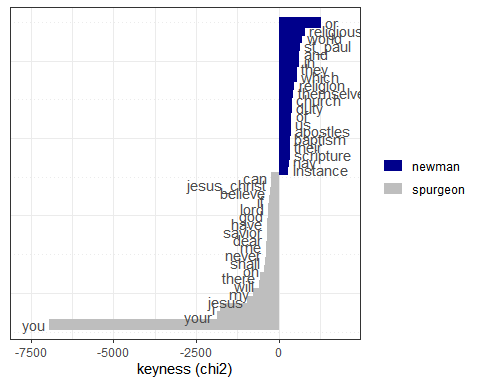
Exploratory data analysis includes techniques that concern serious statistical calculations, though these calculations are often taken care of by predefined functions, one of these is ‘keyness’’.

### Keyness

Now the lexical dispersion of the two terms of ‘church’ and ‘world’ has been clarified for the Newman’s sermons, the question emerges how Newman uses these terms in relation to Spurgeon. Keyness measures the frequencies of a target corpus against a reference corpus and calculates the ‘relative attraction of each word to the target corpus of a keyness statistic’ (Gries 2016, 197). One could, for instance, relate the terms of the sermons (target corpus) against a larger database of spoken English (reference corpus). It will become clear that sermons will reflect a specific type of language, language that is connect to church as a social institution. The language of a sermon is connected to the social roles of preachers and congregations: ‘a preacher cannot give a sermon unless there is (at least) a (potential) congregation’ (Stubbs 2010, 34). Keyness is a statistical measure (e.g. Chi2) that measures how certain words characterise individual texts (Stubbs 2010, 25).

In the first table on word frequencies, we already got an impression of the different vocabularies of Newman and Spurgeon. The calculation of keyness provides an indicator how different the vocabularies are. To make a relatively fair comparison, only the sermons are used that are grounded in the same biblical texts. Hence a balanced corpus of 56 sermons of each preacher is used. The code for calculating and plotting keyness looks like the following:

READ the balanced corpus on the same biblical texts  
TOKENIZE the balanced corpus and remove punctuation, symbols and numbers  
COMBINE a few phrases, such as 'I am' and combination with 'St' (such as St. Paul)  
CREATE a document-term-matrix for each preacher  
CALCULATE keyness of Newman (target) against Spurgeon (reference)  
PLOT the top 20 terms



Apparently, Newman is more likely to make comparisons in his sermons and to have his listeners choose between alternatives given the fact that his use of the term ‘or’ differs most from Spurgeon’s. Denis Robinson points to the fact that Newman often uses ‘binaries’ as a rhetorical device: “Newman often places two or three terms together, drawing dramatic contrasts and inviting comparison, yet remaining rather vague as regards the way in which the terms actually fit together.” (Robinson 2009, 245). On the other hand, Spurgeon addresses the audience far more directly in the second person, ‘you’. Perhaps this fits Oliphant Old’s description of Spurgeon’s sermons as evangelistic (Old 2007,), the ‘you’ being the audience that is addressed by the one who proclaims the gospel.

Exploratory data analysis is endless and each plot that is generated or technique that is applied opens up new findings in the data. For example, clustering techniques (used for document comparisons) or lexical diversity measures (that determine type-token ratios). Computational analyses often start with a lengthy exploration of the data, but the research questions that guide a project often determine that specific computational operations need to be applied. One of the questions that cannot be answered easily by manual coding but requires the use of computational power, is the question how the narrative arcs or rhetorical structures of sermons can be studied. One approach is to apply sentiment analysis, in order to find out how the emotional tone or the mood of sermons develops from the opening lines to the closing words.

## Sentiment analysis

Sentiment analysis is among the well-known tools in *natural language processing*. It has been applied to study positive and negative sentiments in customer reviews (movies, restaurants) or in social media messages (tweets). Anthony-Paul Cooper, for instance, studied the relationship between the sentiment of church-related tweets and church growth. Cooper applied *SentiStrength*, an algorithm that ‘provides a rating of the magnitude of positive sentiment (on a scale of 1 to 5) and a rating of the magnitude of negative sentiment (on a scale of -1 to -5)’ (Cooper 2017).

There are three flavours of sentiment analysis. First, there is a rule-based approach. This approach applies a sentiment dictionary to the data. Positive and negative words are scored. More sophisticated dictionaries rate affect or emotions. In the study of Jonathan Edwards’ sermons, Michael Keller (Keller 2018) applied the dictionary of affect in language (DAL). DAL consists of a list with words that have been rated by people on the dimensions of pleasantness, activation and imagery (Whissell 2009). Sentiment analysis is well-known because it attaches emotional values to certain words, but a word-based approach is also very limited. It leaves out the larger context of single words. Recent approaches distinghuish between three levels of sentiment analysis: the level of the document, the level of sentences, and the aspect-level which is most fine-grained in its analysis (Liu 2020, 10). Second, supervised approaches are based upon annotated datasets. Supervised approaches are more energy consuming, because they start with the annotation of the documents according to sentiment values. After annotation, machine learning is invoke to develop a model that is able to classify new documents according to the sentiment values that have been used in the annotation. Supervised machine-learning methods usually consist of three phases: 1. the texts of a corpus are manually coded; 2. features that are distinctive for the texts are selected; 3. after dividing the corpus in a training- and a test-dataset the machine learns to classify the coded sentiments; 4. the performance of the classification is tested to evaluate the accuracy and precision of the model (Lei and Liu 2021, 17). The third type of methods are so-called deep-learning algorithms. These methods are often able to apply sentiment analysis to multiple languages and they yield better results than dictionary-based and supervised approaches (Mercha and Benbrahim 2023).[[22]](#footnote-83)

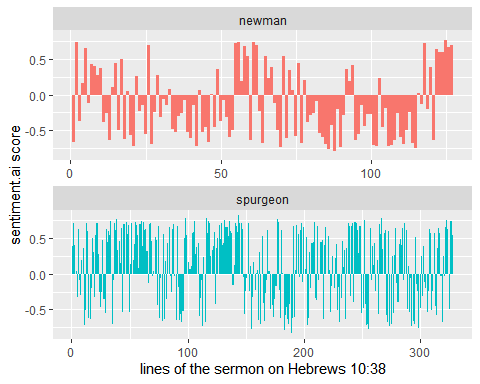
The example belows applies sentiment analysis to the corpus with sermons. It applies the third method, a deep-learning algorithm (‘sentiment.ai’) that has been trained on previous datasets.[[23]](#footnote-84) First, sentiment analysis offers the possibility to visualise the mood of a sermon. This relates to the aspect of *pathos* in rhetoric. Sermons engage the emotions of listeners. Computational analysis makes it possible to study the entire sermon from the perspective of pathos. Further, sentiment analysis is applied to one of the balanced corpora, the corpus with a balanced selection of Biblical texts. To start with the latter, the ‘sentiment.ai’ algorithm is applied to all 112 sermons in the dataset with Newman and Spurgeon’s sermons on the shared 56 biblical texts. The code to calculate the sentiments looks like this:

LOAD library sentiment.ai  
INIT conda for tensorflow environment to run the model  
LOAD balanced corpus  
TRANSFORM balanced corpus into sentences  
CALCULATE sentiment score for each sentence (run model)  
STORE sentiment scores, sentences and metadata in table

After determining the sentiment scores for each sentence in the balanced corpus with biblical texts, the overal sentiment value for each of the 112 sermons is calculated. The measure for this is the median. This means that for each sermon the sentiment value is set to that value of which half of the other sentences has a higher and half of the sentences has a lower value. The median as central tendency measure is thus taken as the measure to determine the overal sentiment value of the document.[[24]](#footnote-85) For a sample of five sermons, this results in the following median-values for Newman and Spurgeon respectively:

## # A tibble: 5 × 3  
## bible newman spurgeon  
## <chr> <dbl> <dbl>  
## 1 Matthew 21:28-30 -0.0775 -0.217  
## 2 Hebrews 12:2 0.0320 -0.328  
## 3 Numbers 32:23 -0.0834 -0.418  
## 4 Romans 10:4 0.420 0.284  
## 5 2 Corinthians 12:11 0.0427 -0.104

Next, the differences are calculated and the most different sermon can be determined. The sermon on Hebrews 10:38 appears to be the sermon in which the overall sentiment of both preachers differs most, Newman uses more negative language than Spurgeon. Based upon this comparison, the sermon on Hebrews 10:38 is a candidate to compare the emotional narratives in both preachers. A visualisation of the sentiment trajectories in both sermons illustrates how sentiment analysis can be used to project the emotional narrative of entire sermons. Further, it provides a tool to compare sermons based upon these plots. The plot shows that Newman has more negatively toned sentences in his sermon, while Spurgeon clearly uses more positive language.

 A brief look at the titles of the sermons shows that Newman has chosen to focus on the second part of the verse in Hebr. 10, namely ‘if any man draw back, My soul shall have no pleasure in him’. The title of Newman’s sermon is: Transgressions and Infirmities. Spurgeon on the other hand preaches on the first part of the verse, ‘Now, the just shall live by faith’. His sermon is titled ‘the vital force’. Already from the choice of the preachers to take a certain perspective on the text, the different tone of the sermons can be explained. This leads to another hypothesis, that the language of preaching interacts with the biblical passage.

Compare the opening line of each sermon, Spurgeon using uplifting language:

See here the germ of the Christian’s life! See, too, how it blooms, blossoms and bears! But observe it is not said the just shall live for his faith, or because of the merit of his believing in God. (Spurgeon, volume 15, sermon 891)

Newman, on the other hand, starts with a warning:

Warnings such as this would not be contained in Scripture, were there no danger of our drawing back, and thereby losing that “life” in God’s presence which faith secures to us. (Newman, volume 5, sermon 14)

Sentiment analysis is among the most widely used techniques in *natural language processing*. It started with lexicon or dictionary based approaches, but the most recent examples are based upon deep-learning algorithms. Using machine learning algorithms in research leads to more advanced computational techniques.

# 6 Advanced computational techniques

From the basic operations in the previous sections, two observations concerning the two Victorian preachers in the dataset emerge. First, the differences in the language of preaching raise the issue of the *personalized* language in sermons. On the other hand, the differences in language seemed to be related to the biblical text (section 5). These two observations can be turned into hypotheses:

*Hypothesis Bible* The language of preaching reflects the language of the biblical passage that is explained in the sermon.

*Hypothesis Preacher* The language of preaching reflects features of the preacher, the preacher’s theology, spirituality and personality.

For both hypotheses positive indicators have been found. Hypothesis ‘Preacher’ is indicated in the frequency tables (section 2). These tables contain various lists of keywords that occur most in the sermons of each preacher. Further, the plot that visualises the keyness of Newman in reference to Spurgeon (section 5) also indicates that the language of both preachers is very different. Hypothesis ‘Bible’ was suggested by the results of sentiment analysis. The emotional tone of the sermons reflected the tone of the biblical text. In the case of Hebrews 10:38 this became clear because the first half of the biblical text was positively preached on by Spurgeon, while Newman focussed on the more negative tones in the second half of the text. In this section more advanced computational methods are used to explore these two hypotheses a bit further.

The distinction between basic and advanced is drawn relies upon the difference between *applying* existing rules and models that have been developed on the one hand and *developing* rules and models on the other hand. The basic operations applied statistical measures (as with keyness) or algorithms (as with sentiment analysis[[25]](#footnote-91)) to the datasets. The advanced methods introduce the development of new models, to fit these models to the data and to validate the accuracy of these models. This section introduces two advanced computational methods, topic modeling and machine learning. Van Atteveldt and others distinghuish between two types of automatic text analysis, unsupervised and supervised modeling. Supervised modeling works with a priori defined categories, while unsupervised modeling the categories are inductively constructed by the computer (Atteveldt, Trilling, and Calderon 2022, chap. 11).

This section introduces both methods. Supervised modeling is used to test the hypothesis ‘Preacher’. The computer ‘learns’ to classify sermon fragments. The classificion model can be used to predict whether a sermon fragment should be classified as a ‘Newman’ or a ‘Spurgeon’ sermon. Unsupervised modeling is introduced to test the hypothesis ‘Bible’. The computer clusters the words of the sermons into ‘topics’. The test will be whether the topics are related to the biblical passages or not.

## Topic Modelling

In the study of religion, topic modeling has been used by various researchers. John Berneau analysed the discourse on pastoral care in 4,054 articles of *Journal of Pastoral Care and Counselling* between 1947-2018 (Bernau 2021). He generated a topic model of 74 topics and visualised the shifts of topics over time. By using topic modelling, Berneau demonstrates that over the decades, the use of traditional religious language disappeared from the journal. Anne Agersnap and Kristoffer Nielbo applied topic modelling to a corpus of 11,955 Danish sermons (Agersnap and Nielbo 2022). Their topic model reconstructed the holidays of the liturgical calendar from the corpus with sermons. Their findings suggest a relationship between the a holiday signal and the content of the sermons. Constantine Boussalis and others studied political speech in religious sermons. Political topics such as economy, war and welfare emerge from the corpus of 110,000 sermons. On the level of pastors, 7 out of 10 pastors delivered at least one sermon with political content (Boussalis, Coan, and Holman 2020).

Topic models have been described as a ‘quantitative tool for a qualitative approach’ (Jacobs and Tschötschel 2019). Agersnap and Nielbo describe the procedure of topic modeling as follows:

A topic model makes the assumption[s] that a document is a distribution over a finite set of latent topics […] [and] that the topics within a text have been chosen by the author before writing the text […] With topic modelling, you try to reverse this process and trace the words back to the topics they came from, and consider the documents in terms of their distribution of topics. (Agersnap and Nielbo 2022, 93)

To test the hypothesis ‘Bible’, the most frequently preached from chapters in Spurgeon’s corpus are selected. From this list, the chapters from the same book are filtered, keeping only the most often preached chapter from the book (see also above, section 5, counting sermons). This means that John 6 is included, but John 14 and John 1 are removed. This results in the following chapters: Psalm 119, Romans 8, John 6, Hebrews 11, Isaiah 53, 2 Corinthians 5. Next, 20 sermons are sampled from each of these chapters. In total 120 sermons are included in the topic modelling excercise.

ADD a document variable to the corpus: chapters  
FIND the most preached chapters in the corpus  
SAMPLE for each of these chapters 20 sermons from the corpus  
TOKENIZE the sampled sermons   
CREATE document-term-matrix   
RUN LDA model for 13 topics (k = 13) on the document-term-matrix,

Topic modelling consists of multiple runs. The variable *k* is used to indicate the amount of topics that need to be generated. After a few tries with different amounts of topics, a model that consists of 13 topics gives a good result. Some chapters have their terms in different topics, but the general tendency is that the chapters are connected to one or two dominant topics. Some biblical chapters have a dominant topic that is shared with other chapters. For instance, the sermons from John 6 are mainly put in topic 13, but topic 13 also contains quite a few sermons on 2 Corinthians 5 and Romans 8. The sermons on Hebrews 11, Isaiah 53, Romans 8, 2 Corinthians 5, and Psalm 119, however, almost entirely make up one particular topic. The sermons and the top 7 words for each topic are projected below:

2 Cor. 5 > topic 3 [reconciliation, stead, ambassadors, wilt, yea, dost, shalt, to-day, wrought, thyself]  
  
Hebrews 11 > topic 8 [staff, pharaoh, pharaoh's, rewarder, rahab, well-pleasing, decided, translated, egyptians, egyptian]  
  
Isaiah 53 > topic 4 [remedy, cures, contemplation, acquaintance, blows, scourged, scourging, application, shrink, lament]  
  
Psalm 119 > topic 9 [statues, petition, teacher, visit, custom, advocate, commandment, catch, uphold, besetting]  
  
Romans 8 > topic 10 [conformed, predestined, first-born, conformity, heirship, undefiled, knock, amidst, relation, firstborn]

In conclusion, topic modelling suggests a positive evaluation of the hypothesis ‘Bible’. Sermons on the same biblical text are grouped together in one or two topics. This indicates that the terms that are used in these sermons are specifically connected to the text from which the preacher preaches.

## Machine Learning: Classification

Hypothesis ‘Preacher’ investigates the relationship between the language of the sermon and the preacher. It states that the language of preaching reflects features of the preacher. Machine learning is used to test this model. The machine learning task that does the job is a so-called classification task. The task is also relatively easy, the computer has to decide between two classes. Does a sermon fragment fits Newman’s style of preaching? If not, the computer decides that it probably closer to Spurgeon’s preaching style. Though this task is very minimal and has multiple limitations, it is straightforward and suited for the illustrative purposes in this chapter.

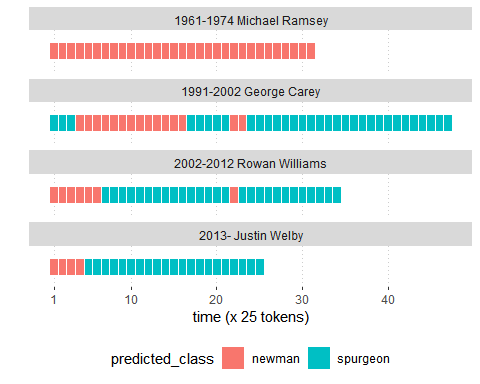
LOAD corpora spurgeon & newman  
SET sample length to 100  
SET chunksize and window to 200 and 10 respectively  
FUNCTION create chunks with argument sermon-id  
 TOKENIZE sermon-id: remove punctuation, numbers, and symbols  
 SET tokens to lowercase  
 REMOVE stopwords-list  
 DIVIDE tokens into chunks of 200 (chunksize) with an overlap of 10 (window)  
 REMOVE the chunks that are shorter than 200 (chunksize)  
SAMPLE 100 (sample length) sermons from corpora  
ADD document variables to sample corpus: preacher and unique sermon-id  
REMOVE all document variables, except preacher and sermon-id  
OUTPUT function 'create chunks' to all sermon-ids to table with sermon-chunks  
BALANCE chunks by drawing an equal sample of all chunks from each preacher  
CREATE document-term-matrix from the chunks  
DIVIDE matrix into train (80%) and test set (20%)  
BUILD model on the train set, use naive Bayes (quanteda: textmodel\_nb)

Finally, the model is evaluated. The first evaluation is on the test set. If the test set performs well, the model is tested against other sermons by Spurgeon and Newman. Models are evaluated by creating a confusion matrix. This matrix projects the positive and negative predicted classes. Important are false positives (Newman is predicted, while the fragment is actually from Spurgeon) and false negatives (a fragment is falsely attributed to Spurgeon). Measurements like accuracy, precision, recall and F1 are used to determine the performance of the model. Since all measurements are around 0.97 (on a scale of 0 to 1), the model performs very well on the test set. A similar result is obtained when the model is tested against sermons that were not included in the original sample.

Confusion Matrix and Statistics  
  
 predicted\_class  
actual\_class newman spurgeon  
 newman 189 7  
 spurgeon 4 226  
   
 Accuracy : 0.9742   
 95% CI : (0.9543, 0.987)  
 No Information Rate : 0.5469   
 P-Value [Acc > NIR] : <2e-16   
   
 Kappa : 0.948   
   
 Mcnemar's Test P-Value : 0.5465   
   
 Sensitivity : 0.9793   
 Specificity : 0.9700   
 Pos Pred Value : 0.9643   
 Neg Pred Value : 0.9826   
 Precision : 0.9643   
 Recall : 0.9793   
 F1 : 0.9717   
 Prevalence : 0.4531   
 Detection Rate : 0.4437   
 Detection Prevalence : 0.4601   
 Balanced Accuracy : 0.9746   
   
 'Positive' Class : newman

Because the measures indicate that the algorithm performs very well in distinghuising between Newman and Spurgeon, the credibility of hypothesis ‘Preacher’ increases. Given a certain sermon fragment, the model predicts the preacher. The language of sermons indeed very much reflects a preacher.

Finally, the proof of the pudding is in the eating. Can the model be used against other preachers? Here the limitations of the model must be taken into account. The model predicts whether a sermon segment belongs to Newman. If not, it assigns it so Spurgeon. The binary model (Newman or not-Newman) could be used to predict of a new sermon fragment would fit the models features. Newman being a 19th Century Anglican preacher - at least before 1843 - should therefore be compared to an other Anglican preacher. The model is tested against four sermons held at Lambeth conferences, from the current (Welby) and a previous Archbishops (Williams, Carey and Ramsey) of the Anglican church.

 Besides the differences in length between the sermons, the plot shows that Ramsey’s sermon during the Lambeth conference (1968) reflects Newman’s style of preaching. At the end of the 20th Century, so it seems, less sermon fragments fit Newman’s style and are thus classified as similar to Spurgeon. Because of the binary classification the computer is forced to put the sermon fragments in one of both categories. A more refined model would be able to distinghuish with more nuances. For this, however, a much larger dataset that includes multiple preachers is needed.

Finally, the test of the model suggests a new hypothesis. Hypothesis ‘Preacher’ may be too much focussed upon the subject of the individual preacher. It is more likely that the language of groups of preachers have certain features in common. This could lead to a new hypothesis:

*Hypothesis Tradition* The language of preaching reflects features of a shared preaching tradition.

# 7 Implications for homiletical research and theory

The previous section explored two homiletic hypotheses and generated a third. The language of preaching is somehow connected to Bible, person and tradition. These insights have not been formulated on the basis of a deductive, pre-empirical, theological theory, but they emerge from analysis of empirical data.

Computational methods enrich the toolbox of the empirical homiletician, in scope as well as in interconnectedness and innovation. Computational tools have an enormous *scope*. It is possible to study large datasets with sermons in one project. For instance, the classifier that is able to recognize Newman’s style of preaching, was based upon data from 200 sermons. They make it possible to pursue research questions that are impossible to answer due to the limitations of manual coding and analysis. These tools are also *interconnected*. Word frequencies, document-term-matrices, stemming and labelling come together in complex pipelines in which several methods are combined, only limited by the creativity and technological skills of the researcher. Computational tools are not only innovative ways for data analysis, they are also part of an ongoing process of technological *innovation*. Sentiment analysis is an interesting example. The technique has been around from the early days of computational research, but the rise of artificial intelligence and deep-learning algorithms, added innovation that goes far beyond a lexicon-approach.

This chapter scratched the surface of computational analysis. Many approaches, such as parts-of-speech (POS) tagging, named-entity recognition (NER ), topic modelling, or the study of rhetorical figures (Harris et al. 2018) and intertextuality have not been dealt with. Intertextuality, for example, is studied by Arévalo who use computational methods to identify biblical references in Medieval religious literature (Arévalo 2021). Further, this chapter has been written by a theologian with a strong interest in empirical methods and computational approaches. It’s main impulse is to wet the appetite to move into territories that are fundamentally interdisiplinary of nature. But, as Hans van der Ven argued in his framework for empirical theology, the theologian needs to adopt an intradisciplinary attitude, to borrow concepts, methods and techniques from other fields, to adapt them to the specific theological questions and to integrate them into research projects (van der Ven 1993, 101). The features of computational research, its scope, interconnectedness and continuing technological innovation, make it necessary for theologians to develop their statistical and data science competences or, and even better, to create research teams with computational scientists.

Three limitations close this introduction in computational sermon analysis. First, theologians should be aware of the critical voices within digital humanities. Augustine Farinola, for instance, challenges the distinction of ‘close reading’ versus ‘distant reading’ which forms the theoretical basis for computational methods and argues for human agency in acts of interpretation (Farinola 2023). Next, this chapter started with building a corpus from sermons available in the public domain. This emphases the need for large databases of sermons. The nature and organisation of data raise epistemological issues and generate practical challenges(Wildman, Diallo, and Shults 2021, 139–45). Finally, Mathew Gillings and others have argued that working with corpora should not move the researcher away from discourse (Gillings, Mautner, and Baker 2023, 47–49). For homiletics this means, that sermon analysis should always remain connected to the practice of preaching.

# References

Agersnap, Anne. 2021. “Collective Testimonies to Christianity and Time.” PhD thesis, Aarhus: Aarhus University.

Agersnap, Anne, Ross Deans Kristensen-McLachlan, Kirstine Helboe Johansen, Uffe Schjødt, and Kristoffer Laigaard Nielbo. 2020. “Sermons as Data: Introducing a Corpus of 11,955 Danish Sermons.” *Journal of Cultural Analytics* 5 (2): 18238. <https://doi.org/10.22148/001c.18238>.

Agersnap, Anne, and Kristoffer L. Nielbo. 2022. “‘What Is Love …?’: A Study of Thematic Constructions in 11.955 Danish Sermons.” *International Journal of Homiletics* 5 (1): 86–116. <https://doi.org/10.21827/ijh.5.1.86-116>.

Arévalo, Enrique Manjavacas. 2021. “Computational Approaches to Intertextuality.” PhD thesis, Antwerpen: University of Antwerp.

Atteveldt, Wouter van, Damian Trilling, and Carlos Arcila Calderon. 2022. *Computational Analysis of Communication*. 1st edition. Hoboken, NJ: Wiley-Blackwell.

Ballod, Georg, and Matthias Ballod. 2000. *Predigthilfen aus dem Rechner? Computerunterstützte Predigtanalysen*. Marnheim: Bern Verlag.

Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2018. “Quanteda: An R Package for the Quantitative Analysis of Textual Data.” *Journal of Open Source Software* 3 (30): 774. <https://doi.org/10.21105/joss.00774>.

Bernau, John A. 2021. “From Christ to Compassion: The Changing Language of Pastoral Care.” *Journal for the Scientific Study of Religion* 60 (2): 362–81. <https://doi.org/10.1111/jssr.12711>.

Black, Michael L. 2016. “The World Wide Web as Complex Data Set: Expanding the Digital Humanities into the Twentieth Century and Beyond Through Internet Research.” *International Journal of Humanities and Arts Computing* 10 (1): 95–109. <https://doi.org/10.3366/ijhac.2016.0162>.

Boehmke, Bradley, and Brandon Greenwell. 2019. *Hands-On Machine Learning with R*. Taylor & Francis.

Boss, Robert L. 2020. “Visualizing Jonathan Edwards.” *Polish Journal for American Studies* 14: 187–96, 269.

Boussalis, Constantine, Travis G. Coan, and Mirya R. Holman. 2020. “Political Speech in Religious Sermons.” *Politics and Religion*, 1–28. <https://doi.org/10.1017/S1755048320000334>.

Center, Pew Research, Suite 800 Washington, and DC 20036 USA202-419-4300 | Main202-419-4349 | Fax202-419-4372 | Media Inquiries. 2019. “The Digital Pulpit. A Nationwide Analysis of Online Sermons.” *Pew Research Center’s Religion & Public Life Project*.

Chollet, Francois, and J. J. Allaire. 2018. *Deep Learning with R*. 1 edition. Shelter Island, NY: Manning Publications.

Cooper, Anthony-Paul. 2017. “Assessing the Possible Relationship Between the Sentiment of Church-related Tweets and Church Growth.” *Studies in Religion/Sciences Religieuses* 46 (1): 37–49. <https://doi.org/10.1177/0008429816664215>.

Drucker, Johanna. 2021. *The Digital Humanities Coursebook: An Introduction to Digital Methods for Research and Scholarship*. London: Routledge. <https://doi.org/10.4324/9781003106531>.

Edwards, O. C., Jr. 2004. *A History of Preaching*. Nashville: Abingdon Press.

Engemann, Wilfried. 2019. *Homiletics. Principles and Patterns of Reasoning*. Berlin / Boston: Walter de Gruyter.

Farinola, Augustine. 2023. “Hermeneutical Postphenomenology: Computational Tools and the Lure of Objectivity.” *Digital Scholarship in the Humanities*, January, fqac074. <https://doi.org/10.1093/llc/fqac074>.

Gibson, William. 2012. “The British Sermon 16891901: Quantities, Performance, and Culture.” In *The Oxford Handbook of the British Sermon 1689-1901*, edited by Keith A. Francis, William Gibson, John Morgan-Guy, Bob Tennant, and Robert H. Ellison, 0. Oxford University Press. <https://doi.org/10.1093/oxfordhb/9780199583591.013.0001>.

Gillings, Mathew, Gerlinde Mautner, and Paul Baker. 2023. *Corpus-Assisted Discourse Studies*. 1st ed. Cambridge University Press. <https://doi.org/10.1017/9781009168144>.

Gries, Stefan Th. 2013. *Statistics for Linguistics with R: A Practical Introduction*. *Statistics for Linguistics with R*. De Gruyter Mouton.

———. 2016. *Quantitative Corpus Linguistics with R: A Practical Introduction*. 2nd ed. New York: Routledge. <https://doi.org/10.4324/9781315746210>.

Grolemund, Garrett, and Hadley Wickham. 2017. *R for Data Science*.

Grözinger, Albrecht. 2008. *Homiletik. Lehrbuch Praktische Theologie*. Gütersloh: Gütersloher Verlagshaus.

Hardie, Andrew, and Tony McEnery, eds. 2011. “What Is Corpus Linguistics?” In *Corpus Linguistics: Method, Theory and Practice*, 1–24. Cambridge Textbooks in Linguistics. Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511981395.002>.

Harris, Randy Allen, Chrysanne Di Marco, Sebastian Ruan, and Cliff O’Reilly. 2018. “An Annotation Scheme for Rhetorical Figures.” *Argument & Computation* 9 (2): 155–75. <https://doi.org/10.3233/AAC-180037>.

Homiletischen Arbeitsgruppe. 1973. *Die Predigt Bei Taufe, Trauung Und Begräbnis. Inhalt, Wirkung Und Funktion. Eine Contentanalyse*. Mnchen: Kaiser.

Jacobs, Thomas, and Robin Tschötschel. 2019. “Topic Models Meet Discourse Analysis: A Quantitative Tool for a Qualitative Approach.” *International Journal of Social Research Methodology* 22 (5): 469–85. <https://doi.org/10.1080/13645579.2019.1576317>.

Jockers, Matthew L. 2014. *Text Analysis with R for Students of Literature*. Quantitative Methods in the Humanities and Social Sciences. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-03164-4>.

Jockers, Matthew L., and Ted Underwood. 2015. “Text-Mining the Humanities.” In *A New Companion to Digital Humanities*, 291–306. John Wiley & Sons, Ltd. <https://doi.org/10.1002/9781118680605.ch20>.

Karcher, Stefan. 2020. “Praktische Theologie und Digital Humanities.” *Verkündigung und Forschung* 65 (2): 132–42. <https://doi.org/10.14315/vf-2020-650209>.

Keller, M. S. 2018. “Experiencing God in Words: Rhetoric, Logic, Imaginative Language, and Emotion in Jonathan Edwards’ Sermons: A Computational Analysis.” PhD thesis, Amsterdam: VU.

Khurana, Diksha, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2023. “Natural Language Processing: State of the Art, Current Trends and Challenges.” *Multimedia Tools and Applications* 82 (3): 3713–44. <https://doi.org/10.1007/s11042-022-13428-4>.

Lei, Lei, and Dilin Liu. 2021. “Conducting Sentiment Analysis.” *Elements in Corpus Linguistics*, August. <https://doi.org/10.1017/9781108909679>.

Lischer, Richard. 2002. *The Company of Preachers. Wisdom on Preaching, Augustine to the Present*. Grand Rapids: W.B. Eerdmans Pub. Co.

Liu, Bing. 2020. *Sentiment analysis: mining opinions, sentiments, and emotions*. Second edition. Studies in natural language processing. Cambridge, United Kingdom ; Cambridge University Press.

Malmström, Hans. 2015a. “The "Other" Voice in Preaching: Intertextual Form and Function in Contemporary English Sermonic Discourse.” *Journal of Communication & Religion* 38 (2): 80–99.

———. 2015b. “What Is Your Darkness?: An Empirical Study of Interrogative Practices in Sermonic Discourse.” *International Journal of Practical Theology* 19 (2): 247–70. <https://doi.org/10.1515/ijpt-2014-0019>.

McGillivray, Barbara, Thierry Poibeau, and Pablo Ruiz Fabo. 2020. “Digital Humanities and Natural Language Processing: Je t’aime... Moi Non Plus.” *Digital Humanities Quarterly* 014 (2).

Mercha, El Mahdi, and Houda Benbrahim. 2023. “Machine Learning and Deep Learning for Sentiment Analysis Across Languages: A Survey.” *Neurocomputing* 531 (April): 195–216. <https://doi.org/10.1016/j.neucom.2023.02.015>.

Moehn, Wilhelmus H. Th. 2001. *"God Calls Us to His Service" : The Relation Between God and His Audience in Calvin’s Sermons on Acts*. Librairie Droz.

Old, Hughes Oliphant. 2007. *The Reading and Preaching of the Scriptures in the Worship of the Christian Church. Volume 6. The Modern Age*. Grand Rapids: William B Eerdmans Publishing Co.

Ong, Walter J. 2002. *Orality and Literacy*. Second. London: Routledge.

Phillips, Peter. 2020. “Defining Digital Theology.” *Cursor\_ Zeitschrift für Explorative Theologie*, November. <https://doi.org/10.21428/fb61f6aa.07b246a7>.

Pieterse, H. J. C. 1995. *Desmond Tutu’s Message a Qualitative Analysis*. Theologie & Empirie 24. Kampen: Kok.

Pieterse, Hendrik J. C. 2020. “A Short History of Empirical Homiletics in South Africa.” *Stellenbosch Theological Journal* 6 (2): 345–64. <https://doi.org/10.17570/stj.2020.v6n2.a15>.

Pleizier, Theo. 2022. “Impact of Religious Speech. Theological and Anthropological Considerations.” In *Dynamics of Speaking and Doing Religion*, edited by Baktygul Tulebaeva and Deepak Ojha, 19–33. Ressourcenkulturen 18. Tübingen: University Press.

Pleizier, Theo, and Henk de Roest. 2023. “6.1.4 Netherlands. Enhanced Streaming and the Rediscovery of Phone Calls. Experiences of Pastors During the First COVID-19 Lockdown in The Netherlands.” In *Churches Online in Times of Corona. The CONTOC Study: Empirical Insights, Interpretations, and Perspectives*, edited by Thomas Schlag, Ilona Nord, Wolfgang Beck, Arnd Bünker, Georg Lämmlin, Sabrina Müller, Johann Pock, and Martin Rothgangel, in press. Springer VS.

Pustejovsky, J., and Amber Stubbs. 2013. *Natural Language Annotation for Machine Learning*. Sebastopol, CA: O’Reilly Media.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Manual. Vienna, Austria: R Foundation for Statistical Computing.

Robinson, Denis. 2009. “Preaching.” In *The Cambridge Companion to John Henry Newman*, edited by Ian Ker and Terrence Merrigan, 1st ed., 241–54. Cambridge University Press. <https://doi.org/10.1017/CCOL9780521871860.012>.

Scott, Geoffrey. 2012. “Sermons in British Catholicism to the Restoration of the Hierarchy (16891850).” In *The Oxford Handbook of the British Sermon 1689-1901*, edited by Keith A. Francis, William Gibson, John Morgan-Guy, Bob Tennant, and Robert H. Ellison, 0. Oxford University Press. <https://doi.org/10.1093/oxfordhb/9780199583591.013.0008>.

Silge, Julia, and David Robinson. 2017. *Text Mining with R. A Tidy Approach*. O’Reilly.

Stefanowitsch, Anatol. 2020. *Corpus Linguistics*. *Language Science Press*. Language Science Press. <https://doi.org/10.5281/zenodo.3735822>.

Stubbs, Michael. 2010. “Three concepts of keywords.” In *Keyness in texts*, edited by Marina Bondi and Mike Scott. Studies in corpus linguistics, v. 41. Amsterdam ; John Benjamins Pub. Co.

Sutinen, Erkki, and Anthony-Paul Cooper. 2021. *Digital Theology: A Computer Science Perspective*. Emerald Group Publishing.

van Atteveldt, Wouter, Mariken A. C. G. van der Velden, and Mark Boukes. 2021. “The Validity of Sentiment Analysis: Comparing Manual Annotation, Crowd-Coding, Dictionary Approaches, and Machine Learning Algorithms.” *Communication Methods and Measures* 15 (2): 121–40. <https://doi.org/10.1080/19312458.2020.1869198>.

van der Ven, J. A. 1993. *Practical Theology. An Empirical Approach*. Kampen: Kok Pharos.

Whissell, Cynthia. 2009. “Using the Revised Dictionary of Affect in Language to Quantify the Emotional Undertones of Samples of Natural Language.” *Psychological Reports* 105 (2): 509–21.

Wildman, Wesley J., Saikou Y. Diallo, and F. LeRon Shults. 2021. “Advanced Computational Methods.” In *The Routledge Handbook of Research Methods in the Study of Religion*, 2nd ed., 136–53. London: Routledge. <https://doi.org/10.4324/9781003222491-10>.

Wymer, Kathryn C. 2021. *Introduction to Digital Humanities: Enhancing Scholarship with the Use of Technology*. New York: Routledge. <https://doi.org/10.4324/9781003149378>.

1. The streaming services <https://kerkomroep.nl> and <https://kerkdienstgemist.nl> (accessed 28 November 2023) provide access to worship services of around 3,000 churches . See for these services further (Pleizier and Roest 2023). The estimation of 3,000-5,000 sermons per week acknowledges that a considerable amount of Dutch churches offer two services a Sunday. [↑](#footnote-ref-20)
2. For the construction of the corpus, see section 4. Since the amount of sermons in each corpus is very different (N = 3516 for Spurgeon; N = 217 for Newman), a sample is drawn from both corpuses, N = 150 for each preacher. Two other balanced corpora are created: the same biblical texts (56 texts for each preacher) and the same biblical chapters (148 chapters for each preacher). See the list of the biblical texts in the balanced corpus: XXX. [↑](#footnote-ref-24)
3. Due to digital availability, for this case study only Newman’s Anglican sermons are included: 8 volumes of his *Parochial and Plain Sermons* and a volume called *Sermons on Subjects of the Day* which includes his farewell sermon entitled ‘The Parting of Friends’. The sermons are preached between 1825 and 1843. On Newman’s Catholic sermons, see (Scott 2012). [↑](#footnote-ref-26)
4. <https://ccel.org/>. [↑](#footnote-ref-27)
5. <https://digitalcollections.newmanstudies.org/>. The digital collection contains digitized books of Newman’s sermons, all files contain photographs of the books with handwritten sermons. The sermons used for the analysis in this chapter are downloaded from the freely available fulltext repository, all files in plaintext format: <https://www.newmanreader.org/>. [↑](#footnote-ref-29)
6. Spurgeon’s sermons are available at <https://ccel.org>. The database started in 1997 with a CD-ROM containing classical Christian literature, which included 63 volumes of sermons by Spurgeon. [↑](#footnote-ref-32)
7. In recent years more of his sermons have been rediscovered, published in a series *The Lost Sermons* (7 volumes between 2017 and 2022). This collection consists of photographs of Spurgeon’s handwritten sermons. [↑](#footnote-ref-34)
8. Web scraping is a method used in data science and digital humanities to extract data from the internet. Cf. (Black 2016). Black argues that the Internet could be used as a low-cost and resource-light source for research data. The scripts used to scrape the sermons from Newman and Spurgeon are published at the Github repository that belongs to this article, see further below, section XXX. He also discusses the legal aspects of webscraping for academic purposes. [↑](#footnote-ref-35)
9. The *Parochial and Plain Sermons* from Newman were scraped from <https://newmanreader.org/>. The full digital collection of Newman’s works can be found at NINS, <https://digitalcollections.newmanstudies.org/>. [↑](#footnote-ref-36)
10. Effectively, downloading the sermons from Spurgeon was more difficult, because each volume with sermons had its own webpage that contained a menu with different formats for downloads. For example, the url for volume 60 is <https://ccel.org/ccel/s/spurgeon/sermons60/cache/sermons60.txt>. [↑](#footnote-ref-39)
11. <http://corpora.lancs.ac.uk/lancsbox/>; <https://www.laurenceanthony.net/software/antconc/>. [↑](#footnote-ref-41)
12. <https://www.python.org/>. [↑](#footnote-ref-44)
13. <https://cran.r-project.org/>. R is often used in a integrated develop environment (IDE) RStudio: <https://posit.co/products/open-source/rstudio/>. [↑](#footnote-ref-46)
14. A good starting point to start data analysis with R is *R for Data Science*, a book that can be found online: <https://r4ds.had.co.nz/>. An introduction to R specifically focussing upon quantitative linguistics - with a strong orientation to statistics - is (Gries 2016, 2013). An accessible introduction in using both R and Python in digital humanties with emphasis on communication, see (Atteveldt, Trilling, and Calderon 2022). [↑](#footnote-ref-49)
15. All R scripts written for this chapter are publicly available at Github: <https://github.com/ttjpleizier/computational-sermon-analysis>. An OSF (open science framework) project is available at <https://osf.io/4a2yz/>. They can be used to reproduce the examples in this chapter. Each script starts with the necessary libraries (packages) that need to be installed within R. [↑](#footnote-ref-51)
16. For the techniques applied in this chapter the package ‘quanteda’ was selected. *Quanteda* is a toolbox for quantitative analysis of textual data (Benoit et al. 2018). The authors of *Quanteda* call it an ‘ecosystem of open source text analysis software’. An alternative ecosystem is *Tidytext*, which is illustrated in (Silge and Robinson 2017). Finally, there is so-called ‘Base-R’, which relies on basic programming functions in R. The books by Jockers and Gries explain textmining techniques and methods of quantitative linguistics in ‘Base-R’ (Jockers 2014; Gries 2016). [↑](#footnote-ref-54)
17. For the technique of pseudocode, see (Gries 2016, 175). Gries refers to John Dalbey who developed the standard for pseudocode in 2003. Retrieved November 26, 2023, from <http://users.csc.calpoly.edu/~jdalbey/SWE/pdl_std.html>. [↑](#footnote-ref-55)
18. In the case of Spurgeon, there is more work to do. The code that fetches the volumes results in 63 textfiles. Each textfiles contains a large number of sermons. These sermons should be located and extracted. [↑](#footnote-ref-57)
19. See for a detailed workflow description with pipelines and organisation of project-folders: <https://tilburgsciencehub.com/tutorials/reproducible-research-and-automation/principles-of-project-setup-and-workflow-management/directories/> (accessed 1-3-2023) See also the TIER protocol: <https://www.projecttier.org/tier-protocol/protocol-4-0/>. [↑](#footnote-ref-59)
20. Mark Barnes created this index in 2012, as found on the forum of Logos Bible Software. He offered this index for users of Logos, but the index could easily be used to clean the downloaded data, for instance to find the correct titles of the sermons and to add consistent references to the bible texts. Barnes’ index can be found on: <https://community.logos.com/forums/t/44342.aspx> (accessed 10-4-2023). [↑](#footnote-ref-62)
21. See for the table with sources for the reconstruction of the dates: <https://www.newmanreader.org/controversies/guides/dates.html>, accessed 4-12-2023. [↑](#footnote-ref-65)
22. Van Atteveldt and others stress the importance of human performance in validating automatic text analysis, see (van Atteveldt, van der Velden, and Boukes 2021). [↑](#footnote-ref-83)
23. The algorithm that is used is ‘sentiment.ai’. It is both open source, trained on embeddings and is written to be used in R. [↑](#footnote-ref-84)
24. The method proposed here is very simple: determine sentiments for each sentences and calculate the median for the entire document. For illustrative purposes this could suffice, but for a more sophisticated approach of determining document sentiments, see (Liu 2020, chap. 3). [↑](#footnote-ref-85)
25. It is important to note that some versions of sentiment analysis also use supervised modelling (Atteveldt, Trilling, and Calderon 2022, chaps. 11.4;@lei2021,chapter 2.3). [↑](#footnote-ref-91)