

Enhancing digital text collections with detailed metadata to improve retrieval

by

Liezl Hilde Ball

24020461

Submitted in partial fulfilment of the requirements for the degree
D.Phil. in Information Science

in the

Department of Information Science

Faculty of Engineering, Built Environment and Information
Technology

University of Pretoria

Supervisor:

Prof. T.J.D. Bothma

October 2020

DECLARATION

I, Liezl Ball, declare that this is my own, original work. Where other work was used, due acknowledgement was given. This thesis has not been submitted by me to another tertiary institution for any degree.

LH Ball _____

Signature

28 October 2020 _____

Date

ACKNOWLEDGEMENTS

I would like to express my gratitude:

- to my supervisor, Prof. Theo Bothma, for his excellent advice and persistent patience over the course of this study. I am most grateful for his guidance;
- to all my family and friends, for their support and encouragement throughout this journey;
- to my father, who said we must have a big celebration when I have finished with this PhD;
- to my mother, who always listened patiently, motivated gently and still inspires me;
- to my brother, who said I must enjoy this, because when I have finished, I will have to supervise other people's work;
- to my sister, for her vibrant nature and love towards me;
- to Linda and Michael, whose friendship has sustained me;
- to Peter, my husband and friend, without whose help, support and superior programming skills I would not have been able to complete this study. I hereby acknowledge that he did the programming of the prototype, as based on the specifications from my research. Thank you for *inkling*;
- to Sharon, my editor, who is both unbelievably meticulous and exceptionally gentle. I appreciate your care with my thesis. Any errors left in this manuscript are my own oversight;
- above all, to God, whose Spirit hovered over the waters and the chaos in the beginning, who then created the heavens and the earth, and lovingly invites us to create.

Dedicated to Leann and Brian

“There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle.” – Attributed to Albert Einstein

ABSTRACT

Digital text collections are increasingly important, as they enable researchers to explore new ways of interacting with texts through the use of technology. Various tools have been developed to facilitate exploring and searching in text collections at a fairly low level of granularity. Ideally, it should be possible to filter the results at a greater level of granularity to retrieve only specific instances in which the researcher is interested.

The aim of this study was to investigate to what extent detailed metadata could be used to enhance texts in order to improve retrieval. To do this, the researcher had to identify metadata that could be useful to filter according to and find ways in which these metadata can be applied to or encoded in texts. The researcher also had to evaluate existing tools to determine to what extent current tools support retrieval on a fine-grained level. After identifying useful metadata and reviewing existing tools, the researcher could suggest a metadata framework that could be used to encode texts on a detailed level. Metadata in five different categories were used, namely morphological, syntactic, semantic, functional and bibliographic. A further contribution in this metadata framework was the addition of in-text bibliographic metadata, to use where sections in a text have different properties than those in the main text.

The suggested framework had to be tested to determine if retrieval was indeed improved. In order to do so, a selection of texts was encoded with the suggested framework and a prototype was developed to test the retrieval. The prototype receives the encoded texts and stores the information in a database. A graphical user interface was developed to enable searching in the database in an easy and intuitive manner.

The prototype demonstrates that it is possible to search for words or phrases with specific properties when detailed metadata are applied to texts. The fine-grained metadata from five different categories enable retrieval on a greater level of granularity and specificity. It is therefore recommended that detailed metadata are used to encode texts in order to improve retrieval in digital text collections.

Keywords: metadata, digital humanities, digital text collections, retrieval, encoding

Table of content

1. Introduction	1
1.1. Background.....	1
1.2. Research problem.....	3
1.3. Clarification of concepts	5
1.3.1. Metadata.....	5
1.3.2. Markup.....	7
1.3.3. Annotations	8
1.3.4. Encoding.....	8
1.3.5. Summary	9
1.4. Research methodology	9
1.5. Limitations to the study.....	10
1.6. Outline of chapters	11
2. Review of relevant literature and software.....	12
2.1. Introduction	12
2.2. Digitisation	13
2.3. Digital humanities.....	16
2.4. Limitations when searching in text collections	22
2.5. Metadata to enhance retrieval of words or phrases	25
2.5.1. Morphological level	26
2.5.2. Syntactic level	35
2.5.3. Semantic level.....	41
2.5.4. Functional level	45
2.5.5. Bibliographic level	58
2.6. Tools and projects.....	61
2.6.1. Google Books Ngram Viewer	62
2.6.2. HathiTrust Bookworm.....	63
2.6.3. Perseus Project.....	65
2.6.4. Voyant Tools	66
2.6.5. TXM	67
2.6.6. BNCweb (CQP-edition)	68
2.6.7. BYU Corpora.....	69
2.6.8. Other projects	70
2.7. Evaluation of existing tools.....	80
2.7.1. Interface design	80
2.7.2. Metadata.....	85
2.7.3. Search options	102

2.7.4.	Filtering	114
2.7.5.	Search results	120
2.7.6.	Complexity of use.....	132
2.7.7.	Help files	135
2.7.8.	Corpus design.....	137
2.8.	Conclusion	138
3.	Research methodology	147
3.1.	Introduction	147
3.2.	Grounded theory study.....	148
3.3.	Phenomenological study	149
3.4.	Literature review.....	149
3.5.	Action research	150
3.6.	Case study research	151
3.7.	Purposive sampling.....	152
3.8.	Prototyping.....	153
3.9.	Heuristic evaluation.....	156
3.10.	Qualitative and quantitative research.....	157
3.11.	Credibility of the study	158
3.12.	Ethical considerations.....	158
3.13.	Conclusion.....	159
4.	Suggested encoding for texts	160
4.1.	Morphological level	160
4.2.	Syntactic level	163
4.3.	Semantic level.....	165
4.4.	Functional level	167
4.5.	Bibliographic level	175
4.6.	Combined example	181
4.7.	Conclusion	188
5.	Encoding of sample texts to improve retrieval	189
5.1.	Introduction	189
5.2.	Texts selected for encoding	189
5.3.	Encoding process.....	193
5.3.1.	Morphological level	193
5.3.2.	Syntactic level	193
5.3.3.	Semantic level.....	194
5.3.4.	Functional level	194
5.3.5.	Bibliographic level	194

5.4.	Examples	195
5.5.	Conclusion	197
6.	Prototype to test retrieval of encoded texts.....	198
6.1.	Introduction	198
6.2.	Implementation	198
6.3.	Searching in the tool	205
6.3.1.	Overview of the tool	205
6.3.2.	Home page	205
6.3.3.	The graphical user interface.....	206
6.3.4.	Simple search	207
6.3.5.	Results.....	208
6.3.6.	Truncation.....	209
6.3.7.	Searching for inflected forms.....	210
6.3.8.	Part-of-speech category.....	212
6.3.9.	Multiple words	213
6.3.10.	Dependencies	215
6.3.11.	Search according to meaning (semantics).....	218
6.3.12.	Search according to functional properties.....	220
6.3.13.	Searching according to bibliographic properties.....	229
6.3.14.	Combining search options	235
6.3.15.	Searching using a query language.....	236
6.4.	Conclusion	238
7.	Automated processing of text.....	243
7.1.	Introduction	243
7.2.	Encoding texts with existing tools.....	244
7.2.1.	Morphological encoding	244
7.2.2.	Syntactic encoding.....	258
7.2.3.	Semantic encoding.....	271
7.3.	Conclusion	276
8.	Conclusion	280
8.1.	Introduction	280
8.2.	Answering the research question and sub-questions	280
8.2.1.	Metadata elements to describe and encode texts.....	281
8.2.2.	Characteristics of tools that are currently used for the retrieval of words from digital text collections	281
8.2.3.	The extent to which current tools allow for the retrieval of words or phrases with specific properties from text collections	282

8.2.4.	Encoding that will allow for retrieval on a greater level of granularity and specificity	282
8.2.5.	Prototype to retrieve words or phrases with specific properties from texts encoded with detailed metadata	283
8.2.6.	Automated encoding of texts	284
8.2.7.	A decision support system as contribution to theory development.....	284
8.2.8.	Recommendations and future research for the encoding of text collections to improve retrieval	291
8.2.9.	The extent to which retrieval of words or phrases with specific properties from digital text collections can be improved by detailed encoding	291
8.3.	Recommendations and future research.....	293
8.3.1.	Encode texts with detailed metadata to enhance retrieval	293
8.3.2.	Identify more metadata useful for encoding	293
8.3.3.	Develop tools that can utilise metadata for improved retrieval	294
8.3.4.	Accommodate laypersons and advanced users when designing tools for retrieval	294
8.3.5.	Improve tools used for automated encoding.....	295
8.3.6.	Explore the scalability of the solution.....	295
8.4.	Conclusion	295
9.	References.....	296

List of figures

Figure 1	An example of the Google Books Ngram Viewer (Google Books Ngram Viewer Info, 2020)	2
Figure 2	Filtering in HathiTrust+Bookworm.....	3
Figure 3	Example of tokenisation (https://nlp.stanford.edu/software/tokenizer.html)	32
Figure 4	A parse tree showing phrase structure grammar	36
Figure 5	A sentence analysed according to dependency grammar (Jurafsky & Martin, 2017).....	37
Figure 6	The word fall in WordNet	43
Figure 7	Example of a TEI document (Van den Branden et al., 2017)	47
Figure 8	A letter by Walt Whitman (The Walt Whitman Archive, n.d.)	48
Figure 9	The letter encoded in TEI	48
Figure 10	The digital representation of the letter	49
Figure 11	Demonstration of Stanford Named Entity Tagger	57
Figure 12	Google Translate.....	57
Figure 13	A search in Google Books Ngram Viewer.....	62
Figure 14	An example from the HathiTrust+Bookworm	65
Figure 15	Against Cataline by Cicero	66
Figure 16	Voyant Tools	66
Figure 17	The GRAAL corpus in TXM	68
Figure 18	Results returned in the BNCweb (CQP-edition)	69
Figure 19	XML query in the XAIRA software (University of Oxford IT Services, n.d.)...71	
Figure 20	The interface for CQPweb with the ability to restrict according to texts.....72	

Figure 21 Corpora available on CQPweb	72
Figure 22 Corpus metadata for Early English Books Online (V3) in CQPweb.....	73
Figure 23 An error message in CQPweb	74
Figure 24 Sketch Engine	75
Figure 25 A word sketch of the word love.....	76
Figure 26 Visualisation of a word sketch	76
Figure 27 Trends in Sketch Engine	77
Figure 28 Using text types in Sketch Engine	78
Figure 29 The Google Books Ngram Viewer interface.....	80
Figure 30 The HathiTrust+Bookworm interface	81
Figure 31 Perseus Digital Library home page.....	82
Figure 32 Different tools in Voyant Tools.....	83
Figure 33 The TXM interface.....	84
Figure 34 Interface of BNCweb (CQP-edition).....	84
Figure 35 Interface of corpus.byu.edu	85
Figure 36 Searching on Google Books by subject.....	87
Figure 37 Subject metadata for the first search result.....	87
Figure 38 Searching on Google Books through keywords	88
Figure 39 A list of items where "polka" appears in the HathiTrust+Bookworm	90
Figure 40 The item "Old heads..." in the HathiTrust Digital Library	91
Figure 41 The catalogue record for "Old heads..."	91
Figure 42 Dropdown in HathiTrust+Bookworm	92
Figure 43 TEI encoding of Against Cataline by Cicero in the Perseus Project	93
Figure 44 Encoding of a foreign language	94
Figure 45 Linking to the meaning of a word.....	95
Figure 46 The suggested meaning of a word in the Perseus Project	96
Figure 47 Information about the encoding in the GRAAL corpus	98
Figure 48 Information about the encoding in the VOEUX corpus.....	99
Figure 49 Example of encoding in the BNC (Burnard, 2007)	100
Figure 50 Header information for a text in the BNC	101
Figure 51 The use of wildcards in Google Books Ngram Viewer	103
Figure 52 Search options in Perseus Project.....	104
Figure 53 A search field in Voyant Tools	105
Figure 54 Searching using truncation in Voyant Tools.....	106
Figure 55 Romeo and Juliet encoded to show speakers	106
Figure 56 Selecting sections of text in Voyant Tools.....	107
Figure 57 Searching for a single word in TXM.....	108
Figure 58 TXM query assistant.....	108
Figure 59 Using the query assistant in TXM to create a complex query.....	109
Figure 60 Query executed in TXM.....	110
Figure 61 All direct speech is selected in TXM	110
Figure 62 All texts by a certain author is selected in TXM.....	111
Figure 63 Standard query in BNCweb	112
Figure 64 List of part-of-speech tags in corpus.byu.edu	113
Figure 65 Filtering according to language in Google Books Ngram Viewer	114
Figure 66 Filtering in HathiTrust+Bookworm.....	116
Figure 67 Filtering according to written or spoken texts in the BNCweb	117
Figure 68 Filtering options on the BNCweb (CQP-edition).....	118
Figure 69 Genres of the BNC (written section)	118
Figure 70 Searching in corpus.byu.edu according to sections	119
Figure 71 Viewing results according to section.....	119

Figure 72 Searching for Cupid and Psyche in Google Books Ngram Viewer	120
Figure 73 A search done in Google Books	121
Figure 74 The top search results for a term in a specific year.....	122
Figure 75 Enlargement of Figure 74	123
Figure 76 An item from the search results in the digital library.....	123
Figure 77 An item in the HathiTrust Digital Library is not available due to copyright restrictions.....	124
Figure 78 Results of a search in Perseus Project	124
Figure 79 More information about a specific word in the Perseus Project.....	125
Figure 80 Vocabulary tool in Perseus Project	125
Figure 81 Search results in Voyant Tools	126
Figure 82 Bubblelines in Voyant Tools	127
Figure 83 Concordance in TXM.....	127
Figure 84 Pages in the GRAAL corpus.....	128
Figure 85 Results for a query in BNCweb.....	129
Figure 86 A search result shown in more context in the BNCweb.....	129
Figure 87 Details for a file in the BNC	129
Figure 88 Distribution of results in the BNCweb	130
Figure 89 Results listed in corpus.byu.edu	131
Figure 90 Items in more detail in the corpus.byu.edu	131
Figure 91 Expanded context in corpus.byu.edu.....	131
Figure 92 Page for a word in corpus.byu.edu	132
Figure 93 Help documentation in corpus.byu.edu.....	137
Figure 94 Two sentences encoded on a morphological level.....	162
Figure 95 Graphical representation of dependencies using the Stanford CoreNLP API	164
Figure 96 Encoding of the dependency grammar of a sentence.....	165
Figure 97 Encoding of the semantic information of a sentence.....	166
Figure 98 Example of functional encoding (Clouds of witness).....	172
Figure 99 Example of functional encoding (The life of St. Teresa).....	173
Figure 100 Defining attributes in a schema	174
Figure 101 Adding attributes to an element	174
Figure 102 An example of a TEI header	178
Figure 103 Example of a MODS record created as demonstration	179
Figure 104 First layer of encoding	182
Figure 105 Second layer of encoding	183
Figure 106 Linking of tokens, sentences and documents	199
Figure 107 Sample of encoding to illustrate parsing	200
Figure 108 Comprehensive example of encoding to illustrate parsing	202
Figure 109 Hierarchical structure.....	202
Figure 110 Core classes	204
Figure 111 Home page.....	206
Figure 112 Menu to add items to search	207
Figure 113 Simple search in inkling.....	207
Figure 114 Context for an item	208
Figure 115 Source of text	209
Figure 116 Biography of author	209
Figure 117 Truncation	210
Figure 118 Option to search for inflected forms appears in the menu.....	211
Figure 119 Inflections retrieved	211
Figure 120 Select part-of-speech from menu.....	212

Figure 121 Select noun (singular or mass) from list.....	213
Figure 122 Only nouns are returned.....	213
Figure 123 Adding a section.....	214
Figure 124 Adding a word to search for.....	214
Figure 125 Searching for words near other words.....	215
Figure 126 Searching across sentences.....	215
Figure 127 Searching in the opposite direction.....	215
Figure 128 Selecting to search for dependencies.....	216
Figure 129 Searching for dependencies.....	216
Figure 130 Example where very modifies an adjective.....	217
Figure 131 Example where his modifies a noun.....	217
Figure 132 Searching for instances where reader is the direct object.....	218
Figure 133 Searching according to semantic sense.....	219
Figure 134 A sense has been selected.....	219
Figure 135 Search for a sense and match the value.....	219
Figure 136 Functional properties in the menu.....	220
Figure 137 Searching for the value *ought on inkling.....	221
Figure 138 Selecting to search only in headings.....	221
Figure 139 Filtered according to heading.....	221
Figure 140 Filtering according to paragraph.....	222
Figure 141 Searching for the value her on inkling.....	222
Figure 142 Filtered according to direct speech.....	222
Figure 143 Searching for the value not on inkling.....	223
Figure 144 Filtered according to quote.....	223
Figure 145 Searching for instances ending in -er on inkling.....	224
Figure 146 Filtered according to name.....	224
Figure 147 Searching for instances with -18- on inkling.....	225
Figure 148 Filtered according to date.....	225
Figure 149 Searching for the value part in inkling.....	226
Figure 150 Filtered according to notes.....	226
Figure 151 Searching for the value saint on inkling.....	227
Figure 152 Filtered according to front matter.....	227
Figure 153 Filtered according to back matter.....	227
Figure 154 Filtered according to body.....	228
Figure 155 Searching for the value 'em on inkling.....	228
Figure 156 Searching values as non-regularised.....	228
Figure 157 Combining functional properties on inkling.....	229
Figure 158 Searching for all instances that end with *men.....	229
Figure 159 Filtering according to English as language on a document level.....	230
Figure 160 Filtering according to English on a text-level.....	230
Figure 161 Filtering according to Latin on a text-level.....	230
Figure 162 Searching for the value just on inkling.....	231
Figure 163 Filtering according to date on the document-level.....	231
Figure 164 Filtering according to date on text-level.....	232
Figure 165 Filtering according to date on text-level (2).....	232
Figure 166 Searching for all instances that begin with come.....	233
Figure 167 Filtered according to genre on text-level.....	233
Figure 168 Filtered according to genre on text-level (2).....	233
Figure 169 Searching for instances that start with see written by Wordsworth.....	234
Figure 170 Searching for instances that start with de written by Shakespeare.....	234
Figure 171 Filtered according to publisher, publication place, subject and title.....	235

Figure 172 Combining search options, example 1	235
Figure 173 Combining search options, example 2	236
Figure 174 Combining search options, example 3	236
Figure 175 Searching by using a query language.....	237
Figure 176 Extra features in inkling	239
Figure 177 The Collapse button in inkling.....	239
Figure 178 Examples in inkling.....	241
Figure 179 A prepopulated query	241
Figure 180 A decision support system to provide guidance when encoding texts with detailed metadata.....	285
Figure 181 Enlargement: First part of the decision support system	286
Figure 182 Enlargement: Second part of decision support system	288
Figure 183 Enlargement: Third part of decision support system	290

List of tables

Table 1 Common word classes	28
Table 2 Penn Treebank tagset	29
Table 3 Universal Dependency Relations.....	38
Table 4 Common TEI elements.....	50
Table 5 Comparison of tools.....	140
Table 6 Dependency relationships in table format	163
Table 7 WordNet labels and definitions for a sentence.....	166
Table 8 Functional encoding	169
Table 9 Bibliographic data in TEI.....	180
Table 10 Data stored for each word	186
Table 11 Texts selected for encoding.....	190
Table 12 Possible encodings of each text	192
Table 13 Examples of encoding in texts	195
Table 14 Morphological annotation 1 – Stanford	246
Table 15 Morphological annotation 1 – spaCy.....	246
Table 16 Morphological annotation 2 – Stanford	247
Table 17 Morphological annotation 2 – spaCy.....	247
Table 18 Morphological annotation 3 – Stanford	248
Table 19 Morphological annotation 3 – spaCy.....	248
Table 20 Morphological annotation 4 – Stanford	249
Table 21 Morphological annotation 4 – spaCy.....	250
Table 22 Morphological annotation 5 – Stanford	250
Table 23 Morphological annotation 5 – spaCy.....	251
Table 24 Morphological annotation 6 – Stanford	252
Table 25 Morphological annotation 6 – spaCy.....	253
Table 26 Morphological annotation 7 – Stanford	255
Table 27 Morphological annotation 7 – spaCy.....	256
Table 28 Syntactic encoding 1	259
Table 29 Syntactic encoding 2	259
Table 30 Syntactic encoding 3	260
Table 31 Syntactic encoding 4	261
Table 32 Syntactic encoding 5	262
Table 33 Syntactic encoding 6	265
Table 34 Semantic encoding 1	273
Table 35 Semantic encoding 2.....	274
Table 36 Semantic encoding 3.....	275

List of abbreviations

API – Application Programming Interface
BNC – British National Corpus
CQL – Corpus Query Language
CQP – Corpus Query Processor
EEBO – Early English Books Online
HTML – Hypertext Markup Language
HTRC – HathiTrust Research Centre
KWIC – Key Word In Context
LAS – Labelled Attachment Score
LA OR LS – Labelled Accuracy Score
MARC – MACHine Readable Cataloguing
MODS – Metadata Object Description Schema
NER – Named Entity Recognition
NLP – Natural Language Processing
OCR – Optical Character Recognition
POS – Part-of-speech
TEI – Text Encoding Initiative
UAS – Unlabelled Attachment Score
UD – Universal Dependency
WSD – Word Sense Disambiguation
XML – eXtensible Markup Language

1. Introduction

Beautiful words fill my mind, as I compose this song for the king.

Psalm 45:1 (Good News Translation)

1.1. Background

Data are being produced and consumed at an alarming rate. According to data from Internet Live Stats (<https://www.internetlivestats.com/>), at the time of writing in October 2020 there were more than 4 billion Internet users in the world and more than 1 800 000 000 websites. In 2018, it was estimated that data were being created at 2.5 quintillion bytes of data each day, and that the pace is only accelerating (Marr, 2018).

Not only are data being created digitally, non-digital material is being converted to digital formats. Digitisation efforts across the globe have resulted in large collections of data. Though many different objects are being digitised, such as books, newspapers, paintings or music, this research will focus on collections that contain textual data. Digital collections are often stored and made available in repositories. Some of the most well-known large-scale repositories with large digital collections include Google Books, Internet Archive and HathiTrust. There are many other smaller repositories and collections. The availability of large digital collections is opening up new possibilities. Firstly, it is providing access to primary sources, both digitised and born-digital material. This supports in-depth qualitative studies. Furthermore, it is also enabling researchers to use computer programs to manipulate, process and analyse texts and do quantitative research that was previously not possible due to the lack of available data.

Various tools have been developed to assist researchers and scholars to retrieve and explore the data in large digital collections. For example, the Google Books Ngram Viewer is a tool that enables a user to visualise the relative frequency of words or phrases over a certain period of time. Figure 1 is taken from the Google Books Ngram Viewer documentation and compares the usage of the words “child care”, “kindergarten” and “nursery school” from 1950 to 2000. The corpus selected in this example is the American English corpus.

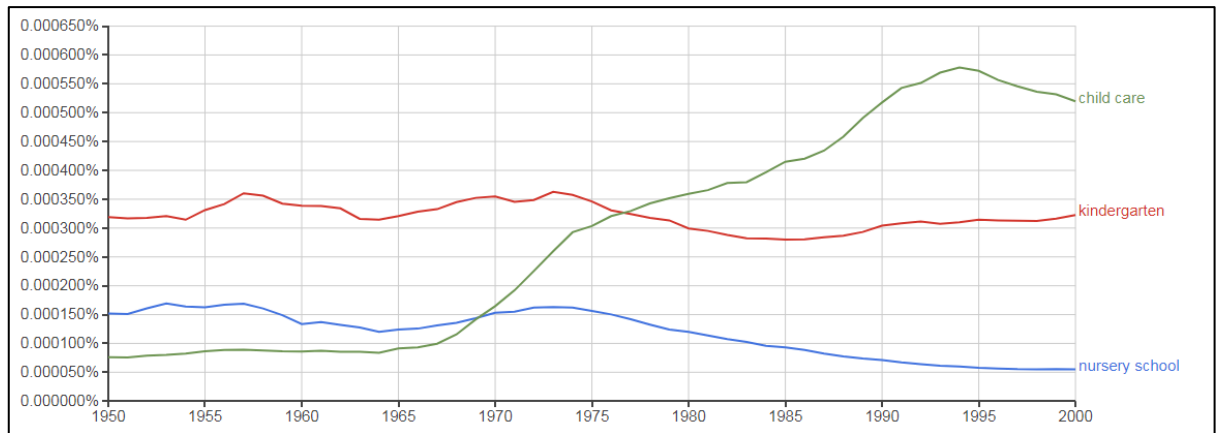


Figure 1 An example of the Google Books Ngram Viewer (Google Books Ngram Viewer Info, 2020)

This computational tool has been met with much enthusiasm and has been used in various studies. For example, Keuleers et al. (2011) discuss how the Google Books Ngram Viewer can be used in psycholinguistic research. Another application is in the field of lexicography, where analysis can help to determine low-frequency words that are not listed in dictionaries and to provide more accurate estimates of current frequency words (Michel et al., 2011: 4). These methods have also been used to study culture. Computational analysis of data on a large scale has been used to observe changes in the frequency of word usage to detect and quantify cultural trends and evolution (Michel et al., 2011: 2). For example, Michel et al. (2011) used the Ngram Viewer to investigate how long fame lasts, and discovered (amongst other things) that fame is increasingly short-lived.

Unfortunately, the Google Books Ngram Viewer has also received much criticism. The main concern seems to be the lack of available metadata (e.g. Jockers, 2010; Koplenig, 2017). Due to copyright restrictions, Google Books cannot release the full-text that is used in the Google Books Ngram Viewer nor can they provide a bibliography of the items in the corpus (Culturomics, 2017). As such, there is no way to know the composition of the corpus and how it changes over time. Pechenick et al. (2015: 12) remark that “the Google Books corpus’s beguiling power to immediately quantify a vast range of linguistic trends warrants a very cautious approach to any effort to extract scientifically meaningful results ... Google Books is at best a limited proxy for social information after the fact”.

The HathiTrust+Bookworm project is another experimental tool that enables researchers to observe trends in large text collections. The HathiTrust+Bookworm is a visualisation tool that works with data from the HathiTrust Digital Library. Though many of the texts in this repository are subject to copyright, there are extensive metadata and

users can extract specific data for analysis by using the bibliographic metadata available. Figure 2 shows that the data can be filtered according to a set of metadata.

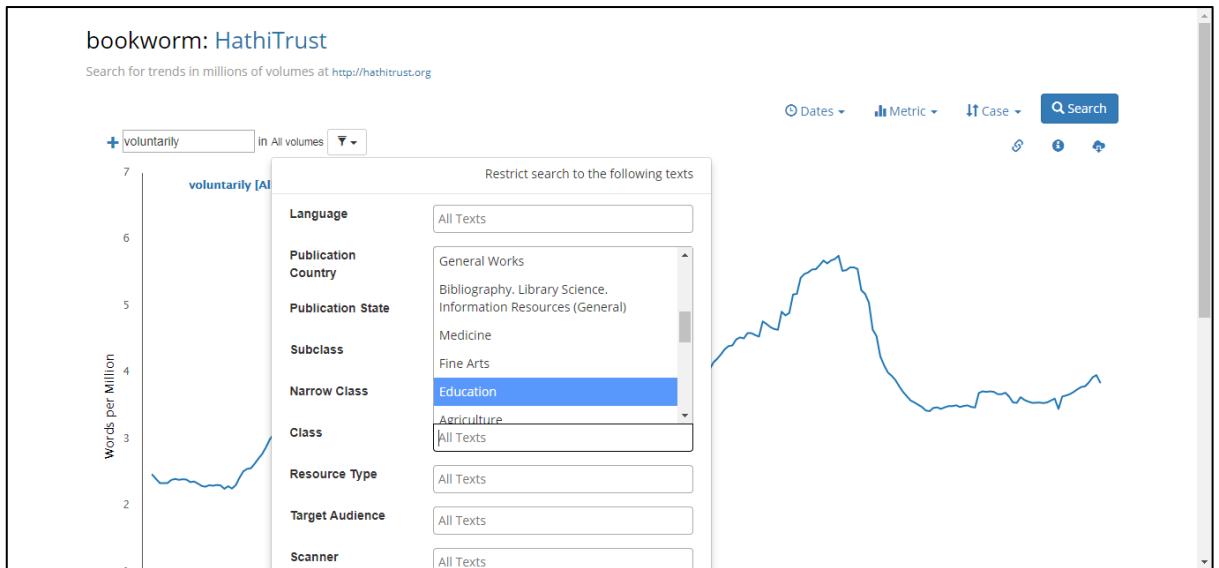


Figure 2 Filtering in HathiTrust+Bookworm

Both the Google Books Ngram Viewer and the HathiTrust+Bookworm project will be discussed in depth in chapter 2.

Selecting texts by using coarse bibliographic metadata does not seem to be sufficient for scholars working with and analysing texts. A study by Fenlon et al. (2014) revealed that scholars want to be able to extract texts on a fine-grained level. In other words, more than just on a high bibliographic level, some users would like to be able to search for words or phrases with certain criteria.

1.2. Research problem

Due to various digitisation efforts, there is a staggering number of digital objects available, and this number is constantly increasing. The idea to exploit these large digital collections for analysis has become increasingly popular. Tools, such as the Google Books Ngram Viewer, have been developed to allow for the analysis of such corpora.

However, as has been argued in the previous section, there are many concerns about the current analysis of large collections of texts, particularly due to the lack of available metadata. In addition to proper (and available) bibliographic metadata, if texts can be encoded on a fine-grained level, improved retrieval and analysis can be done. This refers to the fact that if information about properties of the text, such as the structure of the text, characteristics of words or grammar, can be made explicit it can be understood and used by a machine. Xiao (2010: 158) mentions many types of

information on different levels, for example, at the phonological level, syllable boundaries can be indicated; at the morphological level, the corpora can be annotated in terms of prefixes, suffixes and stems; at the lexical level, part-of-speech, lemmas and semantic fields can be indicated; at the syntactic level, information from syntactic analysis can be indicated; at the discourse level, the anaphoric relations, pragmatic information or stylistic features can be indicated.

A couple of specific examples to illustrate the usefulness of different levels of information that are available about text are mentioned here. One could compare the usage of a word over time in direct speech or in indirect speech (for example *wanna* used in direct versus indirect speech). Texts could consist of sections from different time periods. For example, a published volume could have an introductory section discussing a drama, followed by the drama itself. These two sections could be from very different time periods. The notes about the drama could be from the 1900s and the drama could be a Shakespearean drama from the 1600s. Sometimes text contains words or phrases from another language, such as Latin or French phrases in an English text. A scholar might want to exclude all foreign languages from the corpus that (s)he is working with, or search for only words or phrases that are not in the primary language of the text. A person might want to retrieve words by an author (e.g. Shakespeare) that are quoted in other documents.

In order for such detailed information to be used by a machine, this information will have to be made explicit through encoding the text. The idea that there is a vast amount of information about texts and words in texts that could be useful for retrieving very specific instances from a large collection has led to the following research question:

How can texts be encoded with detailed metadata to improve retrieval of words or phrases with specific properties from digital text collections?

To answer this question, the following sub-questions need to be answered:

- 1) What metadata elements are available with which to describe and encode texts?
- 2) What are the characteristics of some of the tools that are currently used for the retrieval of words from digital text collections?
- 3) To what extent do current tools allow for the retrieval of words or phrases with specific properties from text collections?

- 4) What encoding can be suggested to allow for retrieval on a greater level of granularity and specificity?
- 5) To what extent can a tool (or prototype) be used to retrieve words or phrases with specific properties from texts encoded with detailed metadata?
- 6) What recommendations can be made to automate the encoding of texts?
- 7) How can the process followed in this study be formalised to make a contribution to theory development?
- 8) What recommendations can be made based on the results of this study?

Much of the work in digital humanities is in manipulating, processing and analysing text (Terras, 2016: 1638). This research falls in the field of digital humanities, as it aims to see to what extent text can be encoded (marked up) to improve retrieval.

At this point it is useful to consider different concepts dealing with the idea of describing an item.

1.3. Clarification of concepts

In this section some concepts that are used in different disciplines will be considered: metadata, markup, annotation and encoding.

1.3.1. Metadata

The first concept of interest to this study is metadata. The term metadata has wide use and could have different meanings in different contexts. Zeng and Qin (2016: 18) explain that “the last two decades of metadata development have witnessed a continual expansion and evolution of metadata research and practices at almost all levels and in almost all disciplines”.

It is sometimes used to mean any machine understandable information, but in the library community it is typically used to describe resources (Hodge, 2001). At a very basic level it is defined as “data about data” (Riley, 2017: 1), but much more can be said about this concept. Metadata has been explained as “what one can say at a given moment about any information object at any level of aggregation” (Gilliland, 2016). Pomerantz (2015: 26) argues that data can be described as potentially relevant information and therefore describes metadata as “a statement about a potentially informative object”. Zeng and Qin (2016: 11) explain that, generally speaking, “metadata encapsulate the information that describes any information-bearing entity”, but that metadata can be seen as a broader concept and that “metadata exist not only

in the traditional bibliographic data universe” (Zeng & Qin, 2016: 12). Sicilia (2014: 4) explains that it is important to consider that metadata are structured information, meaning that the metadata elements are organised systematically, and that metadata have specific functions, most often to facilitate search and discover.

Although metadata are not the object that they describe, it has been argued that when enough metadata about an object are combined, a large amount of information about that object is available (Pomerantz, 2015) to the point that the object can be reconstructed (Haynes, 2018).

There are different categories of metadata, including descriptive, administrative, structural, preservation and use metadata (Pomerantz, 2015). Descriptive metadata are used to describe an item and are typically useful for discovery purposes; the information gleaned from administrative metadata will be helpful in managing the resources; structural metadata are used to indicate the relationship between different components of a complex object; preservation metadata provide information about how to preserve an object; and use metadata are used to capture information about how an item has been used.

Metadata can be embedded in an object or be separate from an item and be linked to the item that they describe (Gilliland, 2016; Hodge, 2001; Pomerantz, 2015; Zeng & Qin, 2016). There are advantages and disadvantages to both options. Storing the metadata and the item together reduces the risk of loss but storing them separately could make it easier to manage the metadata.

There are several reasons for describing items with metadata. One of the most important reasons for this study is that it facilitates the discovery of resources (Hodge, 2001; Gilliland, 2016; Haynes, 2018). Other reasons include that it can help to describe items, organise items, facilitate interoperability and integration, ensure authenticity of items (validation), help with management of items and support preservation (e.g. Gilliland, 2016; Haynes, 2018; Hodge, 2001; Zhang & Gourley, 2008). Jett et al. (2016b) emphasise the importance of metadata when working with digital collections, because the way in which scholars select objects or items of interest to work with in a digital library is through the metadata.

According to Gilliland (2016), information objects have three features that are described by metadata, namely, content, context and structure. Content refers to what the object is about, context refers to information about the object’s creation and life, and structure refers to relationships between the object and other objects.

Gilliland (2016) also argues that there has been less of an emphasis on the “structure of information objects in metadata development by the library, archives, and museum communities” and that there are different levels of metadata, namely item-level or within-item-level metadata.

Metadata schemes are “sets of metadata elements designed for a particular purpose, for example, to describe a particular type of information resource” (Hodge, 2001) or as Pomerantz (2015) explains, they are the rules that govern what sort of subject-predicate-object statements are allowed, where the subject is the entity to be described, the object is the value that describes the entity and the predicate is the relationship between the two. There are many metadata schemes, such as Dublin Core, MODS, VRA Core, ONIX international.

Despite the various schemes that have been developed there are often additional requirements in different contexts. This could be addressed by extending the scheme, for example Dublin Core can be extended by the addition of more elements or by adding qualifiers (as defined by the scheme) (Pomerantz, 2015).

1.3.2. Markup

This brings us to the next term that is used in the process of describing texts, namely markup. According to Xiao (2010: 155), corpus markup “is a system of standard codes inserted into a document stored in electronic form”, where these codes can give information about the text itself, as well as structural information of the text. Haynes (2018: 19) explains that it is the “way in which metadata can be applied to and expressed in documents”. Burnard (1995) explains that the term originally meant that marks (annotations) were added to a document to indicate to a typist or compositor how the text should be typed or laid out. As the production of texts started to rely more on machines, the marks were being used in electronic texts to indicate how the text should be processed. Markup is distinct from the actual text and identifies features of the text that a machine can control or process (Burnard, 1995; Renear, 2004).

Marking up a text can be very valuable. It is a necessary step in order to test certain language theories and it is a way of marking features that are not apparent in the text (Anthony, 2013: 148). Furthermore, markup of a text gives context and allows a broader range of interesting research questions to be addressed (Xiao, 2010: 155). Markup that stores information about a text is also critical for information retrieval (Renear, 2004). Different types of information of a text that are made explicit through markup makes it possible to retrieve very specific kinds of data. Gregory et al. (2016: 994) explain that digitised sources should ultimately be used in research to generate

new knowledge, however, emphasising that it is not a simple process. They explain that intermediate work, such as preparing texts for analysis, might help researchers to contribute new knowledge. In other words, marking up a text could help with research.

1.3.3. Annotations

Another concept that is used is annotations. According to Pustejovsky and Stubbs (2012), annotations are “metadata that provides additional information about the text” and any “metadata tag used to mark up elements of the dataset is called an annotation over the input”. Pustejovsky and Stubbs (2012) explain that annotations are valuable as they allow computers to find patterns in texts and make inferences. Different aspects of language can be annotated, including part-of-speech (POS), phrase structure, and dependency structure (Pustejovsky & Stubbs, 2012). Xiao (2010: 158) differentiates between markup and annotation, suggesting that markup refers to “objectively verifiable information regarding the components of a corpus and the textual structure of each text” and that annotation is concerned with “interpretative linguistic information”. However, he accepts that the distinction is not made by all and that the terms are used interchangeably in literature.

1.3.4. Encoding

The next important term is encoding. As has been explained, there is additional information about a text that is implicit (can be inferred) but needs to be made explicit for a machine (Van den Branden et al., 2017). Van den Branden et al. (2017) call this additional information meta-information and the process by which such meta-information is added to a text is called encoding. Renear (2004) described encoding as the “practice of creating machine-readable texts to support humanities research”. This is often done through using tags (Van den Branden et al., 2017). Pioneering work in literary text encoding was done by Father Roberto Busa who created the *Index Thomisticus* and used IBM punched-card equipment for his project.

The advantages of a standard approach to text encoding was soon realised and a project to develop guidelines was established (Renear, 2004). The resulting guidelines, namely the TEI (Text Encoding Initiative) Guidelines, have been very successful (Renear, 2004) and speaking about TEI, Schmidt (2012: 128) says “it is obvious that this technology: embedded XML markup, is the norm for encoding texts in the digital humanities”.

1.3.5. Summary

It is clear that the terms overlap and have very similar meanings, though with distinct emphases in different disciplines. In this thesis the researcher will use the term metadata as a broad, overarching term, based on explanations such as “data about data” (Riley, 2017: 1) or “provide additional information about a text” and thereby meaning that metadata are any additional (meta) items of information about an entity (be it a word or a book) that are made explicit so that they are interpretable and usable by a machine. This means that, in this thesis, metadata will include all types of data about a text, from morphological, syntactic, semantic, functional to bibliographic data. However, in some cases sensitivity to a particular discipline will be maintained, if thought that it will make the discussion clearer. For example, in the field of corpus linguistics, the term *annotations* is often used when referring to morphological or syntactic data added to a corpus. The researcher will use annotations when referring specifically to linguistic data. The terms *encoding* or *annotating* will be used to describe the act of applying metadata or annotation to a text, in other words marking up a text. An encoded text or annotated corpus will therefore be a text where metadata or annotations have been added.

There is much debate about whether corpora should be annotated (Anthony, 2013: 147-148). Those against annotating corpora argue that annotations contaminate the original data. This is particularly problematic for researchers who see the corpus itself as the starting point for analysis (Anthony, 2013: 147).

This study argues that the advantages that metadata added to a text bring, outweigh the disadvantages. Furthermore, Anthony (2013: 148) argues that with modern tools, it should be possible to ignore or exclude certain layers of metadata, and a user should be able to work with the type of corpus that is best suited to his/her needs.

1.4. Research methodology

In order to answer the main research question and sub-questions, the researcher firstly had to determine what metadata could be useful to improve retrieval in digital text collections. The researcher then had to investigate the extent to which current tools are able to allow users to retrieve words or phrases with specific properties. After evaluating the current landscape, the researcher had to suggest a set of metadata with which to encode texts to enhance retrieval. These suggested metadata were applied to a selection of texts and tested to determine if retrieval is in fact improved. In order to test retrieval in these encoded texts, a prototype needed to be developed. This prototype receives encoded texts, parses the texts and stores the data in a database.

An interface was developed to allow for the retrieval of words or phrases based on certain properties.

This research study did not follow a single standard method, but a combination of research methods. Though not a conventional grounded theory research study, the researcher relied on the principles of this approach in the study. In grounded theory research the researcher does not start with a theoretical framework, but works from the data in order to determine categories and theories (e.g. Leedy & Ormrod, 2020 but expanded on in chapter 3). In order to determine what metadata could be useful for retrieval and what current tools are available for working with large digital text collections, a literature review was conducted. In the literature review the focus was on tools that are commonly used to search in text collections, elements that can be encoded to enable retrieval and standards that can be used for the encoding of the metadata.

After identifying current tools used to search in text collections, these tools were evaluated through heuristic evaluation. Heuristic evaluation involves the use of criteria, by an expert, to evaluate a product or system. A set of criteria was identified, and each tool was reviewed according to these criteria.

After identifying detailed metadata that could be applied to texts, a metadata framework was suggested that could be used to improve retrieval. The suggested framework includes five categories of metadata. The way in which these metadata can be encoded was given.

This suggested metadata framework was applied to a selection of texts. By selecting only some texts to encode, the researcher relied on principles from case study research. Instead of building a full-scale system, a prototype was developed. Prototyping is used to prove a concept. By practically testing the retrieval of the encoded texts in a system, the researcher relied on principles of action research.

These methods will be discussed in chapter 3.

1.5. Limitations to the study

The following limitations in this study have been identified.

Firstly, though the ideal is that the findings can be extended to other cases, it might be difficult to generalise the results of this study as only a selection of texts will be used in this study.

Secondly, the researcher might only be able to encode the text to a certain level of granularity and not address all needs. There could be many more items in a text that can be encoded. However, this study will show that though only a selection of items was encoded, the standard used for the encoding is extensible and more items could be added.

Thirdly, though it is important to equip researchers from various fields with programming skills, it will not be the focus of this research. The focus of this research is on how users can analyse texts through searching and retrieving words or phrases with specific properties with the use of a tool. An important aspect of this study is to consider to what extent researchers or laypersons with limited technical skills, as well as advanced users who wish to search with queries that are more complex, can be accommodated with the same tool.

Fourthly, the purpose of the study is to prove the concept of improved retrieval with detailed metadata and not to build a commercially viable product. Therefore, a selection of texts will be encoded and a prototype will be developed to test retrieval.

1.6. Outline of chapters

Chapter 1 introduces the research topic and the motivation for the study.

Chapter 2 will provide an overview of relevant literature, including a discussion of metadata that can be used to enhance retrieval, of tools used to search in and explore text collections, and an evaluation of current tools.

Chapter 3 will discuss the research methodology of this study.

Chapter 4 will present the suggested metadata for text encoding to improve retrieval.

Chapter 5 will include the sample of texts that have been encoded.

Chapter 6 will discuss the prototype that was developed to search and retrieve words or phrases with specific properties from the encoded texts.

Chapter 7 will consider to what extent the encoding of the texts can be automated.

Chapter 8 will conclude the study by answering the research questions and making various recommendations and suggestions for future research.

2. Review of relevant literature and software

Only words, words; to be led out to battle against other words.

Till We Have Faces by C.S. Lewis

2.1. Introduction

The amount of digital material in the world is staggering, and it is continually increasing. Not only are vast amounts of data created digitally or generated by programs, a large amount of non-digital material has been digitised through various digitisation initiatives across the world. Hitchcock (2013: 9) states that “we are witnessing the creation of the Western print archive, second edition”.

Apart from a large amount of information being available and accessible to read, the digital medium has opened ways of working with information that were not possible before (e.g. Gregory, 2014; Nicholson, 2013). New methods or techniques, such as data mining, text mining, digital mapping, visualisation and text analysis, are being used to do work or research previously not possible.

One example will be given here to illustrate how researchers make use of digital collections in innovative ways to support research. One of the projects done by the Library of Congress was to digitise and transcribe the papers of the baseball icon Branch Rickey (<https://www.loc.gov/collections/branch-rickey-papers/>). Volunteers transcribed the 1,926 pages of Rickey’s scouting reports so that they could be used for research. Tucker (2019) explains that this digital collection opens up new avenues for researchers; the collection can be searched for specific keywords, text analysis of the diction can be done, researchers can look for patterns in the text (e.g. names, dates or places) and other data questions can be asked.

Though this can be considered a relatively small collection, collections are growing and there are some very large digital collections that are bringing researchers working with such collections into the field of big data.

It follows that researchers working with large text collections would need skills or tools to enable them to do research with big datasets. This research will focus on the tools that are available to enable effective retrieval.

In order to determine to what extent different types of users can currently retrieve words or phrases from text collections and observe trends, this chapter will investigate some tools used to explore text collections. This means that this chapter is not a

traditional literature review per se, but will mostly consist of the reviewing of software tools. In this evaluation of tools, the background and purpose of the different tools will be discussed and then the tools will be compared.

After evaluating various tools, the researcher will discuss the current situation, as well as make recommendations for future developments in this area.

In order to give some context, this chapter will start by discussing the great progress in digitisation efforts across the globe and some of the implications of large digital text collections for research. This chapter will briefly consider the concept of digital humanities, as the application of technology to humanities scholarship falls in the field of digital humanities. The chapter will also consider what metadata would be useful to use to enhance retrieval. Next, the chapter will cover some of the tools that are used to search for words or phrases in digital textual collections, the extent to which words from specific sections can be retrieved, and how trends over a period of time can be observed. This chapter will also include criticism against current tools.

2.2. Digitisation

Libraries have been involved in digitisation projects since the 1990s (Lopatin, 2006: 273). Digitisation is done primarily to provide access and to preserve material (Lopatin, 2006). Typically, in a digitisation project, books are scanned, optical character recognition (OCR) technology is applied to make the full-text searchable and, depending on copyright, the text or a portion of the text is made available online.

Digitisation projects are complex and involve the managing of budgets, staffing, workflow, technical issues, and metadata (Lopatin, 2006: 274). The digitisation of large amounts of material could be an expensive exercise (Gregory et al., 2016: 994). According to calculations by Brewster Kahle from the Internet Archive, it costs ten US cents to digitise a page (Kahle & Vadillo, 2015: 5). Although a seemingly small amount, this can add up to a large sum for a large collection of books. A digitisation project requires appropriate hardware, software and staff trained to do scanning, quality control and metadata creation (Lopatin, 2006: 275).

Apart from institutions conducting their own digitising projects, there are also collaborative projects, and some large-scale book-digitisation projects. Four well-known large-scale digitisation projects, amongst others, are Google Books, Internet Archive, HathiTrust and Project Gutenberg.

Google Books

Google Books is a massive digitisation effort by Google with the purpose to make information discoverable globally (Bergquist, 2006). Another description of Google Books is that it is an online equivalent of a card catalogue (Baksik, 2006: 403). Books provided by libraries and publishers were scanned and the text made searchable by using optical character recognition (OCR) (Michel et al., 2011: 2). By 2015 more than 25 million volumes had been scanned, including texts from over 100 countries and in 400 different languages (Heyman, 2015). Metadata provided by libraries and bibliographic databases were added to the books (Michel et al., 2011: 2). The fact that the texts of the books are searchable means that a user is not restricted to searching bibliographic data, but can search for words or phrases within the text of a book. A user can use Google's standard search page or the Google Books home page to search for books.

Internet Archive

The Internet Archive is "a non-profit library of millions of free books, movies, software, music, websites, and more" (Internet Archive, n.d.) with the mission to provide "Universal Access to All Knowledge" (Kahle & Vadillo, 2015: 2). The organisation started by archiving web pages and later started to include digital versions of other published works (Kahle & Vadillo, 2015: 3). At the time of writing the Internet Archive contains 330 billion web pages, 20 million books and texts, 4.5 million audio recordings (including 180 000 live concerts), 4 million videos (including 1.6 million Television News programs), 3 million images and 200 000 software programs (Internet Archive, n.d.). Apart from its own digitisation, the Internet Archive assists over 500 libraries to digitise books, preserve them and provide access to them (Kahle, 2018). In addition, individuals with an Internet Archive account can upload items to the archive (Internet Archive, n.d.). In this way, many books that have been digitised by Google were downloaded and uploaded to the Internet Archive by users. The Internet Archive has a particularly large collection of 19th century literature and it has become an important tool for researchers interested in this era (Kahle & Vadillo, 2015: 2).

HathiTrust Digital Library

The HathiTrust Digital Library is a large digital library developed by the HathiTrust consortium. HathiTrust was established in 2008 as a collaboration between the universities of the Committee on Institutional Cooperation and the University of California system (HathiTrust Digital Library – Our Partnership, n.d.). This partnership

between various institutions was formed in order to address the challenges and costs of large-scale digitisation efforts, the increase in digital content, along with the preservation of digital items (Christenson, 2011). HathiTrust soon expanded and included other partners. As a result of this collaboration, it was possible to create a repository of digital resources hosted by libraries at a scale not seen before (Christenson, 2011). The core of HathiTrust is a shared secure digital repository, HathiTrust Digital Library, owned and managed by the consortium of research institutions (Christenson, 2011). At the time of writing, the HathiTrust Digital Library contained a total volume of over 17 million items (HathiTrust, 2020).

Libraries have a responsibility to preserve information. Commercial companies do not necessarily make long-term commitments to preserve items and mass-digitisation efforts by commercial companies have been called “access digitisation” as opposed to “preservation digitisation” (Leetaru, 2008). HathiTrust addresses this issue in that “the HathiTrust partnership enables an aggregation of digital resources not seen before, hosted by libraries for the long term in a continuation of their traditional role as stewards of the scholarly record and supporters of research and other scholarly pursuits” (Christenson, 2011).

The content in HathiTrust Digital Library is from different sources, firstly from the in-house digitisation efforts of partner institutions, but also items from elsewhere, including Google Books, the Internet Archive and Microsoft (HathiTrust Digital Library – Our Digital Library, n.d.). However, other institutions can also contribute. The library provides ingest guidelines for institutions who wish to contribute content to the library, for example, in terms of metadata, institutions are required to provide accurate and bibliographic descriptions that are as complete as possible in valid MARC21 format (HathiTrust Digital Library – Bibliographic metadata specifications, n.d.; HathiTrust Digital Library – Guidelines for Digital Object Deposit, 2011).

Project Gutenberg

Project Gutenberg is a digital library of free ebooks (Project Gutenberg, n.d.). Michael Hart established the repository in 1971 (Vaknin, 2005). Books are added to Project Gutenberg by volunteers and it is part of Project Gutenberg’s mission to interfere very little with the volunteers (Thomas, 2007). Most books on Project Gutenberg are in the public domain, and where titles are still under copyright, permission has been obtained to make them available (Project Gutenberg, n.d.). At the time of writing Project Gutenberg had over 60 000 free ebooks in its collection (Project Gutenberg, n.d.). Most of the ebooks are in English, but there are ebooks in other languages as well (Thomas,

2007). Project Gutenberg is for human consumption only and the collection may not be analysed by an automated tool (Project Gutenberg, n.d.).

2.3. Digital humanities

Digital collections are becoming increasingly important for researchers. A study of the impact of two specific digital collections, Early English Books Online and House of Commons Parliamentary Papers, confirmed that the usage of these collections has increased over time, and the authors conclude that “digital collections have become fundamental to modern scholarship” (Meyer & Eccles, 2016).

Digital collections are not only important because they give access to texts (or other items), but because new ways to engage with texts and new research methods are being used. Rydberg-Cox et al. (2000) explain that “[the] true benefit of a digital library, however, comes not from the replication and enhancement of traditional library functions, but rather in the ability to make possible tasks that would not be possible outside the electronic environment, such as the hypertextual linking of related texts, full text searching of holdings, and the integration of knowledge management, data visualization, and geographic information tools with the texts in the digital library”.

The availability of a large number of digital texts and the ability to process these texts with the use of technology, has indeed opened up new opportunities for scholarly research. Researchers from humanities are increasingly engaging with big data (Howard, 2017). The question ‘what can we do that we couldn’t do before?’ is being asked (Nicholson, 2013: 63). There is a sense that technology and the information that is available digitally should not only enable researchers to work faster and more efficiently, but indeed enable researchers to produce new kinds of research (Nicholson, 2013: 63). As such, there is an increasing interest in using new methods in the humanities (Nicholson, 2013). Methods from corpus linguistics, distant reading, network analysis and geographic information systems are being used on large volumes of text (Gregory, 2014).

This intersection of content and technologies, where computational methods are applied in arts and humanities, is referred to as digital humanities (Henry & Smith, 2010: 107; Terras, 2016: 1637). “While previous attempts to integrate computer technology into the humanities have failed to take off, the Digital Humanities have established a firm foothold in the academy and promise to play an increasingly prominent role” (Nicholson, 2012: 240). Since 2004 the term *digital humanities* has been widely used, partly due to the 2004 Blackwell publication, the *Companion to Digital Humanities* (Kirschenbaum, 2013: 198; Terras, 2016: 1637; Vanhoutte, 2013:

119-120). The term has been applied broadly to a range of activities (Terras, 2016: 1637) and the concept *digital humanities* remains widely debated (Terras et al., 2013: 6). One of the problems in defining digital humanities has been the broad range of projects and topics that are labelled as digital humanities, for example, projects ranging from “textual analysis of medieval texts and establishment of metadata schemes to the production of alternative computer games and artistic readings of nanotechnology” are all considered part of digital humanities (Svensson, 2013: 161). Terras (2016: 1638) tries to emphasise the broad nature of digital humanities, by saying that in the field of digital humanities, scholars are encouraged to “[think] about computational methods in the arts and humanities, and then into culture and heritage, in as broad a sense as possible”. Similarly, Henry and Smith (2010: 107) explain that digital humanists create digital objects and digital collections. They then use computational methods to analyse these digital objects and answer questions. In these cases, digital humanities are still seen as a field where some computational methods are used. An even broader view is that digital humanities “involves the use of digital tools in research, teaching, scholarship and publication in humanities disciplines” (Viiri, 2014: 5).

It is clear that technology has had an impact on research in the humanities. Firstly, the use of technology in the humanities can be used in traditional ways, but faster and more efficiently (Viiri, 2014: 5). Furthermore, the use of technology enables researchers to do work that was not previously possible, for example using maps and geographic information systems technology (Viiri, 2014: 5).

Some researchers have embraced these new methods with enthusiasm and have been so bold as to suggest that scholars should step back from “scrutinizing individual texts to probe whole systems by counting, mapping, and graphing novels” (Parry, 2010). Others do not completely throw out the careful studying of texts, but acknowledge the benefits of computational analysis. For example, in the words of Nicholson (2012: 245) “it is rarely possible to see these wider patterns through close reading alone... [we need] not be afraid to read the archive from a distance”, but once the patterns have been identified a manageable sample can be selected for closer reading.

Many interesting studies using computational methods on text collections have been done. Some of the work done in this field will be discussed here. Franco Moretti coined the term *distant reading*, where quantitative analysis is used to study literary texts (Moretti, 2003). Specifically, Moretti uses graphs, maps and trees to visualise the literary field. One of the more recent collaborative projects that Moretti was involved with is the mapping of the emotions in London (Heuser et al., 2016). Jockers (2011)

used quantitative methods (for example unsupervised topic modelling) to identify differences between 19th century British and Irish novels. It is by no means only digitised books that are studied. Leetaru (2011) applied text mining techniques (sentiment mining and full-text geocoding) to a 30-year worldwide news archive to track changes in the tone of the news. Gregory et al. (2016) did research using the British Library's 19th Century Newspaper collection. The texts in this collection were annotated by using part-of-speech and semantic taggers, then corpus linguistic techniques were used to answer research questions. For example, the researchers established the frequency counts of how often Russia and France appeared in the collection, as well as how often these names appear in the collection close to the word *war* or words related to war. Lansdall-Welfare et al. (2017) used a corpus of British regional newspapers to investigate continuity and change in history, and showed that "computational approaches can establish a meaningful relationship between a given signal in large-scale textual corpora and verifiable historical moments" (Lansdall-Welfare et al., 2017: 461).

Some studies using the Google Books Ngram Viewer have already been mentioned in chapter 1. Additional examples are given here. Michel et al. (2011) used this corpus to investigate cultural trends quantitatively. In this study they quantified the evolution of grammar, examined the rate at which society forgets the past, the rate at which technology is adopted, and how censorship can be investigated. Acerbi et al. (2013) investigated the use of "mood" words to detect positive and negative historical periods. Ophir (2016) analysed the patterns of the term *truth* during the last five centuries. Juola (2013) used this corpus to measure cultural complexity and showed an increasing complexity due to the cumulative nature of culture.

Data from the HathiTrust Digital Library have also been used in different studies. For example, Underwood (2015b) showed (using texts from HathiTrust Digital Library) how machine learning could be used to apply genre labels, not only at volume level, but even at page level. Underwood et al. (2018) also explored the change in gender in English-Language fiction and discovered that there is a decline in the proportion of fiction written by women. Furthermore, Mimno (2014) used the data in a word similarity tool that tracks word co-occurrence patterns from 1800 to 1923; Forster (2015) studied the gender of authors in the collections; Goodwin (2015) created a topic browser to explore the data.

When computational methods are used on corpora that include works over many years to detect macroscopic and long-term cultural trends it is referred to as *Culturomics* (Michel et al., 2011: 7).

Though there are exciting possibilities, it seems that digital resources and computational methods have not been used to the extent that was anticipated. “[There] remain major barriers to the widespread uptake of these datasets, and related computational approaches, by humanities researchers” (Terras et al., 2017: 457).

The next section will consider some of the criticisms against large digital collections and the use of computational methods in the humanities.

The quality of some large digital collections is a concern. It is time-consuming and expensive to create quality collections, and as such, large-scale collections use scalable methods such as automated scanning and OCR to produce searchable text (Gooding, 2013: 426). Unfortunately, these techniques still produce errors and are creating the perception of a “virtual rubbish dump of our cultural heritage” (Gooding, 2013: 425). Gregory et al. (2016: 994) write that historians have been criticised that they have failed to use digitised resources extensively. As response to this criticism, historians point out the poor quality of much of the digitised sources (Gregory, 2014; Hitchcock, 2013). Problems with optical character recognition (OCR) have been pointed out by various researchers (Gregory, 2014; Henry & Smith, 2010; Hitchcock, 2013). Henry and Smith (2010: 106) report on research that was done to investigate the possibilities of research on text corpora made available through selected large-scale digitisation programs and state that until recently the focus of large-scale digitisation projects has been on copyright, technical issues, and quality (Henry & Smith, 2010: 106).

Apart from the quality of the collections, the quality of metadata, or the lack of metadata, in large digital collections has been discussed repeatedly (e.g. Henry & Smith, 2010; Koplenig, 2017; Nunberg, 2009).

Another concern about digital collections (specifically large collections) is the composition of these collections. Henry and Smith (2010: 110) point out that large-scale digitisation projects focus on quantity and as a result, collections can be uneven. Metadata about the items in the corpus could inform users about the structure of the corpus (Koplenig, 2017). However, as was highlighted in the previous section, in many large digital collections, metadata are missing or incorrect. This limits the extent to which the collection can be used reliably.

Copyright remains a challenge when creating a collection of digitised texts (Lopatin, 2006: 277). Copyright in the US may vary depending on the circumstances of the work, but typically protects publications after 1923 (as explained by Organisciak et al., 2017: 4). In South Africa works are generally protected for the lifetime of the author/creator plus 50 years from the end of the year in which (s)he dies (South Africa, 1978, s25). Both Google Books and the HathiTrust Digital Library restrict access to volumes not in the public domain. As a result, the full-texts used to generate the n-grams in the Google Books Ngram Viewer are also not made publicly available (Culturomics, 2017). The HathiTrust Digital Library tries to offer a solution to the challenge of copyright restrictions, by offering a secure environment, where scholars can do computational analysis on texts without accessing or reading individual items (as described by Bhattacharyya et al., 2015; Murdock et al., 2017).

Apart from the problems regarding the content, the methods used to do research on these collections have been criticised. The use of computational methods (such as distant reading or macroanalysis) in the humanities has been met with severe criticism. Some researchers maintain that literature and art, with all its nuances and complexity, cannot really be studied using computational methods, and that in-depth study by humans is necessary. Jenkins (2013) says that

the value of the arts, the quality of a play or a painting, is not measurable. You could put all sorts of data into a machine: dates, colours, images, box office receipts, and none of it could explain what the artwork is, what it means, and why it is powerful. That requires man, not machine.

Sceptics find that computational methods “take the human out of the humanities” (Parry, 2010) and “that it seems to necessitate an impersonal invisible hand” (Trumpener, 2009: 164). Trumpener (2009: 171) continues by arguing that “it is equally important that most of us forego counting ... We can change our parameters and our questions simply by reading more”.

Furthermore, there is a concern that the lure of technology and large datasets will have a potential negative impact upon qualitative research (Gooding et al., 2012; Nunberg, 2010).

Another point of criticism is that digitisation and computational approaches do not consider the medium, context or history of items. “When a cultural artefact is removed from its original form, there is a danger that it will be stripped of its context, its history, and thus its authenticity” (Gooding et al., 2013: 634). Bode (2017) criticises studies

where the textual scholarship of texts (bibliographic and editorial practices) are ignored. She argues that literary works are not stable and singular entities, but that they have historical context that should be captured in a scholarly edition. She is in the process of creating such a curated dataset, which will include “detailed bibliographic metadata and digitized text for approximately sixteen thousand stories published in nineteenth-century Australian newspapers” (Bode, 2017: 18).

Some researchers accept computational approaches in the humanities, but criticise the type of distant reading where the underlying dataset is not revealed or where the representativeness of the data is not acknowledged in the analysis of the data (e.g. Bode, 2017). There is a perception that distant reading uses “a massive database that includes ‘everything that has been thought and said’” (Underwood, 2015a). Bode (2017) argues that the work of Moretti and Jockers in particular dominate the discourse about computational approaches in humanities in the academic and public spheres, and as a result, are to be blamed for enhancing this perception. Moretti and Jockers have also been criticised for not revealing their datasets in some of their research (Bode, 2017: 6). The Google Books Ngram Viewer has been criticised for not releasing the bibliography of its data (e.g. Koplenig, 2017) and consequently some of the research done using Google Books Ngram Viewer or dataset have been criticised as the composition of the dataset is not clear (e.g. Koplenig, 2017) .

There is also some speculation about the actual value or usefulness of very large collections for research in the humanities. Henry and Smith (2010: 112) comment that though large-scale digitisation projects are important, it might be that smaller digitisation projects could support research in the humanities more effectively. Kitchin (2014: 10) writes that “Big Data will enhance the suite of data available for analysis and enable new approaches and techniques, but will not fully replace traditional small data studies ... partly due to philosophical positions, but also because it is unlikely that suitable Big Data will be produced that can be utilized to answer particular questions, thus necessitating more targeted studies”.

Lastly, the technology, tools and infrastructure available for research on large collections are criticised. Hoffmann and Evert (2006: 177) note that due to the fact these electronic corpora are typically large and have complex data structures, special tools are often required to do searches in them. Due to the reliance on software tools, the research done using these resources depends to some extent on the quality of the tools and the features offered by them. Terras et al. (2017) conducted a pilot project to identify barriers to complex analysis of large-scale digital collections in the arts and

humanities. Their project showed that there are currently too many technical challenges for individuals in the humanities to analyse large open datasets. They suggest that there should be central service providers for complex queries rather than train researchers themselves. They also found that infrastructure for the analysis of scientific data is not suitable for data from the humanities. Hardie (2012: 380-381) emphasises that it is the combination of corpus and software that enables research, and therefore that more powerful and usable software to analyse corpora should be developed. The uncritical use of key-word searches and basic functionality provided by systems are a concern to some (Gregory, 2014; Hitchcock, 2013). Terras et al. (2017: 463) suggest that funding should be given to develop software engineering capacity to start and maintain digital collections in the humanities. Anthony (2013: 156) also suggests linguists and software engineers should form research teams to develop tools that meet the needs of users. Furthermore, he suggests that the different needs of different users should be taken into consideration. For example, some researchers need tools that can handle annotated corpora and provide sophisticated analysis. Teachers and students do not typically need research tools, but intuitive tools that will quickly give relevant results. Bode (2017: 18) also states that some datasets are only accessible to those with programming expertise, but proposes a curated dataset that has an interface for searching and browsing that makes the data available to all scholars, as well as an option to export data for those interested in analysing trends. While commenting on the Google Books Ngram Viewer specifically, Nicholson (2012: 241) states that it “lacks the sophistication of more advanced corpus analysis tools, the speed and ease with which it allows lone researchers to test new hypothesis makes it a powerful device.” Clearly, there is a need for both sophisticated functions, and also tools that can be used by “lone” researchers or researchers with little programming experience.

Nicholson (2013: 63) states that though the disadvantages of digital research have been well documented, not enough has been done to explore the opportunities and emphasises that it is necessary to start using new methods in research. This researcher agrees and postulates that though there is criticism against some of the new methods for analysing and exploring texts, the availability of large text collections opens up new possibilities that need to be explored. The next section will offer an overview of research done on searching in digital text collections.

2.4. Limitations when searching in text collections

In the previous section, it was highlighted that there are a vast number of digital resources available to researchers and these digital collections are as a result of

digitisation efforts as well as born-digital content. The new opportunities that technology brings to exploring these digital collections were also highlighted. Collections are used more regularly for research and they are not necessarily used in traditional ways. Some tools have been developed to allow researchers to explore these collections. Google Books Ngram Viewer has already been mentioned, and more tools will be discussed in depth in section 2.6. Pioneering work in the field of digital humanities by scholars such as Father Busa (section 1.3.4) is recognised; this study will focus on current developments.

However, despite the availability of large collections and the possibilities that new technologies bring, there have been some concerns and criticism about this type of research. Some of the concerns and barriers in the adoption of the digital humanities have been highlighted, for example, copyright issues, quality of OCR and the validity of new methods. This section will consider some of the research concerning searching in digital text collections.

A study by Terras et al. (2017: 462) into the use of large-scale digital collections in the arts and humanities revealed that a core set of queries resembled most of the information humanities researchers were looking for, namely:

- searching for variants of a word,
- observing the usage of words in context over a specific period of time,
- excluding certain words from a search,
- searching words in close proximity to each other, and
- searching in image metadata.

Much work has been done by HathiTrust Research Center (HTRC) to enable users to explore the large amount of data in the HathiTrust Digital Library and the importance of metadata in large digital collections is recognised (Jett et al., 2016b). One of the tools that will be reviewed in this study is HathiTrust+Bookworm. This tool uses the extensive bibliographic metadata in the HathiTrust Digital Library to allow detailed filtering. One of the recognised challenges in a large digital library is the duplication of records (Jett et al., 2016b). However, Jett et al. (2016b) emphasise that the HTRC has “objectives beyond that of reconciling the digital library's vast collection of metadata” and that their research reveals that “scholars have a great interest in engaging with the HT digital library's corpus at a much finer grained level than that of volumes”. In another study, Jett et al. (2016a) discuss the fact that researchers often need to gather research material together and this ability to gather material into a set “needs to be as

fine as possible, to the point that a workset may comprise individual poems, paragraphs, or even smaller tokens". In order to allow researchers to select smaller segments of texts there is research done to "create a layer of metadata objects that describe finer-grained resources" (Jett et al., 2016b).

In addition to proper bibliographic metadata, there seems to be a desire for more fine-grained metadata. Fenlon et al. (2014) conducted research to find out what researchers' requirements are when working with textual corpora and found that scholars would like access to finer-grained entities than at the level of a volume, for example, chapter, page, poem, etc. (Fenlon et al., 2014: 6). The study by Fenlon et al. (2014: 9) revealed that respondents have a need for "more and better metadata that transcend the conventions of the bibliographic record". It is interesting to note that the study revealed that respondents stated that they are willing to help create and share metadata (Fenlon et al., 2014: 9). In the study by Fenlon et al. (2014: 6) it was found that there is a variety of units that scholars wish to collect for analysis. The following units were identified in this study: abstracts, archaeological data, Archive.org data, book, chapter, genre, image, metadata, narrative, page, page image, paragraph, poem, theme, volume, word, word phrase and XML. It was emphasised that more granular metadata are needed. One of the participants in this study stated: "[T]he book is not a unit of great interest in many cases. In many cases you want to get all the poems that are quoted in the book that are not listed in the metadata. Or whatever [is] of interest to you. So the metadata from the library is very coarse." (Fenlon et al., 2014: 7).

Furthermore, the desired kinds of metadata enrichment were noted as follows: author gender, author names and affiliations, data quality, dates, document format, encoding and typing and different levels of analysis, entities within text, genre, integrated tools and methods of exploration, languages, links to other sources, metadata correction, publication and circulation data, related editions, scholarly annotations (Fenlon et al., 2014: 7).

Underwood (2015b) explains the need for fine-grained metadata in the context of genre metadata. Underwood has done research on automating the process to identify genre, and explains that even if there are reliable metadata about genre available, it might not be enough to support computational analysis of large collections, as volumes are often heterogeneous, consisting of various genres. It would be beneficial if sections in a volume could be differentiated by their genre so that sections of similar genre can be gathered together.

It is this need to go beyond the level of the book or volume that is of particular interest in this study, specifically the way in which texts can be prepared or encoded with metadata to make such smaller sections or properties within a text understandable and usable to a machine so that a user can gather such sections or filter according to such sections.

In addition to preparing texts with deep metadata, the way in which such data are presented to the user is also of paramount importance in this study. One of the aspects highlighted in the previous section is the importance of tools or technology that can work with texts that have been encoded, and the concern that some datasets are not usable by researchers who do not have programming skills was noted. Encoding can be complex and require expert knowledge in itself and Bode (2017) specifically called for a search and browse interface for a curated dataset so that all scholars can use it.

Some tools that allow researchers to explore large-scale textual corpora have already been developed. Before discussing these tools, the researcher will first devote a section to the discussion of metadata. It has already been made clear that the type of metadata that is used in a text collection will have an impact on the type of retrieval that is possible. As such, section 2.5 will consider some levels of metadata that can be applied to texts to identify sections or even tokens to enable fine-grained retrieval in a text collection. Section 2.6 will then discuss some of the current tools used to explore large text collections and in section 2.7 these tools will be evaluated to consider to what extent such fine-grained searching and filtering are allowed and how accessible such searching is to a person without programming knowledge. After reviewing existing practices, the researcher will be able to make a recommendation about detailed encoding and a way in which to retrieve relevant results from such an encoded collection.

2.5. Metadata to enhance retrieval of words or phrases

By enhancing text with metadata, more interesting, and in some cases more reliable, research can be done on large digital text collections. Roller et al. (2016: 69) explain that “the first step towards any information extraction is the definition of information of interest ... [which] is then defined within an annotation schema”.

There are various types of information that can be associated with a text or used to describe a text. For example, there is bibliographic information, such as author, title, genre or language. There is also information about the text or structure of the text, such as chapters, headings, paragraphs or sentences. Other information can also be

encoded, for example, direct speech, dates or lists. There is information about the part-of-speech categories, syntactic structures and semantics for words.

Though there can be large amounts of information available about a text, this research will focus on the following categories of information that can be associated with a text: morphological, syntactic, semantic, functional and bibliographic. They can be considered as different levels or layers of information, moving from a very detailed level of information about words to a higher level of information about the text as an entity.

It is important to note that the metadata elements discussed in this study are not meant to be exhaustive or definitive. Various categories were identified and will be discussed. Depending on the needs of researchers, other elements could be added, elements could be changed or removed. The focus of this study is to demonstrate retrieval on a fine-grained level. The work done in this study should be extensible so that researchers could create customisations according to their needs.

As was mentioned in chapter 1, these different types of information regarding a text can be made explicit through encoding. In this section, each of the levels of information that can be associated with a text will be discussed, as well as how it can be made explicit through encoding.

Much research has been done to use computer programs to automate the encoding of some metadata. This section will also consider research that has been done on this topic. This section is part of the literature review and will only consider what has been reported on automation. In chapter 7 the extent to which computer programs can be used to automate the encoding of texts will be investigated.

Therefore, the discussion of each level of metadata will consist of two sections. The first section will consider the concepts of that level and the second section will consider research that has been done on using computer programs to do the encoding relevant to that level.

2.5.1. Morphological level

Theoretical basis

Morphology refers to the study of words and how they are formed, offering understanding of how a language works, indicating the need for categories of words, examining the internal structure of words, and looking at ways in which words are changed (O'Grady, 2010: 116).

Linguists consider words as minimal free forms, meaning that these elements do not have to occur in fixed positions, but can even stand alone (McGregor, 2009: 58; O'Grady, 2010: 116). Yet, words have internal structures, and morphemes refer to “the smallest unit of language that carries information about meaning or function” (O'Grady, 2010: 117). Simple words have no internal structure, for example, *dog*, *teach* or *run*, whereas, complex words, such as *dogs*, *teacher*, *running*, can be divided into meaningful parts, such as, *dog-s*, *teach-er*, *run-ning* (McGregor, 2009: 58). A complex word can be analysed and be divided into parts (O'Grady, 2010: 119). There are different morphological procedures to be aware of, particularly inflectional, derivational and compounding procedures (Zhou & Marslen-Wilson, 2000: 48). Inflectional and derivational morphology typically combines a stem morpheme (morphemes that can be single words) and a bound morpheme (affixes that cannot stand alone or have independent meaning) (Zhou & Marslen-Wilson, 2000: 48). Compounds are typically formed by combining free stems (where each component is a word in its own right) (Zhou & Marslen-Wilson, 2000: 50). Morphemes can change the part-of-speech category of a word (O'Grady, 2010: 119). For example, the word *teach* is a verb, but adding the morpheme *-er* changes the word to a noun.

Inflection is “the modification of a word’s form to indicate grammatical information” (O'Grady, 2010: 131). In English, inflections are mostly indicated by affixes (O'Grady, 2010: 131), for example, *book* and *books*, or *small*, *smaller* and *smallest*. Grammatical change can also be indicated by internal change or suppletion (O'Grady, 2010: 135-136). Internal change is where one nonmorphemic segment is replaced by another, for example, in *sing* – *sang* or *foot* – *feet*. Suppletion refers to the process where one morpheme is replaced by another, for example in *go* – *went* or *be* – *was* – *were*.

Lexemes are important concepts in morphology and natural language processing, as a lexeme is a unit of meaning for a set of words in different forms (Blevins, 2013: 7). Lexemes are related to lemmas, which are the citation form of an item. Blevins (2013: 7) points out that the lemma is the distinguished form, whereas the lexeme is a set of grammatical words. For example, the lemma (root word) for *run*, *runs*, *running* and *ran* is *run*. The lemma is the form that is entered in the dictionary. As words can be morphologically complex, it is necessary to analyse the structure of words on a morphological level (Hippisley, 2010: 31).

Morphological analysis includes looking at categories of words, also referred to as part-of-speech categories, for example, nouns or verbs. The rest of the section will discuss the concept of parts-of-speech and how it is used in various applications.

A part-of-speech (also known as a word class) is a category of words that show similar grammatical behaviour (McGregor, 2009: 83-85). Some of the most common parts-of-speech classes found in languages across the world are listed in Table 1; however, different languages might have different word classes and have different criteria for a word to be part of a word class (McGregor, 2009: 83-84).

Table 1 Common word classes

Nouns	Adjectives	Pronouns
Verbs	Auxiliaries	Adverbs
Prepositions	Conjunctions	Interjections

Knowing the part-of-speech of a word can provide useful information for other natural language processing tasks (Jurafsky & Martin, 2017: 4). For example, it can give information about the possible neighbouring words and the syntactic structure around the word.

Unfortunately, many words are ambiguous and can have a different part-of-speech in different contexts, for example, *play* in “the children *play* outside” where it is a verb, and “the *play* was performed at noon” where it is a noun. Jurafsky and Martin (2017: 7) explain that based on the data in the Brown corpus and the Wall Street Journal corpus, only about 14 – 15% of words are ambiguous. However, these words are some of the most common words in English and about 55 – 67% (depending on the corpus) in a running text are ambiguous.

A tagset is a list of part-of-speech tags that can be used to label the tokens in a corpus with their appropriate part-of-speech for that context. In tagged corpora, the part-of-speech tag is typically added to the word, for example, the sentence *The beautiful child sang a song* can be annotated as *The/DT beautiful/JJ child/NN sang/VBD a/DT song/NN* to indicate that *The* is a determiner, *beautiful* is an adjective, *child* is noun, *sang* is a verb in the past tense, *a* is a determiner and *song* is a noun. Some English tagsets include more word classes than were indicated in Table 1 (e.g. Jurafsky & Martin, 2017: 3).

There are many different tagsets available. A commonly used tagset developed by Marcus et al. (1993) used to label English corpora is the Penn Treebank tagset. Many corpora have been labelled with this tagset, for example, the Brown corpus, the Wall Street Journal corpus, and the Switchboard corpus (Jurafsky & Martin, 2017: 4). The Penn Treebank tagset contains 45 tags, which includes punctuation (Marcus et al.,

1993). The original set included 48 tags, which included a straight double quote and differentiates between double and single quotes.

Table 2 Penn Treebank tagset

Tag	Description	Tag	Description
CC	conjunction, coordinating	SYM	symbol
CD	numeral, cardinal	TO	"to" as preposition or infinitive marker
DT	determiner	UH	interjection
EX	existential there	VB	verb, base form
FW	foreign word	VBD	verb, past tense
IN	preposition or subordinating conjunction	VBG	verb, present participle or gerund
JJ	adjective or numeral, ordinal	VBN	verb, past participle
JJR	adjective, comparative	VBP	verb, present tense, not 3rd person singular
JJS	adjective, superlative	VBZ	verb, present tense, 3rd person singular
LS	list item marker	WDT	Wh-determiner
MD	modal auxiliary	WP	Wh-pronoun
NN	noun, common, singular or mass	WP\$	Wh-pronoun, possessive
NNP	noun, proper, singular	WRB	Wh-adverb
NNPS	noun, proper, plural	\$	\$
NNS	noun, common, plural	#	pound
PDT	pre-determiner	``	opening quotation mark
POS	genitive marker	"	closing quotation mark
PRP	pronoun, personal	(opening parenthesis
PRP\$	pronoun, possessive)	closing parenthesis
RB	adverb	,	comma
RBR	adverb, comparative	.	sentence terminator
RBS	adverb, superlative	:	colon or ellipsis
RP	particle		

Another well-known tagset is the English CLAWS part-of-speech tagset and is used in the CLAWS (the Constituent Likelihood Automatic Word-tagging System) tagging software (<http://ucrel.lancs.ac.uk/claws/>). The British National Corpus was tagged with CLAWS (<http://ucrel.lancs.ac.uk/claws/>). There are currently eight versions of this tagset (<http://ucrel.lancs.ac.uk/claws/>).

Not all corpora are encoded with the same tagsets (Bird et al., 2015). Due to the variety of tagsets available, various attempts to create a standard tagset have been made. Petrov et al. (2012) have suggested a universal part-of-speech tagset that consists of twelve universal part-of-speech categories that cover the most frequent parts-of-speech that exist in most languages. Such a tagset can facilitate unsupervised tagging, a technique used in automated encoding. The tagset consists of 12 coarse tags. Petrov et al. (2012) also developed a mapping from fine-grained part-of-speech (POS) tags from 25 different treebanks to this universal set.

More recently, the Universal Dependency (UD) project started as an open community effort to create cross-linguistically consistent treebank annotations (Nivre et al., 2016: 1659). This project merged several previous efforts at a universal treebank annotation. UD consists of two layers of annotation, namely a morphological layer and a syntactic layer. The UD morphological layer combines the work done in the Google universal tagset (Petrov et al., 2012), as well as Intersect, “a means of converting among various tagsets in natural language processing” (Intersect, n.d.).

The Universal POS tags from UD are the following 17 tags (Universal Dependencies, n.d.) ADJ (adjective), ADP (adposition), ADV (adverb), AUX (auxiliary), CCONJ (coordinating conjunction), DET (determiner), INTJ (interjection), NOUN (noun), NUM (numeral), PART (particle), PRON (pronoun), PROPN (proper noun), PUNCT (punctuation), SCONJ (subordinating conjunction), SYM (symbol), VERB (verb) and X (other).

Automated encoding of morphological metadata

Much work has been done in the field of natural language processing on identifying words in a text and on identifying their part-of-speech. The next section will consider the work in this field.

Tokenisation

In natural language processing, text preprocessing is when raw text is taken and linguistically meaningful units in the texts are identified, typically words or sentences (Palmer, 2010: 9). The words or sentences that are identified can then be used in numerous natural language processing tasks, such as part-of-speech tagging (Palmer, 2010: 9). These two types of preprocessing, word segmentation and sentence segmentation, will be discussed here.

Word segmentation is the process of identifying words in a piece of text. Palmer (2010: 10) notes that the words identified in this manner are referred to as tokens when used in computational linguistics; as such, this process is often referred to as tokenisation. Issues in tokenisation are language dependent (Manning et al., 2008b). Techniques used in space-delimited languages (e.g. English) are very different to those used in unsegmented languages (e.g. Chinese). Though it would appear obvious to use whitespaces as markers for word boundaries in space-delimited languages, there are many issues that have to be resolved in tokenisation (Palmer, 2010: 16-19). Firstly, the use of punctuation needs careful consideration. Sometimes punctuation marks need to be seen as separate tokens, at other times punctuation needs to be kept when they

occur in words (Jurafsky & Martin, 2017: 13). Some examples where a punctuation mark can have different purposes are noted here: a period can denote a sentence boundary or an abbreviation, an apostrophe can indicate the genitive form (possession) or mark contraction. Secondly, multi-part words are words that consist of multiple grammatical parts. In English a hyphen is used to create words like *end-of-line* or *Boston-based*. Thirdly, multi-word expressions should also be considered. Jurafsky and Martin (2017: 14) argue that is why tokenisation is closely linked to named entity recognition.

Sentence segmentation is the process where the sentences that make up a text are identified (Palmer, 2010: 10). Sentence segmentation is language specific as some languages do not use punctuation marks to indicate sentences (Palmer, 2010: 22). However, even languages that use punctuation marks to denote sentences can present challenges as punctuation marks can often be used for other purposes (Palmer, 2010: 22). Some typical problems include abbreviations, nonstandard sentence ending such as parentheses, ellipses, initials and quotes. Most recent methods for sentence segmentation include machine learning (Jurafsky & Martin, 2017: 17).

The Penn Treebank is a common tokenisation standard. Jurafsky and Martin (2017: 14) explain that this standard separates contractions, keeps hyphenated words and separates punctuation.

Various natural language processing tools or frameworks for English include the ability to do sentence and word segmentation. Tokenisation is typically one of the first steps in the processing pipeline of natural language systems as the output from this stage is used in other tasks (Palmer, 2010: 9).

An example will be given here to illustrate tokenisation. The Stanford Tokenizer, the PTBTokenizer (used in Stanford CoreNLP), was initially designed to mimic the Penn Treebank 3, but has since been enhanced (<https://nlp.stanford.edu/software/tokenizer.html>). It provides a class that focuses on the English language. Figure 3 shows an example of word segmentation from their website (<https://nlp.stanford.edu/software/tokenizer.html>).

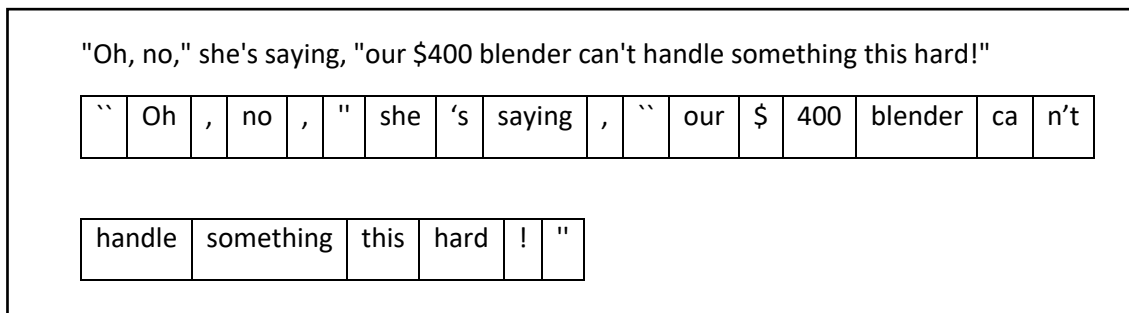


Figure 3 Example of tokenisation (<https://nlp.stanford.edu/software/tokenizer.html>)

Part-of-speech tagging

Once words and sentences have been identified, other tasks such as applying part-of-speech categories to words can be done. Part-of-speech tagging is the process whereby a part-of-speech tag is assigned to a word following specific rules (Jurafsky & Martin, 2017: 6; O'Grady, 2010: 583). A tagging algorithm receives a sequence of words and a tagset as input, and the output is the word and tag pairs (Jurafsky & Martin, 2017: 6). A very simple algorithm is used to assign the most frequent tag for a specific word in the training corpus to an ambiguous word in the text, however, statistical algorithms can achieve much greater accuracy than this baseline algorithm (Jurafsky & Martin, 2017: 7). Statistical algorithms used to determine the correct part-of-speech of a word include Hidden Markov Models, Maximum Entropy Markov Models and log-linear models (Jurafsky & Martin, 2017: 7). Corpora that are labelled with parts-of-speech are very important for training and evaluating statistical tagging algorithms. Current part-of-speech taggers for English are around 97% accurate (Manning, 2011; Petrov et al., 2012).

Tagging typically follows tokenisation in the natural language processing pipeline (Bird et al., 2015); this allows for punctuation to be tagged (Jurafsky & Martin, 2017: 6). As a part-of-speech tag can be useful for stemming (the word class can determine possible morphological structures), it is useful to do it before the stemming process.

The task of part-of-speech tagging for English texts has been investigated extensively and there is a perception that it is a solved problem (Manning, 2011: 1).

Lexical analysis

As was explained previously, a word can sometimes take many forms and the lemma has to be identified. After the part-of-speech category has been identified, the next step

is to change words to their standard form, which is known as normalisation (Jurafsky & Martin, 2017: 13). Two tasks in this stage of language processing are common, namely, stemming and lemmatisation.

Stemming refers to the process where the affixes of words are removed to retain the stem (root) (Jurafsky & Martin, 2017: 16; Manning et al., 2008a; Shwartz, 2016). One of the most well-known stemming algorithms is the Porter algorithm (Jurafsky & Martin, 2017:16; Manning et al., 2008a). This algorithm creates stems based on rules, for example a rule can state that *ate* replace *ational*, which will change *relational* into *relate*. Though a fairly crude method, it is useful when variants of the same lemma need to be identified (Jurafsky & Martin, 2017:17).

Lemmatisation is the process of determining the lemma of a word as it appears in the dictionary (Jurafsky & Martin, 2017: 16; Manning et al., 2008a; Shwartz, 2016).

Lemmas can also include some morphosyntactic information, for example, “*delivers*” is referenced by the item DELIVER + [3rd, Sg, Present]” (Hippisley, 2010: 31). There are two main approaches to morphological parsing. Finite state morphology makes use of finite-state transducers (Hippisley, 2010: 31; Jurafsky & Martin, 2017: 16) and in word and paradigm approaches a lemma is associated with a table where morphological variants of the lemma are associated with a set (Hippisley, 2010: 33).

Some libraries include various algorithms that perform this function, for example, off-the-shelf stemmers such as the Porter and Lancaster stemmers (Bird et al., 2015). It is recommended that a person chooses the stemmer that best suits their purpose.

Natural language processing libraries or toolkits

There are various natural language processing libraries available that are used in personal or commercial applications, as well as for research purposes. Many of the libraries include various functions. Some examples of libraries are the NLTK toolkit (<https://www.nltk.org/>), Apache OpenNLP (<https://opennlp.apache.org>), Stanford CoreNLP (<https://stanfordnlp.github.io/CoreNLP/>) and spaCy (<https://spacy.io>). Each of these libraries/toolkits have tokenisers, part-of-speech taggers and options to identify lemmas.

An example of where tokenisation was used in a research study, is the corpus annotation by Finlayson (2015). The Stanford Tokenizer was used and it is noted that though the tokeniser is extremely accurate, some errors were fixed manually (Finlayson, 2015: 291). The Stanford Part-of-Speech tagger was used by Finlayson

(2015) in the annotation of the corpus for his study. It is noted that some tags were corrected manually.

Much progress has been made in the field of natural language processing. Pinto et al. (2016) even state that for widely spoken languages such as English, there are so many libraries available for lower-level natural language processing tasks that complex applications do not have to be built from scratch. The main task is to select the most suitable library.

It is beyond the scope of this research to do an in-depth evaluation of different natural language processing libraries. However, some comparative or evaluative studies should be discussed.

Evaluations of part-of-speech taggers indicate that current taggers perform at slightly over 97% (Manning, 2011: 1). A number of systems and techniques used for part-of-speech tagging and their performances (with references) are listed in the ACL wiki for part-of-speech tagging (ACL Wiki, 2019). In terms of part-of-speech tagging for English texts, Manning (2011: 1) notes that there is a perception among computational linguistics that the performance of current systems cannot really be improved upon.

In terms of sentence segmentation, Read et al. (2012) note that this is not regarded as one of the grand challenges of natural language processing and the relatively few studies about this topic have been done recently. They compared nine publicly available systems and toolkits with sentence segmentation components. The tests were performed over a sample of edited English corpora. When looking at the average over all the corpora, the best performing systems were RASP and tokenizer, with a score of 97.6% each (Read et al., 2012: 991). A similar study was done by Griffis et al. (2016), where five popular off-the-shelf NLP toolkits were evaluated in terms of sentence segmentation across different corpora, specifically including material from the clinical domain. Toolkits perform extremely well on well-formed text (F-scores of up to 0.99), but, as expected, performance deteriorates on texts that include many abbreviations, forms of shorthand and are ungrammatical.

Al Omran and Treude (2017) compared four natural processing libraries, namely Google's SyntaxNet, Stanford CoreNLP, NLTK, and spaCy, to determine which library performed best for analysing software artefacts written in natural language. Their study focused on tokenisation and part-of-speech tagging. They found that 91% of the tokens identified by the four libraries were the same and 64% of the tokens were given the same part-of-speech tag. Compared to manual annotation, spaCy achieved the best

results for texts from two sources and SyntaxNet worked best on texts from the third source.

Another comparison between natural language processing libraries was done by Pinto et al. (2016). In this study NLTK, Apache OpenNLP, Stanford CoreNLP, Pattern, TweetNLP, TwitterNLP and TwitIE were compared using newspaper and social network texts. They conclude that there is no one toolkit that outperforms the others in all situations. They recommend OpenNLP for news texts and TwitterNLP for social media texts.

The reason why a particular natural language processing library was selected to use in a project is not always clear. A study was done by Al Omran and Treude (2017) to compare different natural language processing toolkits. In their study they found that in only 14% of the papers they investigated was the specific library mentioned and none provided a thorough justification for the selection of the specific library. This attitude to natural language processing libraries was noticed in the study by Finlayson (2015), as no reason was given for selecting the Stanford tools for certain annotation tasks.

From the preceding discussion it is clear that not all natural language processing libraries produce similar results. Al Omran and Treude (2017) found that the choice of the best natural language processing library depends on the task that should be performed as well as the nature of the text that is being analysed. It should also be noted that for some types of encoding there are fairly stable and advanced tools available.

2.5.2. Syntactic level

Theoretical basis

Syntax refers to the rules regarding the structure of sentences in a language and how words are arranged in a sentence.

There are two main types of grammars that are used to do syntactic analysis, phrase structure grammar (or constituency grammar) and dependency grammar (Leech et al., 1996).

The focus of phrase structure grammars is constituency relations, meaning how words (or symbols in a language) are combined. Constituents are groups of words that form units (Bird et al., 2015). The concept of constituents is evident from the fact that groups of words can be replaced with another word, for example, in the sentence *the little boy saw the worm* the words *the little boy* can be replaced with *he*.

Grammar rules express how the symbols (or constituents) of a language can be grouped or arranged (Jurafsky & Martin, 2017). For example, a rule can express that a sentence can consist of a noun phrase and a verb phrase, another rule could express what a noun phrase can consist of, and so forth. An example of a parse tree expressed according to phrase structure grammar is shown in Figure 4.

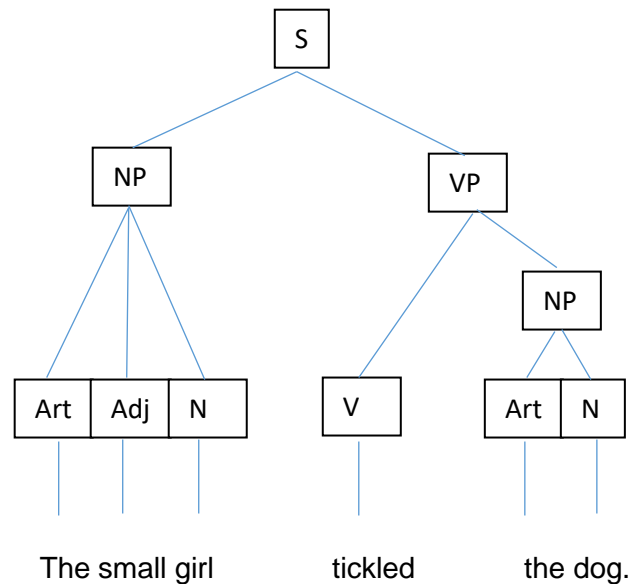


Figure 4 A parse tree showing phrase structure grammar

Syntactic analysis can be ambiguous. A well-known example of an ambiguous sentence is *I shot an elephant in my pajamas*. The sentence can be analysed in two ways depending on whether the phrase *in my pajamas* describes the elephant or the shooting event. Another example with different interpretations is *Time flies like an arrow*.

Phrase structure grammars are used in many applications, such as grammar checking, semantic interpretation and machine translation (Jurafsky & Martin, 2017).

The focus of dependency grammars (the other main type of grammar) is dependency relations, meaning how words relate to each other. According to Bird et al. (2015) “dependency is a binary asymmetric relation that holds between a head and its dependents”. Arrows with the name of the grammatical relations are typically used to illustrate the dependencies (Manning, 2014). The arrow is used to connect the head (governor) with the dependent (modifier) (Manning, 2014).

For example, the sentence *Anne loves chocolate* has the following relations:

loves —>_{subj} Anne

loves —>_{obj} chocolate

This could be expressed as *Anne* and *chocolate* depend on *loves*, *Anne* and *chocolate* modifies *loves* or *loves* governs *Anne* and *chocolate*.

Another example from Jurafsky and Martin (2017) shows a sentence analysed using dependency grammar:

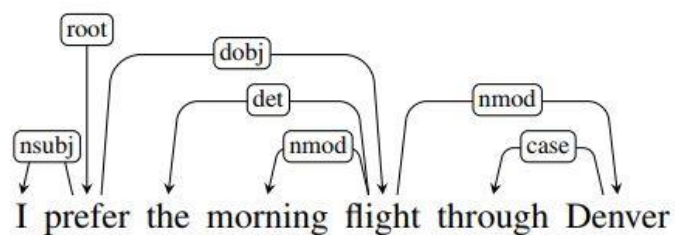


Figure 5 A sentence analysed according to dependency grammar (Jurafsky & Martin, 2017)

In this example, *prefer* is the root of the sentence. *I* is a nominal subject and dependent on *prefer*. *flight* is a direct object and dependent on *prefer*. *the* is a determiner and dependent on *flight* and *morning* is a nominal modifier and dependent on *flight*. *through* is a preposition and is governed by *Denver*, where *Denver* is a nominal modifier which modifies *flight*.

The relationships in dependency analysis make certain information directly available that is often hidden in complex phrase structure trees (Jurafsky & Martin, 2017). There is no tree structure as in phrase structure grammar, there are direct binary links between words. Apart from indicating head-dependent pairs, dependency grammars can also indicate the grammatical function a word plays with respect to its head (Jurafsky & Martin, 2017).

Many types of relationships have been identified. The Universal Dependencies framework was already mentioned in the previous section. The Universal Dependencies propose “universal grammatical relations which can be used with relative fidelity to capture any dependency relation between words in any language” (Universal Dependencies – v1, n.d.). These relationships as identified in version 1 of the Universal Dependencies are listed in Table 3.

Table 3 Universal Dependency Relations

acl: clausal modifier of noun (adjectival clause)	expl: expletive
advcl: adverbial clause modifier	foreign: foreign words
advmod: adverbial modifier	goeswith: goes with
amod: adjectival modifier	iobj: indirect object
appos: appositional modifier	list: list
aux: auxiliary	mark: marker
auxpass: passive auxiliary	mwe: multi-word expression
case: case marking	name: name
cc: coordinating conjunction	neg: negation modifier
ccomp: clausal complement	nmod: nominal modifier
compound: compound	nsubj: nominal subject
conj: conjunct	nsubjpass: passive nominal subject
cop: copula	nummod: numeric modifier
csubj: clausal subject	parataxis: parataxis
csubjpass: clausal passive subject	punct: punctuation
dep: unspecified dependency	remnant: remnant in ellipsis
det: determiner	reparandum: overridden disfluency
discourse: discourse element	root: root
dislocated: dislocated elements	vocative: vocative
dobj: direct object	xcomp: open clausal complement

This table does not include subtypes (e.g. poss for possessive, prt for particles, tmod for temporal modifiers).

Schuster and Manning (2016) suggest that there is need for enhanced dependencies and present the enhanced English Universal Dependency representation. The enhanced representation makes implicit relations between words more explicit and adds information to widely used, but uninformative, labels (Schuster & Manning, 2016: 2372; Universal Dependencies, n.d.). Enhanced graphs include all the information of the basic Universal Dependency graphs, but add additional information. Some enhancements are discussed here, but see Universal Dependencies (n.d.) for more information. For example, modifiers are augmented to include the type of modifier (e.g. nmod: on). Similarly, conjunctions are augmented (e.g. conj: and). Furthermore, in conjoined phrases the relations between all the conjuncts are made explicit (e.g. in the case of conjoined noun phrases, each noun phrase becomes the subject of the main verb) (Schuster & Manning, 2016).

Dependency grammar is often used in languages where there is greater freedom in terms of word order (Leech et al., 1996). Dependency grammars are used in contemporary speech and language processing systems (Jurafsky & Martin, 2017).

A treebank is “a corpus that has been annotated for syntactic structure according to some syntactic information” (O’Grady, 2010: 583). A treebank is often developed on a corpus that has already been annotated with part-of-speech information. A well-known

example of a treebank is the Penn Treebank for English, which is tagged with part-of-speech information and also parsed with phrase structure grammar.

In the Google Books Ngram Viewer the texts are parsed and the n-grams are then annotated with part-of-speech and modifier dependencies (Lin et al., 2012). The Google Books team have labelled both types of annotations “syntactic”. However, in this study, a distinction will be made between the levels and part-of-speech tagging will be seen as part of the morphological level and dependencies will form part of the syntactic level.

This study will consider the use of dependency grammar to improve retrieval in a text collection.

Automated encoding of syntactic metadata

Syntactic parsing is an important part of natural language processing (Gómez-Rodríguez et al., 2019: 2). A parser is “a computer program that analyses the syntactic structure of a sentence” by using a set of rules that describes the language (O’Grady, 2010: 584). In recent years parsing has been widely used in several artificial intelligence applications, such as machine translation, information extraction or sentiment analysis (Gómez-Rodríguez et al., 2019: 2).

Dependency grammar is the predominant representation used by artificial intelligence applications. Many dependency parsers are now available publicly (Choi et al., 2015: 387). Most dependency parsing systems can be grouped into two broad categories, graph-based and transition-based (shift-reduce) parsers (Gómez-Rodríguez et al., 2019: 3). Graph-based parsers “use models that score dependency relations or groups of them, and perform a global search for a parse that will maximize the combined score of all dependencies” whereas transition-based parsers are “based on a state machine that builds syntactic analyses step by step” (Gómez-Rodríguez et al., 2019: 3-4). Parsers can require a significant amount of computational resources, which becomes particularly concerning in large-scale applications (Gómez-Rodríguez et al., 2019: 2).

There are various parsers available, some of which are part of natural language processing libraries. Examples of dependency parsers are Turbo (<https://github.com/andre-martins/TurboParser>), RBG (<https://github.com/taolei87/RBGParser>), spaCy, ClearNLP (https://github.com/clir/clearnlp-guidelines/blob/master/md/components/dependency_parsing.md) and Stanford Parser. The parser in the OpenNLP library is a constituency parser.

The standard metrics to evaluate the accuracy of a dependency parser are the unlabelled attachment score (UAS), which is the portion of words linked to the correct headword regardless of whether the label of the arc is correct; the labelled attachment score (LAS), which is the portion of words linked to the correct headword and the correct dependency has been identified; and the label accuracy score (LA or LS) which is the portion of words where the correct dependency has been identified (Choi et al., 2015: 387; Gómez-Rodríguez et al., 2019: 5).

Annotated corpora are used to train and evaluate natural language processing applications (Berzak et al., 2016: 1). In order to save time, such corpora are created by firstly annotating the texts by using automated methods, after which they are corrected manually (Berzak et al., 2016: 1). According to Marcus et al. (1993) this approach is not only quicker, but also results in annotations that are of higher quality than creating annotations from scratch. However, a study by Berzak et al. (2016) suggests that anchoring (the natural bias towards a pre-existing value) could lead to parser bias in the annotation. This effect could lead to the over-estimation of the performance of natural language processing tools and suggest a “more extensive investigation of tagger and parser evaluation in NLP” (Berzak et al., 2016: 2).

One of the observations in the study by Berzak et al. (2016) is that where different parsers give the same annotations, it does not significantly lower the quality of human annotations. As a result, they suggest a hybrid annotation strategy, where “human annotators review annotations on which several parsers agree and then complete the remaining annotations from scratch” (Berzak et al., 2016: 8).

Though the main aim of the study by Berzak et al. (2016) was not to compare NLP libraries, it is interesting to note the performance of the two libraries used in their study to investigate parser bias. Two combinations of taggers and parsers were used. (It makes sense to evaluate combinations of taggers and parsers, as the part-of-speech tags assigned will influence the syntactic parsing.) The Turbo tagger and parser and the Stanford tagger and the RBG parser were used. Compared to human annotations the performance of the Turbo-Turbo output is POS 95.32, UAS 87.29, LA 88,35 and LAS 82.29. The performance of the Stanford-RBG output is POS 95.59, UAS 87.19, LA 88,03 and LAS 82.05. It is interesting to note that the performance of the libraries is comparable and is relatively good.

A parser accuracy study was conducted by Gómez-Rodríguez et al. (2019). They tested well-known syntactic parsers in a syntax-based sentiment analysis system. The parsers were the MaltParser, Stanford RNN parser, TurboParser and YaraParser. In

their experiment the YaraParser achieved the highest LAS, UAS and LA scores. However, they note that there is a trade-off between speed and accuracy.

2.5.3. Semantic level

Theoretical basis

Semantics is the study of the meaning of words. A word can have different meanings (senses), for example, *fall* could refer to the season (autumn) or to a downward movement. A word sense is “a discrete representation of one aspect of the meaning of a word” (Jurafsky & Martin, 2017).

Important concepts when regarding the meaning of words (the senses of words) are homonymy and polysemy (Goddard & Schalley, 2010; Jurafsky & Martin, 2017: 95). Homonyms are senses that are unrelated (O'Grady, 2010: 206), for example, *bat* that refers to an instrument for hitting and *bat* that refers to an animal, or *well* that has multiple senses (meanings), including referring to a shaft in the ground from where water is obtained or to good health. Polysemy refers to words that have different senses, but are related (O'Grady, 2010: 206), for example, *bank* that refers to a financial institution and *bank* that refers to the building of the institution. Both phenomena are problematic for natural language processing (Goddard & Schalley, 2010). Word sense disambiguation is the process of finding out which sense of a word is being used in a particular context (Jurafsky & Martin, 2017).

It is also important to consider concepts that explain the relationships between senses, such as, synonymy, antonymy, hyponymy, hypernymy, meronymy and holonymy (Jurafsky & Martin, 2017). Words are synonyms when they have similar meanings and sometimes can act as a substitute for each other (Bird et al., 2015; Jurafsky & Martin, 2017; O'Grady, 2010: 204). It should be noted though, that “perfect” or “full” synonyms are extremely rare, if they exist at all (Taylor, 2003: 265). It is more common to find words that are “near synonyms”, in other words, words that display “a low degree of implicit contrastiveness”, for example, *little/small*, *high/tall*, *start/begin*, *stop/finish* (Taylor, 2003: 266). Antonyms are words that are “opposites with respect to some component of their meaning” (O'Grady, 2010: 205), for example, *up* and *down* are opposite with respect to direction, but both relate to movement. Hyponym and hypernyms denotes an “is-a” relationship between a specific type of a general type, for example, apple is a hyponym of fruit (Loria, 2013: 205; O'Grady, 2010). Meronymy and holonymy denotes a “part-whole” relationship, for example, *wheel* (meronym) is part of a *car* (holonym) (Loria, 2013: 205; O'Grady, 2010).

One of the most well-known resources for semantic analysis in English is WordNet. WordNet is a large lexical database of English, where words expressing specific concepts are grouped into synsets (WordNet, n.d.). WordNet resembles a traditional thesaurus, but has a richer structure (Bird et al., 2015). WordNet for English was developed at Princeton University, but wordnets in other languages have since been developed. There are currently more than 70 wordnets in different languages (The Global WordNet Organization, n.d.).

Figure 6 shows the senses for the noun *fall* (the rest of the data are not in the screen capture). Each sense shows the list of synonyms (the synset) for that sense, as well as a gloss (short definition) and example sentences for some senses. WordNet has about 117 000 synsets that are linked to other synsets through relations (WordNet, n.d.).

The relationships between concepts in WordNet form a hierarchy (Loria, 2013). WordNet can express hypernymy and hyponymy (is-a) relationships, as well as meronymy (the part-whole relation) (WordNet, n.d.). The beginning of all the noun hierarchies is the root node [entity].

WordNet Search - 3.1

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options: (Select option to change)

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- [S:](#) (n) [fall](#), [autumn](#) (the season when the leaves fall from the trees) "*in the fall of 1973*"
- [S:](#) (n) [spill](#), [tumble](#), [fall](#) (a sudden drop from an upright position) "*he had a nasty spill on the ice*"
- [S:](#) (n) [Fall](#) (the lapse of mankind into sinfulness because of the sin of Adam and Eve) "*women have been blamed ever since the Fall*"
- [S:](#) (n) [descent](#), [declivity](#), [fall](#), [decline](#), [declination](#), [declension](#), [downslope](#) (a downward slope or bend)
- [S:](#) (n) [fall](#) (a lapse into sin; a loss of innocence or of chastity) "*a fall from virtue*"
- [S:](#) (n) [fall](#), [downfall](#) (a sudden decline in strength or number or importance) "*the fall of the House of Hapsburg*"
- [S:](#) (n) [fall](#) (a movement downward) "*the rise and fall of the tides*"
- [S:](#) (n) [capitulation](#), [fall](#), [surrender](#) (the act of surrendering (usually under agreed conditions)) "*they were protected until the capitulation of the fort*"
- [S:](#) (n) [twilight](#), [dusk](#), [gloaming](#), [gloam](#), [nightfall](#), [evenfall](#), [fall](#), [crepuscule](#), [crepuscle](#) (the time of day immediately following sunset) "*he loved the twilight*"; "*they finished before the fall of night*"
- [S:](#) (n) [fall](#), [pin](#) (when a wrestler's shoulders are forced to the mat)
- [S:](#) (n) [drop](#), [fall](#) (a free and rapid descent by the force of gravity) "*it was a miracle that he survived the drop from that height*"
- [S:](#) (n) [drop](#), [dip](#), [fall](#), [free fall](#) (a sudden sharp decrease in some quantity) "*a drop of 57 points on the Dow Jones index*"; "*there was a drop in pressure in the pulmonary*"

Figure 6 The word fall in WordNet

Automated encoding of semantic metadata

Word sense disambiguation (WSD) refers to the task of identifying which sense (meaning) of a word is used in a specific context by using computational methods (Moro et al., 2014: 231; Navigli, 2009: 4; Ustalov et al., 2018: 1). It is one of the areas of research in the field of Natural Language Processing (Raganato et al., 2017: 99) and has been regarded as an AI-complete problem (Navigli, 2009: 2). There are various benefits of knowing the sense of a word; for example, it can help when dealing with large-scale data and machine translation (Navigli, 2009: 2-3). The methods used to do WSD are categorised in three main types, namely, knowledge-based, supervised and unsupervised (Moro et al., 2014: 232).

Knowledge-based approaches rely on machine-readable knowledge sources (Navigli, 2009: 30; Raganato et al., 2017: 100), such as semantic networks, dictionaries and

thesauri (Iacobacci et al., 2016: 898). WordNet is one such machine-readable dictionary (Agirre et al., 2018: 1). These resources are used to identify suitable meanings (Iacobacci et al., 2016: 899). Some of the techniques used in knowledge-based WSD are the LESK algorithm, structural approaches, selectional preferences (Navigli, 2009).

In supervised approaches, machine learning techniques use manually sense-annotated data to create classifiers (Iacobacci et al., 2016: 898; Navigli, 2009: 15-16; Raganato et al., 2017: 100). Supervised approaches perform well (Agirre et al., 2018: 2; Navigli, 2009: 16; Yuan et al., 2016: 2). However, they require a large amount of data which is expensive to build (Agirre et al., 2018: 2; Moro et al., 2014: 232; Yuan et al., 2016: 2). Some of the techniques used in supervised WSD are decision lists, decision trees, Naïve Bayes and neural networks (Navigli, 2009). Due to the expense of creating large annotated corpora, some methods have begun to use unlabelled corpora as well to overcome the knowledge acquisition bottleneck of conventional supervised models, and are called semi-supervised methods (Raganato et al., 2017: 100).

Unsupervised approaches do not rely on external knowledge resources or sense-annotated corpora (Navigli, 2009: 26). The assumption is made that similar senses occur in similar contexts (Iacobacci et al., 2016: 898; Navigli, 2009: 26). Words are clustered together and senses are induced (Iacobacci et al., 2016: 898; Navigli, 2009: 26). This approach is also known as word sense discrimination (Navigli, 2009: 26). The main methods used in unsupervised WSD are context clustering, word clustering, and co-occurrence graphs (Navigli, 2009).

Various WSD implementations have been done. Some examples will be mentioned here, but it is by no means an exhaustive list. IMS (It Makes Sense) is a supervised system that allows users to integrate additional features and different classifiers (Zhong & Ng, 2010). DKPro is a framework for word sense disambiguation which implements multiple WSD methods (Miller et al., 2013). Babelfy is a system based on the BabelNet that implements a multilingual graph-based approach to entity linking and WSD (Moro et al., 2014). UKB is a graph-based system which makes use of random walks over a semantic network (Agirre et al., 2014). The PyWSD project is the Python implementation of popular WSD methods (<https://pypi.org/project/pywspd/>).

In the study by Finlayson (2015), WordNet version 3.0 was used for semantic annotation. It is noted that though some WSD can perform well, it was not sufficient for the particular study, and annotation of word senses was done manually.

The various WSD approaches should also be compared and evaluated. However, due to various techniques, datasets and knowledge resources used, it is difficult to compare systems. In order to address this issue and to test and compare different computational semantic analysis systems, the SemEval competition was established (https://aclweb.org/aclwiki/SemEval_Portal). Each competition has different tasks that participants can choose to address (<http://alt.qcri.org/semEval2019/>). This event has evolved and now includes many different aspects of semantic analysis.

Raganato et al. (2017) also developed an evaluation framework to enable fair comparison among WSD systems. In this framework, datasets and training corpora were standardised into a unified framework, datasets are semi-automatically converted to WordNet 3.0, and datasets are preprocessed by using the same pipeline. This framework includes some of the datasets from the SemEval competition. This framework was then used to compare some of the main techniques proposed in the WSD literature. They evaluated three supervised WSD systems (IMS, IMS+emb, Context2Vec) and three knowledge-based WSD systems (Lesk, UKB, Babelfy). According to their evaluation, supervised systems consistently outperform knowledge-based systems across datasets. The best overall results were achieved by the IMS+emb system trained on the SemCor+OMSTI corpus, with the Context2Vec system (trained on the same corpus) a close second. However, their evaluation has been criticised by Agirre et al. (2018), who argue that they used UKB with suboptimal default parameters. Agirre et al. (2018) used the same dataset as Raganato et al. (2017), but used the optimal settings for UKB as described in the documentation. In this evaluation UKB outperforms other knowledge-based systems. They do note that supervised systems sometimes still perform slightly better than knowledge-based systems (Agirre et al., 2018).

2.5.4. Functional level

Theoretical basis

A text can contain different textual phenomena. It is useful to discuss the different textual phenomena according to categories. This research will follow categorisation suggested by Van den Branden et al. (2017). They categorise the different textual phenomena as structural, renditional, logical and analytical.

Structural features consist of those features that are used to organise a text, for example, chapters, sections, and paragraphs (Van den Branden et al., 2017). When working with digital material it is not necessarily useful to think in terms of the properties of printed material. For example, printed material works with pages, but a

digital text in HTML format is a long continuous piece. Although pages are superfluous in these situations, the texts are often still structured. There could be different sections with headings. Paragraphs are still used and there could be a distinction between the content and front and back matter.

Renditional features refer to the way a text is presented (Van den Branden et al., 2017). Sometimes typographic features (font, size, hue, etc.) are used in a text in order to indicate visually that a section of text is different in some way or should stand out for some reason.

The logical (or semantic) category relates to the meaning of words or phrases (Van den Branden et al., 2017). Some features of a text are identified through reading and understanding a text. For example, text could include direct speech, names, titles, dates, measures.

Other features can be assigned to the text after analysing the text (the analytical category), for example, editorial corrections or notes, the regularisation of text (Van den Branden et al., 2017).

These textual phenomena can be made explicit through markup or encoding. In this research these textual phenomena will be called the functional level of encoding.

If this implicit information in a text has been made explicit through encoding, it can be processed by a machine (Cummings, 2013). Texts that are encoded can be analysed, searched and their relationship with other texts can be indicated (Drucker, 2013).

Although analysis of plain text can reveal some information, for fine-grained questions the text would need to be encoded (Mason, 2015). For example, if a researcher wanted to compare the words in soliloquies to the words used in dialogues in Shakespeare's plays, the text would need to be encoded to differentiate between the two categories.

The most widely used language to represent texts in digital form is TEI (TEI – Text Encoding Initiative), which is developed and maintained by the Text Encoding Initiative (TEI) consortium (TEI – Text Encoding Initiative, 2016). The TEI Guidelines are the recommendations published by the TEI for the encoding of texts (TEI – Text Encoding Initiative, 2016). The most recent version of the guidelines is called P5 with the latest release (4.0.0) made available in February 2020 (TEI – News, 2020).

The TEI Guidelines are expressed as a modular, extensible XML schema. TEI contains elements to represent a wide range of textual phenomena. For example, it can be used to define paragraphs, page breaks, words that need to be italicised or quotes. TEI P5

includes over 500 elements (Van den Branden et al., 2017). These elements are organised into 21 modules. For example, the TEI Header is a module that includes elements to describe bibliographic data, whereas the Performance Texts module includes elements specifically to describe drama material.

The following is an example of a TEI document from Van den Branden et al. (2017).

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>A sample TEI document</title>
      </titleStmt>
      <publicationStmt>
        <publisher> KANTL </publisher>
        <pubPlace>Ghent</pubPlace>
        <date when="2009"/>
      </publicationStmt>
      <sourceDesc>
        <p>No source, born digital</p>
      </sourceDesc>
    </fileDesc>
  </teiHeader>
  <text>
    <body>
      <p>This is a sample paragraph, illustrating
      a <nametype="organisation">TEI</name> document.</p>
    </body>
  </text>
</TEI>
```

Figure 7 Example of a TEI document (Van den Branden et al., 2017)

It is interesting to note that the header of a TEI document stores information about the original, physical text as well as the digital copy. However, Cummings (2016) explains that the header was designed to store two kinds of information and as such the TEI is typically used slightly differently by different groups of people. Firstly, metadata stored in the header can contain bibliographic information similar to the information in a library catalogue and is used by librarians and bibliographers. Secondly, editors store information about encoding practices in the header. This results in TEI headers that have different styles. The “librarian’s header” tends to contain more structured data so that complete bibliographic descriptions can be obtained, whereas the “editor’s header” is concerned with editorial principles and the information is often ad-hoc and individualistic.

Apart from more bibliographic information in the header, the elements used in the text are used to describe the text itself. An example from the Walt Whitman archive will be

used to illustrate how TEI can be used to describe the characteristics of a text (The Walt Whitman Archive, n.d.). A letter written by Whitman is shown in Figure 8.

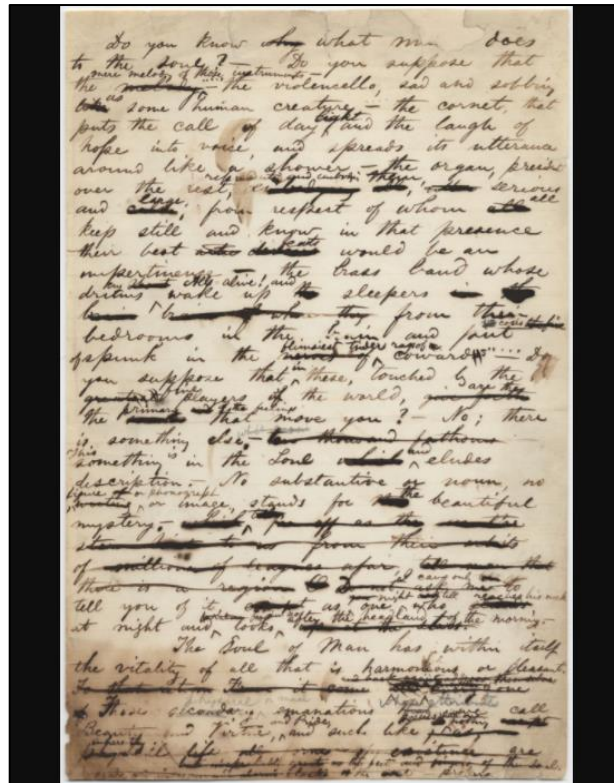


Figure 8 A letter by Walt Whitman (The Walt Whitman Archive, n.d.)

The TEI shows how different features are handled, for example, text that has been crossed out (overstrike), is illustrated in Figure 9.

```

<text type="manuscript">
  <body>
    <pb xml:id="leaf001r" facs="tex_jc.00037.jpg" type="recto"/>
    <note type="editorial" resp="#h3" place="top" rend="circled">1</note>
    <p>
      <handShift new="#h1"/>
      <seg xml:id="q01">
        Do you know
        <del rend="overstrike">why</del>
        what m
        <unclear reason="cut away" cert="high" resp="#kc">usic</unclear>
        does to the soul?—Do you suppose that the
        <subst>
          <del rend="overstrike" seq="1">melody</del>
          <add rend="insertion" place="supralinear" seq="2">mere melody of those instruments—</add>
        </subst>
        <add rend="unmarked" place="supralinear">
          <seg xml:id="w2">. . . .</seg>
        </add>
        <seg xml:id="w1"></seg>
        <alt target="#w1 #w2" weights="0.5 0.5"/>
        the violencello, sad and sobbing
      </seg>
    </p>
  </body>
</text>

```

Figure 9 The letter encoded in TEI

Lastly, the digital representation from the TEI document is shown in Figure 10.

Do you know why what music does to the soul?—Do you suppose that the mere melody of those instruments
 —the violoncello, sad and sobbing like some human creature—the cornet, that puts the call of day
 and the laugh of hope into voice, and spreads its utterance around like a shower—the organ, president over
 the rest, embodying representing and embodying all, with them, serious and calm, large, from respect of whom all keep
 still and know in that presence their best feats would be an impertinence—the brass band whose drums
 cry shout All-alive! and wake up the sleepers in the brain of where they from their bedrooms in the brain and put red

Figure 10 The digital representation of the letter

TEI has made a significant impact on digital scholarship and is internationally recognised as an important tool (UCLA Research Guide, n.d.). The TEI maintains a list of projects that make use of TEI that includes close to 200 projects (TEI – Projects Using the TEI, 2017). Organisations such as the US National Endowment for the Humanities, the UK's Arts and Humanities Research Board, the Modern Language Association, the European Union's Expert Advisory Group for Language Engineering Standards, and the US Library of Congress have endorsed the use of TEI (UCLA Research Guide, n.d.).

Particularly referring to the use of TEI in corpora for linguists, Hardie (2014: 77) argues that TEI is fairly top-heavy and requires an extensive amount of encoding. He discusses some practices that have become the *de facto* standard in encoding linguistic corpora and lists the most common elements and attributes that are being used in practice when encoding corpora for linguistic analysis (Table 4).

Table 4 Common TEI elements

Tag	Use	Tag	Use
<p>	Paragraph	<event>	Sound break on a recording
<s>	Sentence	<stage>	Stage directions
<head>	Heading	<text>	Tag that enclosed the corpus
<pb/>	Page break	<unclear>	Mark something that cannot be heard or seen clearly
<q>	Quoted text	<header>	Metadata about the text
<gap>	Something that has been omitted	<body>	Start of the actual text
<reg>	Regularised spelling	<w>	Word
<u>	Utterance	<c>	Punctuation
<pause>	Pause in spoken text	<anon>	Used to anonymise data
<voc>	Non-linguistic vocalisation		

The “modest level of XML” proposed by Hardie (2014) that originated in TEI was used to encode the 2014 version of the British National Corpus (Love et al., 2017: 339).

It is furthermore important to note that TEI can be customised. Different TEI customisations can be created to suit different projects. Customisation in TEI is the process of selecting or eliminating certain aspects of the standard, or even adding specific features (Bauman & Flanders, 2018). Bauman and Flanders (2018) argue that there are two main reasons for customisation in TEI. Firstly, it is to address the diverse needs of the different people that make use of TEI. Furthermore, the need for a common encoding language and an expressive language is addressed by customisation.

One example of customisation is from the Women Writers Project. In this project the element docRole is used instead of docAuthor, as the developers wanted to note other ways than authorship in which people can contribute to documents (Bauman & Flanders, 2018). Another example of customisation given by TEI, is where the requirements of the div element is changed so that the type attribute (which is inherited) is made mandatory (a fixed set of values for this attribute is provided) (TEI – Text Encoding Initiative, n.d.).

TEI P5 provides some example customisations that meet the needs of certain groups of people (TEI – Text Encoding Initiative, n.d.). For example, TEI Lite, is a widely used customisation which includes basic elements for simple documents; TEI Bare contains the minimal number of elements for encoding; and TEI Corpus includes the necessary information to encode linguistic corpora.

There is some criticism regarding TEI. Encoding is an act of interpretation and is not always consistent (Drucker, 2013). This means that, at some point, a person makes a

judgement call about what a specific piece of text is. For example, should it be encoded as a paragraph, heading or quote? In some instances, people might disagree on the type of encoding to use for a specific section in the text. The tags themselves are also not necessarily used consistently, and even an individual who works on different days, might use tags differently (Drucker, 2013).

The hierarchical nature of XML, and as such TEI, can be problematic. In texts, there are often overlapping hierarchies, for example a poem can be printed across two pages. Both these ideas cannot be represented in XML at the same time (Drucker, 2013). TEI therefore often focuses on the content, as opposed to the physical features (Drucker, 2013).

Some of these issues with TEI contribute to the gap between the encoding and analysis of text. Tools to analyse texts encoded in TEI have not developed as well as the content model itself (Mason, 2015). Bauman et al. (2012) contend that the process of encoding for publishing and the analysis process are very different, and sometimes encoding in a document can hinder analysis. They note the following problems regarding analysing texts that have been encoded with TEI:

- arbitrary encoding cannot be handled well by analysis tools
- as a result of customised encoding, tools have to be tweaked for different collections
- analysts typically do not have a detailed knowledge of the encoding that was followed in a particular project, but require in-depth understanding in order to do effective analysis
- even to extract a simple word list can be challenging as there might be some elements that should be ignored, such as <choice> when the editor suggests an alternative word
- encoders will encode what is of interest to them, not necessarily what is useful for analysis, for example one encoding project might encode place names, and another might encode proper names of people

Consequently, Bauman et al. (2012) note “scholars interested in text analysis typically find it is more efficient to use plain texts without any encoding, and then apply an ad hoc system of manual tagging, or named entity recognition (NER) tools in combination with manual correction of tagging”.

Some projects that use TEI include:

- The Perseus Digital Library: <http://www.perseus.tufts.edu/hopper>
- The Newton Project: <http://www.newtonproject.ox.ac.uk/>
- The Mark Twain Project: <http://www.marktwainproject.org/>

Automated encoding of functional metadata

Much of the encoding at the functional level is subjective and requires a human encoder. According to Qin et al. (2018: 35), scanned documents require humans to markup the content with metadata, which is “an expensive, tedious, and slow process” (Qin et al., 2018: 35). However, with the advances in software development, some of the encoding of the characteristics of a text could be automated. In this section, some of the research on automatic text analysis will be discussed. Firstly, research done to identify structures (e.g. headings and paragraphs) in the text will be discussed. Next, research done to identify direct speech and entities in the text will be discussed. Lastly, the automated detection of language used in a text will be discussed.

Document layout analysis (or document image analysis or document semantic structure extraction) is the area of research that aims to identify sections in a document automatically (Cristani et al., 2018: 71). Document layout analysis can be divided into two parts, namely geometric layout analysis (or physical layout analysis or page segmentation) and logical layout analysis (or logical structure analysis) (Cristani et al., 2018: 71; Qin et al., 2018: 35; Yang et al., 2017: 5315). The purpose of geometric layout analysis is to determine homogeneous text and non-text regions in a document (in other words, automatically identifying images and areas of text) (Cristani et al., 2018: 71; Grana et al., 2016: 4; Qin et al., 2018: 35). During logical layout analysis the regions (or sections) that were identified in the geometric layout analysis are described from a logical perspective and typically labels are assigned, for example title, page number, footnote, text (Cristani et al., 2018: 4; Grana et al., 2016). This part can include determining a logical order of the content (Grana et al., 2016: 4).

Logical labelling is a complex problem (Qin et al., 2018: 36). Documents can be complex with different formats and structures, for example, business documents, user manuals and scholarly articles typically have different structures (Rahman & Finin, 2017). Not much reviewed work on logical layout analysis is available, and most of the research is directed to the analysis of business letters, invoices and periodicals (Qin et al., 2018: 36). Gander et al. (2011) mention that logical layout analysis is not as

popular as research in OCR or physical layout analysis. In addition, most of the research datasets are private which can make development in the field more challenging (Qin et al., 2018: 36).

Despite challenges, there has been some research to address the problem of logical labelling. Some of these studies will be discussed here.

Qin et al. (2018) developed a system called LABA, which is a supervised machine learning system used for the logical layout analysis of scanned pages of Arabic books. They used classifiers to identify paragraphs, titles and headers, page numbers, pictures, captions and noise. The pixel was used as basic unit that must be labelled correctly in the evaluation of the system, and LABA achieved high pixel class membership accuracy values of 96.5% (Qin et al., 2018: 38). The code for LABA is publicly available.

Rahman and Finin (2017) propose a framework in which the logical sections in a document are identified and labelled with semantically meaningful names. In terms of logical labelling, they identify sections as regular text or headings. Headings are further classified as top level section headings, subsection headings or sub-subsection headings. A topic modelling algorithm was used to obtain a semantic concept for each section.

Yang et al. (2017) proposed a model that uses both visual and textual information to identify the following components in a document: figure, table, section, caption, list and paragraph. The model was tested on three datasets and improved on previously established benchmarks (Yang et al., 2017: 5322).

Most research on logical layout analysis focuses on fixed format documents. However, there are some studies on re-flowable documents, for example, Hao et al. (2018) worked on the identification of document components in re-flowable documents. Format and content features were used to identify the following document components: abstract, body, keyword, heading, picture, caption, table, and list. Format features include aspects such as font size, alignment, spacing and indentation. Content features are specific words, such as appendix.

Research to automate the encoding of digitised dictionaries was done by Khemakhem et al. (2017). Though dictionaries are highly structured information sources, unlike novels that are not as structured, their research is pertinent to this study as the encoding standard chosen for this project was TEI. They achieve some success, but

also suggest that more focus should be given to the feature selection process and more annotated data should be used for training data.

Gander et al. (2011) recognised that books as items of analysis are rarely covered by the research community and developed a system that attempts to understand historical books. Their system detects the following components: page number, page header, signature-mark, text, footnote and heading. The system works with scanned images that have been OCR processed.

The current research on document analysis relies on formatting or layout to some extent to determine the function of text (either in fixed document layouts or flowable layouts). Examples include the position on the page (e.g. Qin et al., 2018) and typesetting information (including font size, shape, line spacing) (e.g. Gander et al., 2011; Hao et al., 2018; Rahman & Finin, 2017).

It is currently difficult to compare logical labelling systems, because there are few standardised benchmarks or evaluation sets available (Qin et al., 2018: 36). However, one dataset that has been made available is from the Competition on Recognition of Documents with Complex Layouts, which is part of the International Conference on Document Analysis and Recognition (ICDAR2015) (Qin et al., 2018: 36).

The next aspect of interest in this section, is the ability to identify direct speech automatically. Various studies have been done to identify and classify direct speech automatically. Not all studies elaborate about the way in which direct speech is identified. A simple assumption is to assume that direct speech is speech between quotation marks. However, different punctuation marks can be used to mark direct speech. Pouliquen et al. (2007) used the following punctuation marks to identify direct speech in their study, “[”] (two single apostrophes), [“”] (two curly apostrophes), [.,.] (two commas, used in some Dutch newspapers, [« /.../ »] (French quotes), [“ /.../ ”] (the English curly quotes), [<< /.../ >>] (two brackets), ["/.../"] (double single-quotes), ["/.../'] (single quotes)". Furthermore, these punctuation marks can be used for other purposes; for example, quotes can be used to indicate a title (Koch et al., 2014). A selection of studies that aimed to identify or classify direct speech will be discussed here.

Pouliquen et al. (2007) developed a tool that automatically extracts quotations. It also identifies the speaker of a quotation, as well as the people referred to in the quotation. It was developed for news media and is multilingual. They discuss their algorithm for

detecting quotes and all the components required for their algorithm to identify a section of text as a quote.

A study by Elson and McKeown (2010) focuses on direct speech in literature and aims to attribute instances of quoted speech to their speakers. They do not elaborate on how the direct speech is identified, except for saying that “quoted speech is a block of text within a paragraph falling between quotation marks” (Elson & McKeown, 2010: 1014).

A study was done by Yang et al. (2017) to identify both the speaker and listener of direct speech. They state that they employed “a simple rule-based method to extract all direct quotes” (Yang et al., 2017: 326).

Another study that is of interest here was done by Koch et al. (2014). They developed a system to do in-depth analysis of large text documents and offers a visual representation. This system includes some automatic annotation of texts, specifically literal quotations and proper names. A regular expression was initially used to identify quotes. However, an active learning algorithm was used to create automatic annotations that were more complex, for example, where quotation marks are used for other purposes than an indication of direct speech.

Another area where research is done to automate the identification of features in a text is the extraction of named entities. Named entity recognition (NER) is the process of “locating and categorizing important nouns and proper nouns in a text” (Mohit, 2014: 221). The term *named entity* was first used at the Sixth Message Understanding Conference (MUC-6) (Goyal et al., 2018: 22). While working on information extraction, it was recognised that it is important to identify units such as, persons, locations, organisations, dates and time (Nadeau & Sekine, 2007). The most common classes of person, organisation and location are found in most general NER systems (Mohit, 2014: 222). However, it should be noted that the entities to be identified depends on the domain of interest, for example, in the biomedicine domain, entities of interest are gene and gene products (Goyal et al., 2018: 22).

Named entity recognition and classification (NERC) became a subtask of information extraction and plays an important role in various natural language processing tasks, for example, automatic text summarization, machine translation, information retrieval, text clustering, information extraction, knowledgebase population, opinion mining semantic search and question answering (Goyal et al., 2018: 24-25).

Named entity recognition has been studied for many languages and various systems have been developed to do this task (Mohit, 2014: 221). Techniques used to develop

named entity recognition systems are classified as rule-based approaches and learning or statistical approaches (Goyal et al., 2018: 26-36; Mohit, 2014: 224). Statistical approaches include supervised, semi-supervised and unsupervised systems. Different tagsets have been developed, each tagset including different entities (Goyal et al., 2018: 26).

This area of research can be considered fairly successful. Already at the MUC-7 in 1998, the best systems achieved near human performance with an F score of 93.39%, while human annotators scored 97.60% (MUC-7, 2007).

Some examples of Natural Language Processing libraries that can perform Named Entity Recognition are Apache OpenNLP, Stanford CoreNLP, spaCy and GATE (<http://services.gate.ac.uk/annie/>).

One example where the Stanford Named Entity Tagger was used will be mentioned here. In the study by Koch et al. (2014), where a tool was developed to visualise a large text collection, the Stanford tagger was used to detect the names of people and places.

The Stanford Named Entity Tagger is available through a web interface (<http://nlp.stanford.edu:8080/ner/process>). This interface will be used to give an illustration of named entity recognition. In this example (Figure 11), the English classifier using MUC-7 was selected. The potential tags are location, organisation, date, money, person, percent and time. The tagger correctly identified two persons, a date and a location.

Stanford Named Entity Tagger

Classifier:

Output Format:

Preserve Spacing:

Please enter your text here:

I was first introduced to the novels by Dorothy L. Sayers by my brother in 2017. I have since enjoyed the adventures of the detective Lord Peter Wimsey. Though most of his adventures take place in London, there are some set in the countryside.

I was first introduced to the novels by **Dorothy L. Sayers** by my brother in **2017**. I have since enjoyed the adventures of the detective Lord **Peter Wimsey**. Though most of his adventures take place in **London**, there are some set in the countryside.

Potential tags:

- LOCATION**
- ORGANIZATION**
- DATE**
- MONEY**
- PERSON**
- PERCENT**
- TIME**

Figure 11 Demonstration of Stanford Named Entity Tagger

Automatically detecting the language of a given text is largely considered a solved problem (Goutte et al., 2014: 139). “Automated language identification is the problem of determining the language that a passage of text is written in” (McNamee, 2005: 94). Some work has reported almost perfect language identification, given certain conditions (McNamee, 2005). The performance of automatic language identification can be experienced first-hand by general users through the publicly available translation tool from Google. Figure 12 shows an example of a short sentence that was correctly identified as Dutch.



Figure 12 Google Translate

The two most common techniques used to automatically identify the language of a particular text are statistical language modelling and analysing the frequency of common words in a text (McNamee, 2005).

However, there are some cases where automatic language identification could be improved, namely, where very little input data are available, where multiple languages are present in one text and where similar languages are used (Goutte et al., 2014). Most automatic language identification systems assume a single language and not much work has been done on identifying multiple languages (Lui et al., 2014: 28). Handling multilingual documents has been termed an open research question and sentence-level and word-level language identification have been suggested as fields for further research (Jauhiainen et al., 2019: 739). Some research in this field has been done. For example, Lui et al. (2014) present a method that determines whether a document contains multiple languages, which languages are present in the multilingual document, and provides an estimation of the portion of the document that consists of which language. As future work, they suggest segmenting a document according to language. Some other proposed methods are mentioned by Jauhiainen et al. (2019: 740).

2.5.5. Bibliographic level

Theoretical basis

The most common metadata assigned to items are bibliographic metadata. Information to describe information resources is familiar and widely used, for example, catalogue records in libraries (Haynes, 2018; Hodge, 2001). Consider the following bibliographic metadata, describing a book, extracted from the library of the University of Pretoria:

<p>Title: Aquinas. Author: Ralph McInerny. Subjects: Thomas, Aquinas, Saint, 1225?-1274; Christian philosophers Italy 13th century. Publication details: Cambridge, UK : Polity Press Language: English</p>
--

The importance of this kind of metadata has already been referred to in section 1.3, and a variety of uses and purposes have been mentioned, but in the context of this study it is worth emphasising that it is critical for retrieval or for the discovery of resources (Pomerantz, 2015).

There are many well-known standards, such as MARC, MODS, ONIX, BIBFRAME and Dublin Core that can be used to capture this kind of bibliographic metadata. Much work has been done on establishing and implementing bibliographic metadata standards.

Gilliland (2016) calls it a “bewildering array of metadata standards and approaches from which to choose”. Each metadata schema will have advantages and disadvantages. For example, MODS (Metadata Object Description Schema) is less complicated than the traditional MARC records, but can create richer descriptions than the simpler Dublin Core metadata schema (Library of Congress, 2016).

Though the importance of metadata for search and retrieval cannot be overstated, the problem in some existing systems regarding bibliographic metadata does not have to do with a lack of standards, rather it has to do with the availability or quality of metadata. For example, the metadata for the Google Books corpus cannot be made available due to copyright restrictions (Culturomics, 2017).

In HathiTrust there is extensive bibliographic metadata available. When submitting to HathiTrust, the bibliographic records should be in MARCXML format and be valid MARC21 (HathiTrust Digital Library – Bibliographic metadata specifications, n.d.). These bibliographic metadata make discovery and filtering possible.

Another point of interest is the level at which bibliographic metadata are assigned. At the moment, bibliographic metadata are applied to the book/volume level and it seems that researchers are interested in having information about sections in an entity. As was explained in section 2.4, researchers are not always interested in a book as such, but about sections in book. Books could contain different sections that could be of interest, for example, consider a book with a preface by a different author from the rest of the text, a collection of poems by different poets or sections of quoted texts in a book. Two specific examples will be given to illustrate this concept. The first is the researcher’s copy of the book *Jane Eyre* written by Charlotte Bronte and edited by Q.D. Leavis. The text of the novel starts on page 39 and ends on page 477. Before that, there is a preface and note by the author. Before that, there is an introduction by the editor, and the last pages in the book contain notes by the editor (pages 479-489). This means that the author of the volume (Charlotte Bronte) is different from the author of the introduction and the notes (Q.D. Leavis). Another example is the use of epigraphs at the start of chapters in *Middlemarch* by George Eliot; for example, the quote at the beginning of chapter 42 is from *Henry VIII* by William Shakespeare. The author of the quote is not the same as the author of the novel. This information could be of interest to researchers and therefore in-text bibliographic information about such sections in texts will make it possible to retrieve words from these sections specifically.

Automated encoding of bibliographic metadata

Large scale digitisation projects produce a large amount of material and manual assignment of metadata is expensive (Dobрева et al., 2013: 6). The use of automated methods to help with the creation of metadata has been suggested (Dobрева et al., 2013: 6). Greenberg (2004: 62) defines automatic metadata extraction as using an algorithm to extract metadata from the content of a resource. The benefits of automatically generating or extracting metadata have been well documented (Lu et al., 2008: 167).

It is beyond the scope of this study to provide an in-depth analysis of some of the research that has been done in this field. Good summaries are provided by Dobрева et al. (2013) and Park and Brenza (2015). According to Dobрева et al. (2013: 11), the element for which most extraction methods have been developed is the title element. Most of the examples discussed by Dobрева et al. (2013) extract metadata from research papers. These are publications with predictable structures and are possibly more suitable for automation than other creative works. Some of the research in this field links to document analysis discussed in the previous section, as formatting and positioning can be part of the rules to determine elements on a page. Though most work focuses on scholarly publications, an interesting study was conducted by Gao et al. (2011) to automatically extract metadata from the title pages of Chinese books.

Apart from metadata that should be extracted directly from the content, such as title, author and publisher, there is also research on creating metadata based on the analysis of the content. Park and Brenza (2015: 29) discuss the concept of content extraction, where “computing techniques are used to extract information from the information resource itself”. They point out that this technique can be used to identify key terms. Further research is being done to map the extracted terms to controlled vocabulary (Park & Brenza, 2015: 32).

Unfortunately, it has been noted that much of the research in this field has been fragmented (Dobрева et al., 2013: 9). Many tools are developed to address only a specific need in a specific domain, consequently the applicability of the tools is diminished (Park & Brenza, 2015: 40). It is also apparent that much of the research in this area focuses on extracting metadata from scientific publications (e.g. Lu et al., 2008; Marinai, 2009). The tools also often typically only extract one or a few metadata elements and there is no comprehensive solution (Park & Brenza, 2015: 40).

Genre is an example of a metadata field. Underwood (2015b) has done research on detecting the genre algorithmically in large digital libraries, specifically on a page level. Different models were trained to make predictions about genre for each page in a volume of text. An example to illustrate their work, is a volume of *Wordsworth's Poetical Works* that was analysed and the model correctly identified that the first part of the volume consists of poetry, whereas the last part of the volume consists of a tragedy that should rather be classified as drama.

This section considered the types of metadata that can be applied to texts to enhance retrieval. The next section will review some tools and projects that have been developed to search in, analyse and manipulate large text collections. As part of the review, it will be important to consider the extent to which words or phrases can be retrieved using metadata to filter the results.

2.6. Tools and projects

Numerous software applications have been developed to search in, explore and analyse collections of texts. In the next section a few prominent examples will be discussed. It is beyond the scope of this study to discuss all tools that are used to explore and analyse large corpora. Furthermore, this research will not cover the use of programming (e.g. R or Python) for computational analysis of large text collections. Though it is an increasingly important field of research, this research will focus on tools that can be used by people who are not familiar with programming to search for words or texts with specific properties in large text collections.

A tool that is used to query a text collection is dependent on the text (corpora) used by that tool. As such, when the tools are discussed, their associated corpora will also be discussed.

The first two tools that will be discussed are built on some of the world's largest digital libraries, namely, the Google Books Ngram Viewer and the HathiTrust+Bookworm, which get their data from Google Books and HathiTrust Digital Library respectively.

Next, two tools that can be considered to be from the digital humanities will be discussed. The Perseus Project is one of the earlier efforts that took advantage of the opportunities that technology brings to text analysis. The other project that will be discussed is Voyant Tools.

Another tool that will be considered is TXM, which was developed to process documents that have been encoded with specific structural properties in TEI.

As researchers in the digital humanities are interested in using methods from the field of corpus linguistics, it will be useful to explore some tools from the field of corpus linguistics. Two tools from this area will be discussed, namely, BNCweb, and BYU Corpora at corpus.byu.edu.

Lastly, some other tools and research projects that enable work on large text collections will be considered briefly. Also, some corpora that were not included in the discussion of these tools will be mentioned.

2.6.1. Google Books Ngram Viewer

The Google Books Ngram Viewer is an interactive online tool that allows users to display the relative frequencies of words (or phrases) from the corpus over a certain time period (<https://books.google.com/ngrams>). This application has made it possible for researchers to study cultural and linguistic changes on a large scale (Lin et al., 2012).

Figure 13 shows the frequency of the words *umbrella*, *parasol* and *broolly* in the English corpus from 1800 to 2019.

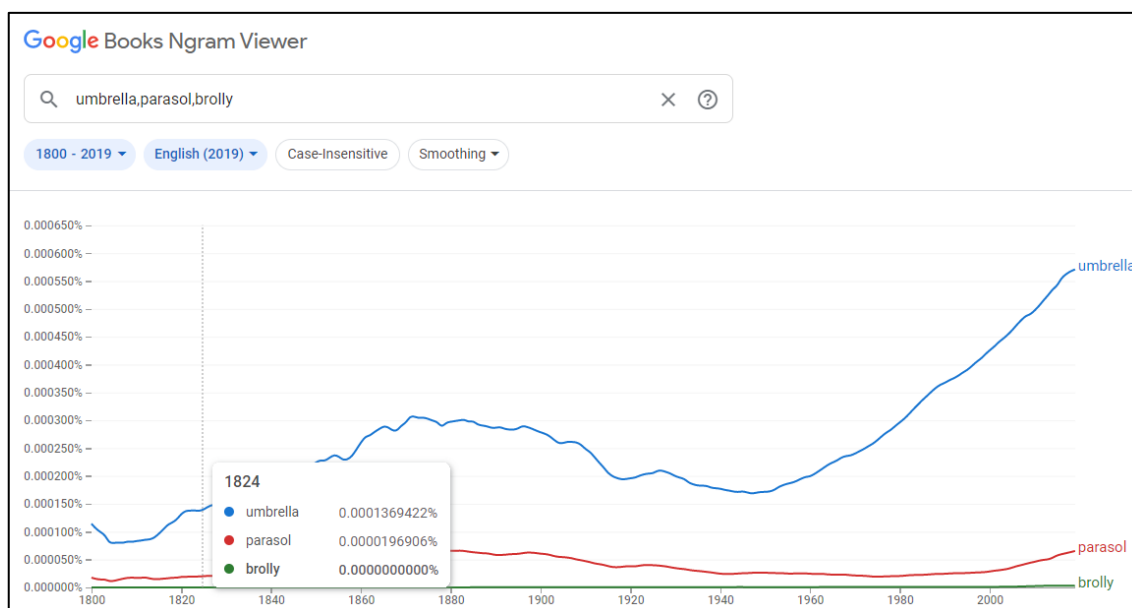


Figure 13 A search in Google Books Ngram Viewer

The corpus used in the Ngram Viewer is created from the documents in Google Books (Michel et al., 2011: 2). Not all books in Google Books have been used to create the corpus for the Google Books Ngram Viewer. For example, some books where the metadata for the date of publication are possibly incorrect were excluded (Culturomics, 2017).

The first version of the corpus contained over 500 billion words in seven different languages with the oldest books from the 1500s (Michel et al., 2011: 2). The different subcorpora that could be searched in the first version are the following: American English, British English, English, English Fiction, English One Million, Chinese, French, German, Hebrew, Spanish and Russian. The second version of the corpus is based on a larger selection of books than the first and the Italian language has been added (Lin et al., 2012: 170). The third corpus for the Ngram Viewer, the 2019 corpus, was released in February 2020 (Google Books Ngram Viewer Info, 2020). Data are obtained from over 8 million books in total, resulting in over 800 billion words. There is only one version of the English One Million corpus. The newer versions use improved OCR, improved library and publisher metadata, and improved tokenisation (Google Books Ngram Viewer Info, 2020). N-grams are extracted from the corpus, where the maximum is a 5-gram and the n-grams must occur at least 40 times in the corpus (Michel et al., 2011: 2). An n-gram is adjacent sequences of n items in a text (Friginal et al., 2014: 51). This means that *season* is a 1-gram, *season of* is a 2-gram, *season of mists* is a 3-gram, etc. In the 2012 version n-grams are extracted that span page boundaries, but n-grams that cross sentence boundaries are not formed (Google Books Ngram Viewer Info, 2020; Lin et al., 2012: 171).

The frequency of n-grams is calculated by dividing the number of occurrences of n-grams in a given year by the total number of words in the corpus for that year (Michel et al., 2011: 2). This is necessary as the number of books published and available in later years is much higher than earlier years (Ophir, 2016).

In order to comply with copyright laws, the full-text of the material in the corpus is not available nor is the data (metadata) accompanying the corpus (Culturomics, 2017; Koplenig, 2017: 182).

The Google Books Ngram Viewer has been used successfully in various studies. See the discussion in section 2.3.

2.6.2. HathiTrust Bookworm

Apart from building and maintaining a large-scale digital repository, HathiTrust also aims to improve access to the material in the repository (Christenson, 2011). The HathiTrust Research Center (HTRC) provides support to users who wish to explore the massive amounts of data in the Digital Library (Jett et al., 2016b: 1). They specifically provide tools that enable computational analysis of items in the library in a secure environment (HathiTrust Research Center, n.d.). The secure environment allows a researcher to do computational analysis on a selection of texts without reading or

downloading the texts, ultimately allowing a researcher to work with texts that are still under copyright (Howard, 2017). The Research Center focuses on quantitative research (investigating a collection) and seeing how aggregations of data can answer macroscopic questions (HathiTrust Research Center, n.d.). The HathiTrust Research Center (HTRC) is a joint research initiative and is based at Indiana University and University of Illinois at Urbana-Champaign (Jett et al., 2016b: 1).

Over 60% of the items in the HathiTrust Digital Library are subjected to copyright restrictions. In order to enable scholars to do research on these items, the Research Center provides a non-consumptive research environment in which scholars collect items into collections and then use computational methods to derive insights from them, without getting actual access to the items that are restricted (Jett et al., 2016b: 1). As a result, the metadata describing the items are critically important (Jett et al., 2016b: 1-2) . To accommodate the need to access fine-grained units, the HathiTrust Research Center is working to "create a layer of metadata objects that describe finer-grained resources so that scholars can identify them and make use of them in their analysis" (Jett et al., 2016b: 36).

One way in which the HathiTrust Research Center enables researchers to do research while respecting copyright laws, is through the Extracted Features Dataset that they have made available (Capitanu et al., 2016). This dataset is constructed from volumes in the HathiTrust Digital Library and contains useful information (features) about the volumes. Various features are included, such as part-of-speech tagged token counts, header and footer identification and line-level information.

One of the experimental tools that is used to analyse the data in the HathiTrust Digital Library is the HathiTrust+Bookworm, a tool used for the visualisation of the frequency of the usage of words over time. It was developed by the HathiTrust Research Center in collaboration with the Cultural Observatory team, the team also responsible for developing the Google Books Ngram Viewer in collaboration with Google (The iSchool at Illinois, 2014). This project enables a user to filter specific data for analysis by using the available metadata (The iSchool at Illinois, 2014). The HathiTrust+Bookworm, for example, allows a user to filter by genre, class, resource type, contributing library, amongst others. Their project aims to improve the discovery of items in the library and the analysis of these items (The iSchool at Illinois, 2014).

The example in Figure 14 of the HathiTrust+Bookworm shows the use of the word *carriage* as published in the UK and the USA over the time period 1760 to 2010.

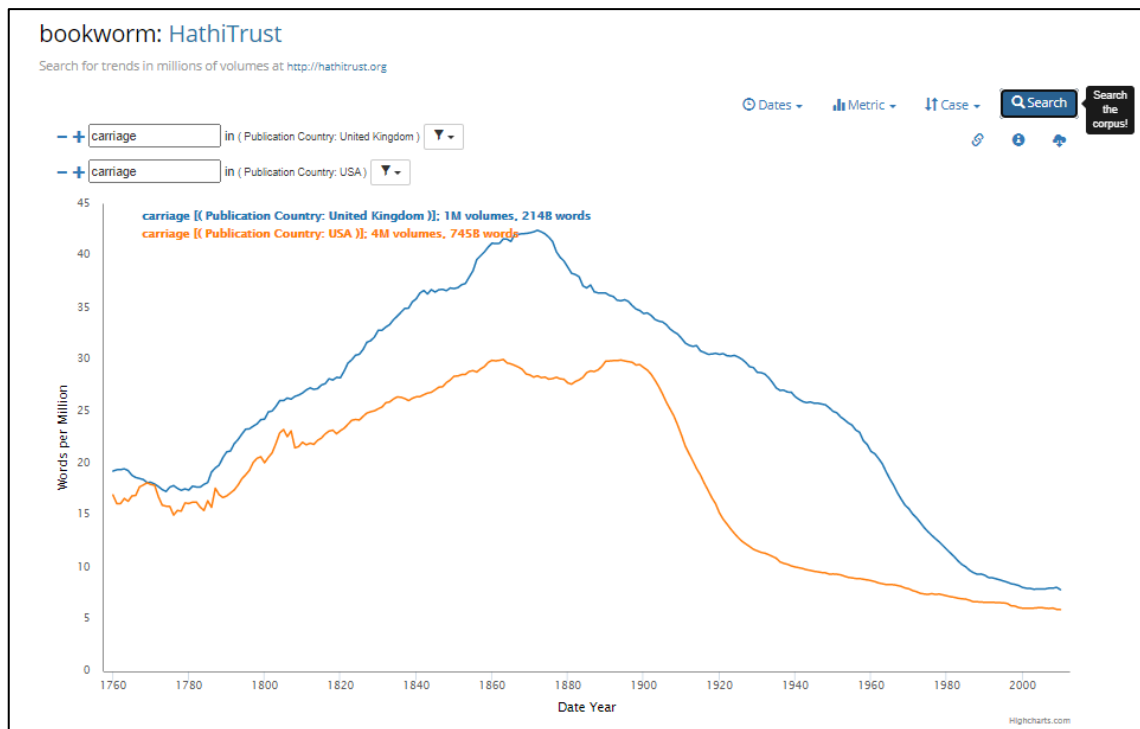


Figure 14 An example from the HathiTrust+Bookworm

2.6.3. Perseus Project

The Perseus Digital Library (<http://www.perseus.tufts.edu/hopper/>) is a project that explores the possibilities that online digital collections offer. The library started with material from the Greco-Roman world, but their collection has since expanded (Perseus Digital Library, n.d.-b). This digital library does not only contain texts, but also artefacts and images. According to Crane (1998) the “long-term goal must be to make accessible, both physically and intellectually, to every human being on this planet the complete record of humanity”.

In the Greek and Roman collection there are currently 44,462,693 English words, 13,507,448 Greek words and 10,525,338 Latin words (Perseus Digital Library, n.d.-a). There are also other collections in the Perseus Project. Although a smaller library than, for example, Google Books, it still holds a large amount of texts that are all encoded to make analysis easier and has made a significant contribution to research in the humanities. The main aim of the Perseus Project is to give access to individual items (e.g. texts) so that they can be studied in depth. Tools are provided to aid in the understanding and interpreting of texts. For example, a user can click on a specific word in a text to see the parses for that word, dictionary entries and frequency statistics. In order to get to a specific text, a user can browse through the collections or search for a specific text. A user can then select a text to view, for example *Against Cataline* by Cicero (see Figure 15).

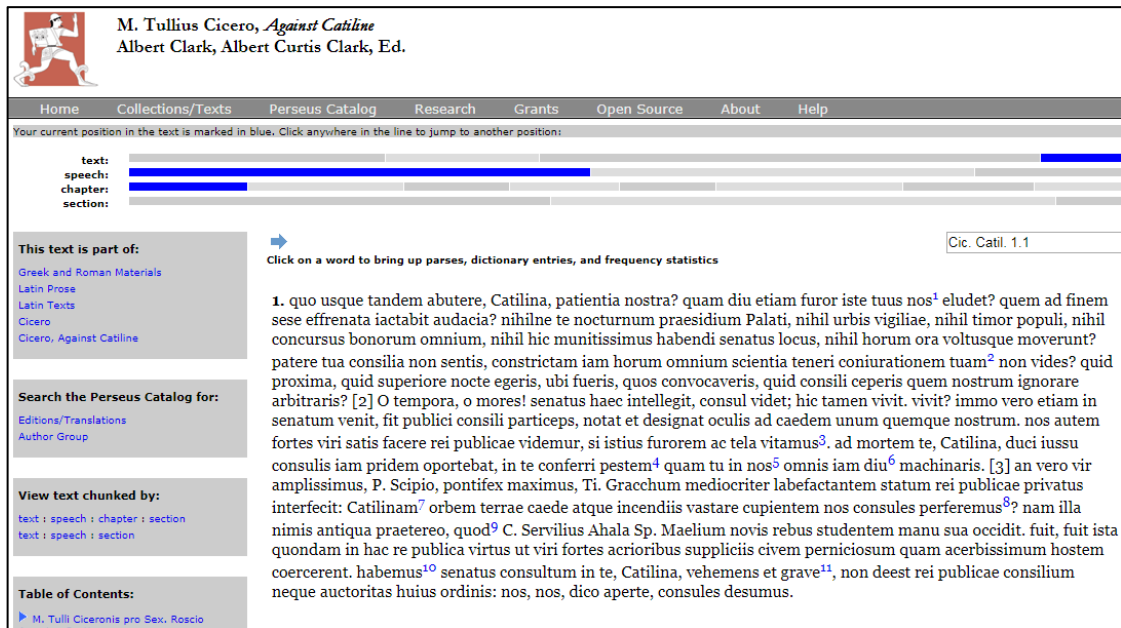


Figure 15 *Against Catiline* by Cicero

2.6.4. Voyant Tools

Voyant Tools is a free, web-based tool for text analysis developed by Stéfan Sinclair, Geoffrey Rockwell, and their project team (Gallant et al., 2014; Welsh, 2014: 96). It can analyse texts in a variety of formats, including plain text, HTML, XML, PDF, RTF, and MS Word (Sinclair & Rockwell, 2016).

The main page of Voyant Tools (Figure 16) consists of various panels, each containing a different tool useful for text analysis (e.g. to view trends). These panels can be customised to display the tool with which the user wishes to work.

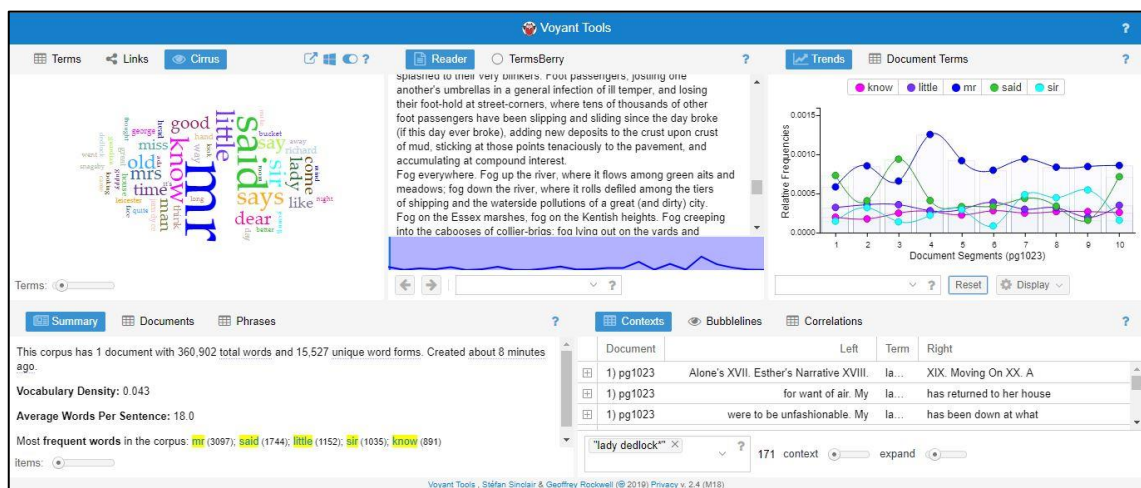


Figure 16 Voyant Tools

One of the great advantages of Voyant Tools is its user friendly interface (Gallant et al., 2014; Welsh, 2014: 96). It has been noted that most text analysis tools are not

designed for general humanities scholars, that tools often require specialist knowledge, either from the field of linguistics or digital humanities (Welsh, 2014: 96). This issue is addressed by Voyant Tools, through its simple user interface and extensive documentation.

Voyant Tools is corpus independent and a user can add his/her own text to analyse. A user can add texts in different ways. A user can paste text into the text box on the start screen, enter the URL for text to be analysed, upload text from the computer or open existing corpora. A user can work on one or several documents.

2.6.5. TXM

TXM differs from the Google Books Ngram Viewer and the HathiTrust+Bookworm, in that the primary function is not to visualise the frequency of the usage of words over time, but it will be discussed as an example of a tool that can extract data from a corpus on a detailed level.

TXM is a “free, open-source Unicode, XML & TEI compatible text/corpus analysis environment and graphical client based on CQP and R” (TEI wiki – TXM, 2016). This software platform is the result of a project by textometry teams from mainly French universities (Heiden, 2010). One of the goals of the project was to be compatible with TEI encoded data (Heiden, 2010). As a result, the TXM software allows documents encoded with TEI to be queried and explored. Version 0.8.1 of the TXM desktop software was released on 29 June 2020.

TXM is a powerful corpus analysis framework that incorporates various analysis tools. The TEI wiki page for TXM list its most prominent features (TEI wiki – TXM, 2016). For example, it is able to produce KWIC concordances, frequency lists and progression graphics. Furthermore, it uses R packages to do factorial correspondence analysis, cluster analysis, specific word patterns analysis and collocations analysis.

The project differs from text mining, in that it allows the user to get back to the original context (Heiden, 2010). It is possible to import corpora into the TXM framework, for example, the Leviathan by Thomas Hobbes, 1588 – 1679, is an XML-TEI P5 text sample from the EEBO-TCP Phase 1 project (SourceForge.net, n.d.) that can be imported.

This research will use the two example corpora, GRAAL and VOEUX, to illustrate the functions of this tool.

The GRAAL corpus (Figure 17) is “based on an edition of the K manuscript (Lyon, Municipal Library, 77 Palais des arts) from the book *La Queste del saint Graal*” (TXM User Manual, 2018).

The VOEUX corpus contains 54 transcripts of presidential addresses by seven French presidents covering the period from 1959 to 2012 (TXM User Manual, 2018).

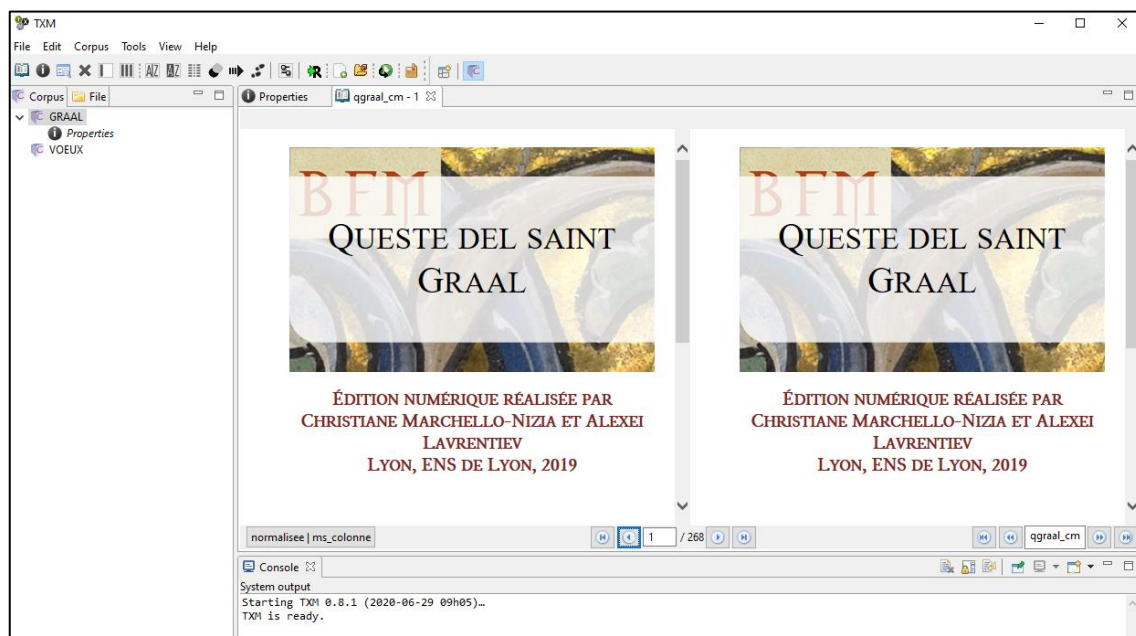


Figure 17 The GRAAL corpus in TXM

2.6.6. BNCweb (CQP-edition)

The British National Corpus (BNC) is an English corpus of 100 million words, containing written and spoken English from a variety of sources (Burnard, 2009; Grant, 2005: 437-438; Kennedy, 2003: 471). Several universities, publishers and the British government collaborated in the early 1990s to develop this corpus (Kennedy, 2003: 471). In 2007 an XML edition of the corpus was released. This version is encoded with TEI to include both part-of-speech information and a variety of other structural properties of texts (e.g. quotes, paragraphs, lists) and bibliographic information of each text is included in a TEI-conformant header (University of Oxford IT Services, 2015a). The encoding in the corpus enables automatic retrieval and analysis (Burnard, 2009: 471; Kennedy, 2003).

There are various options to use and explore the British National Corpus (BNC). One of the tools that allows one to explore the British National Corpus online is the BNCweb (CQP-edition) hosted by Lancaster University. The CQP-edition of the BNCweb was developed by Sebastian Hoffmann and Stefan Evert. The original BNCweb was considered user-friendly and intuitive to use; however, it was restricted in terms of the

complexity of queries that could be executed (Hoffmann & Evert, 2006). As such, the CQP-edition of the BNCweb was designed to incorporate the powerful Corpus Query Processor (CQP) (Hoffmann & Evert, 2006). According to Hoffmann and Evert (2006: 180), the greatest strengths of CQP are the integration of metadata and queries and the ability to perform general searches on very large corpora. The BNCweb (CQP-edition) makes use of the XML edition of the British National Corpus (BNC) (BNCweb (CQP-edition), 2008).

Apart from simple queries, more complex searches can be done using the CQP-syntax (BNCweb (CQP-edition), 2008). Furthermore, various post-query features, such as, the display of sorted results, thinning of results, frequency breakdown and distribution, are available. The results for the simple query *mountain* are shown in Figure 18.

Your query "mountain" returned 3816 hits in 968 different texts (98,313,429 words [4,048 texts]; frequency: 38.81 instances per million words) (0.134 seconds)		
<input type="button" value="<"/> <input type="button" value="<<"/> <input type="button" value=">>"/> <input type="button" value=">"/> Show Page: <input type="text" value="1"/> Show KWIC View Show in random order Show extended audio data controls New Query <input type="button" value="Go!"/>		
No	Filename	Hits 1 to 50 Page 1 / 77
1	A04 1527	The third perspective is Kao yuan , in which the viewer is looking up towards a mountain scene, as William Willetts puts it, 'through successively receding heights represented by flat parallel planes, each with its own horizon'.
2	A05 1092	Where can Jenny have been, in the course of her adolescence, to be willing, if only out of nervousness, to accept that the Reds in Spain have been swept out from under the bed and up into mountain caves?
3	A06 957	Your cheeks like damask, the soft white loveliness of your breasts, leading to the firm dark mountain peaks of your, Laura, now I'm dreading which part of my body he will choose next on which to turn the great white beam of his fucking sincerity.
4	A08 990	Genius is the bust of Beethoven and Keats dying and Shelley dying and the size of War and Peace and poor old Sartre banging away at his trilogy and Hemingway paring it down to its essence and Monet unable to distinguish colours any more and Picasso staring out at the camera with his chest bare and his eyes blazing and Cézanne snarling like a dog and then walking out of Aix with his canvas and paints on his back to paint that mountain and Byron dying and Pushkin dying and all the rest of it.
5	A08 1661	Now planning huge work to take place simultaneously in every town in Greece and on every mountain .
6	A0C 852	The wines include Le Bonheur Blanc Fume (Sauvignon Blanc) from Stellenbosch which is unwooded with a fresh, grassy character; Fleur du Cap Chenin Blanc Sec (crisp and fruity); Witzenberg Emerald Stein (semi-sweet Fleur du Cap); and Roodebloem from the Bergkelder or ' mountain cellars' of Stellenbosch.
7	A0F 117	You're making a mountain out of a molehill, Dorothy.
8	A0L 272	Go to Woodstock, the sea, the top of a mountain , a river, go forever from the flat respectability of home and market town.
9	A0N 1972	He had started out to make a rough count of the houses to be visited and then let his thoughts drift into a reverie of his own old home, the far tropical look of the mountain skyline beyond Loch Arkraig on the rare hot days.
10	A0P 408	Leonard recently referred to the memory of his father as 'a dark mass or mountain ,' of which, clearly, the details were too painful for the young boy to register or the adult to express.
11	A0P 409	(The image actually appeared in a somewhat different way in The Favourite Game : 'Concerning the bodies Breavman lost ... a man on the mountain ,' a reference to the cemetery on Mont Royale probably.)
12	A0P 1536	He needed solitude to write, as well as a place of his own for entertaining his girlfriends, which he found on Mountain Street.
13	A11 166	Peppercorn K1 2-6-0 No 2005 reflects the morning sun as it passes Bolden Colliery with the Northumbrian Mountain Pullman on 22 January 1983.

Figure 18 Results returned in the BNCweb (CQP-edition)

The BNCweb (CQP-edition) will further on be referred to as the BNCweb.

2.6.7. BYU Corpora

The corpus.byu.edu system (also referred to as BYU) gives access to various corpora, amongst others, the BNC, Corpus of Contemporary American English (COCA), Corpus of Historical American English (COHA) and an interface to the n-grams from Google Books. One of the latest additions to this system is the iWeb corpus (The Intelligent Web-based Corpus), which contains about 14 billion words in 22 388 141 web pages from 94 391 websites (Davies, n.d.). Many of the corpora on the corpus.byu.edu system include copyrighted material (Davies, n.d.). As a result, to abide by copyright law, only small sections are displayed in KWIC format.

The underlying architecture and interface of corpus.byu.edu was created and is maintained by Mark Davies at Brigham Young University (Davies, n.d.; Hardie, 2012). The corpus.byu.edu system follows a relational database approach and uses an SQL server to handle large corpora (Davies, n.d.; Hardie, 2012). It is one of the most widely used tools for corpus linguistics (Anthony, 2013: 153).

Two corpora, namely iWeb and BYU-BNC will be used in this study to investigate the functionality of corpus.byu.edu. The BYU interface to the n-grams from Google Books is slightly different than the BYU interface to the other corpora that it gives access to. The aim of this interface is to provide more search options to the Google Books data than the Google Books Ngram Viewer does. However, from the point of view of this study it does not offer significant other options and will only be discussed briefly in section 2.6.8.

2.6.8. Other projects

This section will consider various projects or examples that analyse large collections of text.

XAIRA

XAIRA is a retrieval tool for corpora. It will not be discussed in depth, but is worth mentioning here, due to the influence it had on other tools.

XAIRA (XML Aware Information Retrieval Architecture) is an open source tool that allows a user to search in an XML corpus (TEI wiki – XAIRA, 2007). It was developed primarily by Lou Barnard and Tony Dodd at the Oxford University Computing Services (University of Oxford IT Services, n.d.). It was designed specifically to work on the British National Corpus (BNC), but can work on other similar corpora; however, it takes advantage of the detailed encoding in the British National Corpus (University of Oxford IT Services, 2015b). The XAIRA software is provided with the BNC XML Edition, BNC Baby, and BNC Sampler corpora (University of Oxford IT Services, 2015b). According to the TEI wiki, the main features of XAIRA are that it works on any collection of XML documents, it indexes XML structures and words, and it can generate concordances, word indexes, collocations, summary statistics and text partitions (TEI wiki – XAIRA, 2007).

Various queries can be made through the XAIRA software. For example, depending on how the corpus has been encoded, one can search for the occurrences of words, patterns of words, specific part-of-speech categories, lemmas, phrases, as well as in specific structures of texts. Figure 19 shows an XML query in the XAIRA software,

where all the instances that have been encoded with the 'catRef' tag where the default attribute is 'LAW' will be found (University of Oxford IT Services, n.d.).

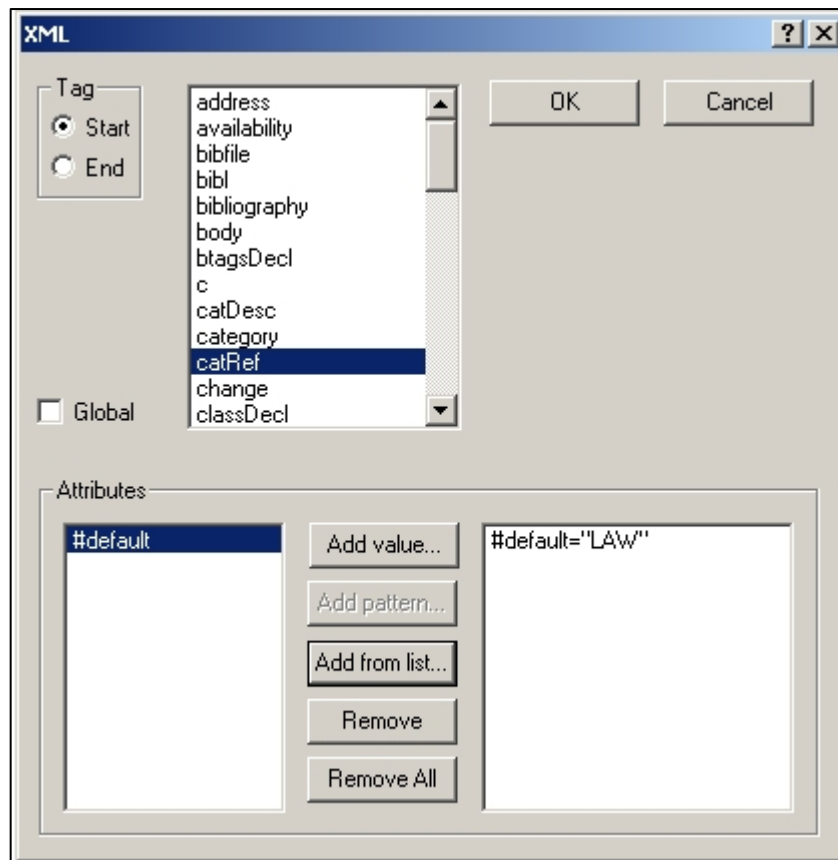


Figure 19 XML query in the XAIRA software (University of Oxford IT Services, n.d.)

In 2014 it was announced that XAIRA is no longer supported by the Oxford University and would henceforth be a community project (Wynne, 2014). It is currently still available on SourceForge (<https://sourceforge.net/p/xaira>).

Though a powerful search tool, specifically for searching in documents encoded with TEI, XAIRA has some limitations. There is a considerable learning curve to be able to use this tool effectively. In addition, the tool only lists the TEI elements as they are found in the corpus, and a user will have to have knowledge of TEI or the corpus to know how to search in the TEI elements. Furthermore, XAIRA does not seem to be widely used currently, but seems to have paved the way for more current systems.

CQPweb

CQPweb (Corpus Query Processor) is a web-based corpus analysis system (Hardie, 2012: 381). It has been designed to follow the design of BNCweb interface, so that a person familiar with the BNCweb interface can use the CQPweb without further training (Hardie, 2012: 388-389). Figure 20 shows how similar the CQPweb is to the BNCweb.

However, instead of being only limited to one corpus it is compatible with many corpora (Hardie, 2012: 381). From the main webpage, a user can select the corpus to work with (Figure 21). A notable example is the section of the Early English Books Online (EEBO) that is made available through CQPweb.

Menu		British English 2006: powered by CQPweb		
Corpus queries		Restricted Query		
Standard query		<div style="border: 1px solid black; height: 30px; width: 100%;"></div> <p>Query mode: <input type="text" value="Simple query (ignore case)"/> Simple query language syntax</p> <p>Number of hits per page: <input type="text" value="50"/></p> <p>Match strategy: <input type="text" value="Standard"/></p> <p><input type="button" value="Start query"/> <input type="button" value="Reset query"/></p>		
Restricted query				
Word lookup				
Frequency lists				
Keywords				
Analyse corpus				
Saved query data		Select the text-type restrictions for your query:		
Query history		Broad genre	Sample from	Text category
Saved queries		<input type="checkbox"/> Fiction <input type="checkbox"/> General prose (non-fiction) <input type="checkbox"/> Learned (academic) <input type="checkbox"/> Press	<input type="checkbox"/> Entire text <input type="checkbox"/> End of text <input type="checkbox"/> Middle of text <input type="checkbox"/> Beginning of text	<input type="checkbox"/> A. Press: Reportage <input type="checkbox"/> B. Press: Editorial <input type="checkbox"/> C. Press: Reviews (theatre, books, music, dance) <input type="checkbox"/> D. Religion <input type="checkbox"/> E. Skills and Hobbies <input type="checkbox"/> F. Popular Lore <input type="checkbox"/> G. Belles Lettres, Biography, Memoirs, etc. <input type="checkbox"/> H. Miscellaneous non-fiction <input type="checkbox"/> J. Learned (academic writing) <input type="checkbox"/> K. General Fiction <input type="checkbox"/> L. Mystery and Detective Fiction <input type="checkbox"/> M. Science Fiction <input type="checkbox"/> N. Adventure and Western Fiction <input type="checkbox"/> P. Romance and Love Story <input type="checkbox"/> R. Humor
Categorised queries				
Upload a query				
Create/edit subcorpora				
Corpus info				
View corpus metadata				
Corpus manual				
CLAWS7 Tagset				
USAS Tagset				
Oxford Simplified Tags				
Lemma/OST				
About CQPweb				
CQPweb main menu				
Help system				

Figure 20 The interface for CQPweb with the ability to restrict according to texts

UCREL		Welcome to CQPweb at Lancaster		CORPUS QUERY PROCESSOR		CWB	
<p>This server is maintained by Andrew Hardie</p> <p>Welcome back to the CQPweb server, Liezl Ball. You are logged in to the system.</p>							
Recently-used corpora				Quick links			
British National Corpus (XML edition)		Early English Books Online (V3)		Your corpus access privileges		Help! system	
The Arabian Nights (Richard Burton translation)		Works of Dickens		Your user account details		Log out of CQPweb	
Corpora available on this server (click here to view your own corpus access privileges)							
Present-day English							
The Arabian Nights (Aldine edition)		American English 2006		B-Brown (AmE 1930s)			
British English 2006		Spoken BNC2014		BNC Sampler			
British National Corpus (XML edition)		The Arabian Nights (Richard Burton translation)		Works of Dickens			
Brown Family (extended)		Kolhapur Corpus		Lanky Corpus			

Figure 21 Corpora available on CQPweb

Though CQPweb is a completely different system to the BNCweb from a technical point of view (Hardie, 2012: 387), it was purposefully modelled after the BNCweb, and

for the purposes of this study will not be discussed in depth. However, some pertinent aspects will be highlighted.

This tool sees a corpus as a sequence of tokens (e.g. words and punctuation) where each token can also have annotations associated with it, such as part-of-speech tags (Hardie, 2012: 390). The system stores information about annotations that are available for a specific corpus, as not all annotations are available in all corpora (Hardie, 2012: 391). The CQPweb also requires a corpus to be broken into texts that are indicated by the XML element <text> (Hardie, 2012: 390). As a corpus is divided into texts, metadata for each text can be stored and allow a user to filter a search on a text-level (Hardie, 2012: 394). Figure 22 shows the metadata for the Early English Books Online used in CQPweb. The ellipse in the figure indicates that some of the text-level metadata have been omitted to fit in the word-level annotations. An example of text-level metadata is title and an example of word-level is part-of-speech.

Menu		Early English Books Online (V3): powered by CQPweb	
Corpus queries		Metadata for Early English Books Online (V3)	
Standard query		Corpus title	Early English Books Online (V3)
Restricted query		CQPweb's short handles for this corpus	eebov3 / EEBOV3
Word lookup		Total number of corpus texts	44,422
Frequency lists		Total words in all corpus texts	1,202,214,511
Keywords		Word types in the corpus	4,713,326
Analyse corpus		Type:token ratio	0.0039 types per token
Saved query data		Text metadata and word-level annotation	
Query history			Additional Title(s)
Saved queries			Alternative Title(s)
Categorised queries			ID numbers (STC)

...

		Year Status
	The primary classification of texts is based on:	Century
	Words in this corpus are annotated with:	Simple POS (Oxford Simplified Tagset)
		Full USAS analysis (USAS tagset)
		Lemma
		Tagged lemma
		Unregularised spelling
		Part-of-speech (C6 tagset)
	The primary word-level annotation scheme is:	Semantic tag (USAS tagset)
		Part-of-speech

Figure 22 Corpus metadata for Early English Books Online (V3) in CQPweb

This means that a user can filter according to the text-level metadata when using the Early English Books Online corpus in CQPweb. A user can also use the word-level annotations (e.g. part-of-speech tags) in their search. CQPweb uses the Open Corpus Workbench (CWB) and a MySQL relational database system, making it a powerful search tool (Hardie, 2012: 381). The CQP-syntax, as well as a simpler custom query language can be used to query the corpora (Hardie, 2012: 396).

The extent to which a user can search according to the structural encoding of a corpus is not clear. Two example searches will be discussed. The first search was conducted on the BNC XML edition. The search “<quote>(*)* love (*)</quote>” (searching for love in direct speech, with multiple tokens before and one after) on the BNC through CQPweb returns the same results as the same search through the BNCweb. This means that the processor can take XML encoding into consideration. The next search was done on the Early English Books Online corpus. According to the researcher’s understanding (from other sources), the texts in the Early English Books Online corpus that are available through the CQPweb are texts selected and made available by the EEBO-TCP (Meyer & Eccles, 2016) and that these texts are also encoded with TEI (TCP, n.d.). The researcher therefore assumed that it would be possible to search for the word *love* that was highlighted in some way by using the TEI tag <hi> in the Early English Books Online (V3) corpus but received an error. It is only through the error message that the researcher was informed what tags could be used in the search (Figure 23) through CQPweb.

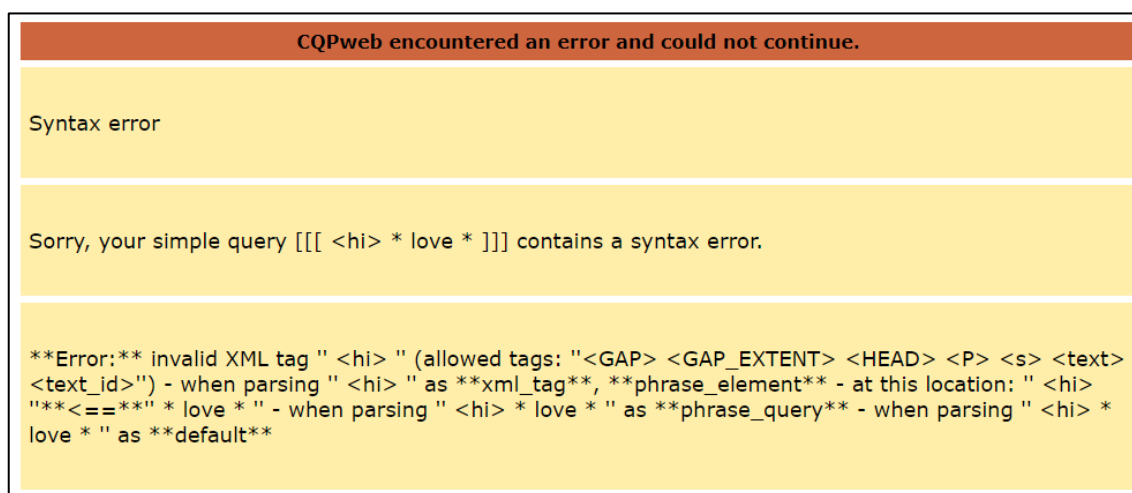


Figure 23 An error message in CQPweb

However, in discussing the tool, Hardie (2012: 401) notes that CQPweb is not TEI compatible. It is, therefore, probably not the main purpose of the tool, but it is primarily

designed (and used) to filter according the text-level metadata and search according to word-level annotations.

Hardie (2012: 389) notes that the usability of the CQPweb has made it suitable as a tool through which linguists, and other researchers from the humanities that are not familiar with corpus linguistics, can apply corpus techniques to datasets. However, this researcher suggests that, like the BNCweb, the tool still requires some learning before it can be used most effectively. The user is presented with a search box and, for more than simple word searches, will most likely have to learn the syntax. The tool is probably most suitable for, and can be used most powerfully by, someone with some knowledge of corpus linguistics.

Sketch Engine

Sketch Engine is a commercial tool used to study language, its use, patterns and trends (Sketch Engine, n.d.). It is used by linguists, lexicographers, translators, historians and others interested in studying language usage (Sketch Engine, n.d.). At the time of writing, Sketch Engine contained 500 off-the-shelf corpora in over 90 languages (Sketch Engine, n.d.). This tool will not be discussed in depth, as other corpus analysis tools have been covered. Although it does not offer significant other features than already discussed with other tools, it will contribute to this study to briefly look at some of the useful functions offered.

Through this powerful linguistic tool, the user may, for example, see typical combinations of words, synonyms, find examples in text, compare words and calculate n-grams (see Figure 24).

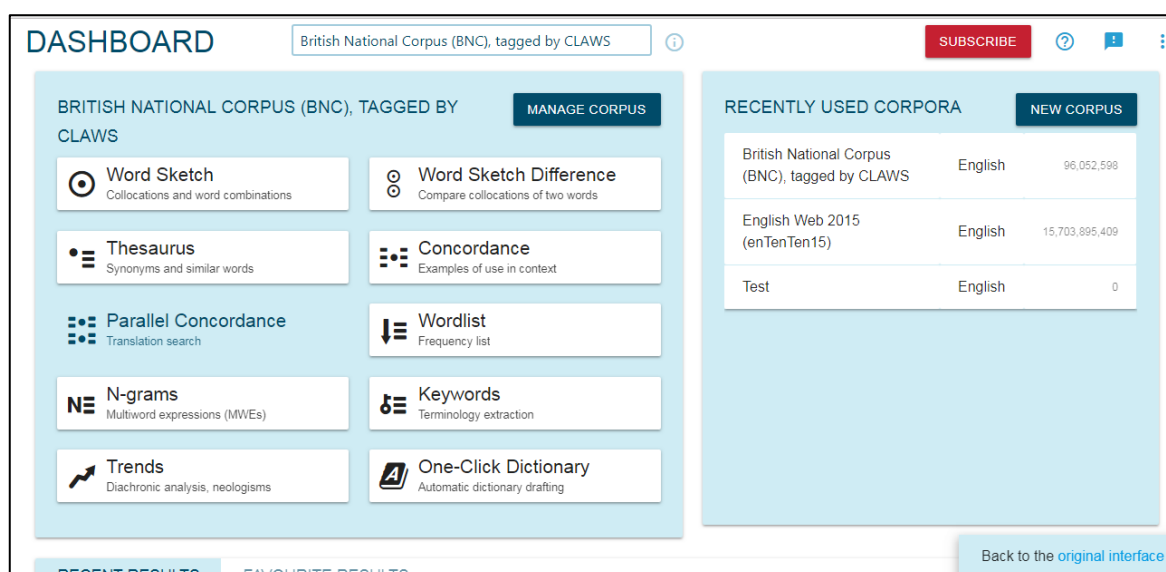


Figure 24 Sketch Engine

In the example in Figure 25 the word *love* was used to create a word sketch (the most typical combinations with a word).

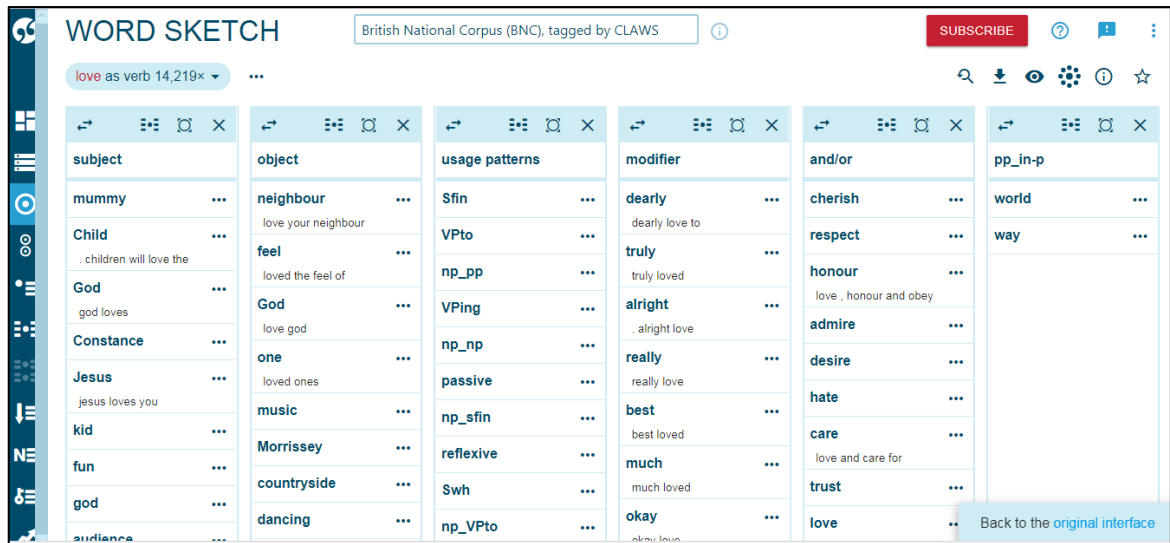


Figure 25 A word sketch of the word *love*

There is a visualisation option available for a word sketch, as seen in Figure 26.

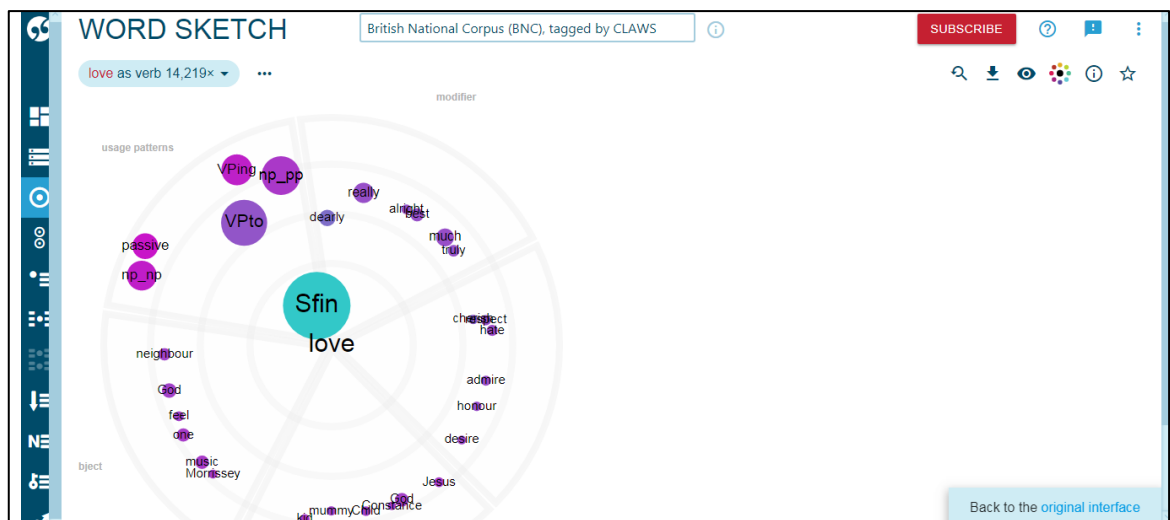


Figure 26 Visualisation of a word sketch

Trends of words can be viewed if the corpus that was selected supports this function (Figure 27). In this example the British National Corpus was selected.

The screenshot shows the 'TRENDS' section of the Sketch Engine interface for the British National Corpus (BNC), tagged by CLAWS. It displays a list of words and their frequencies, organized into three columns. Each word entry includes a rank, the word itself, a green checkmark icon, and the frequency count. A 'Back to the original interface' button is visible at the bottom right of the table.

word	Frequency	word	Frequency	word	Frequency
1 tasting	214	18 ambassadors	218	35 thirsty	256
2 treaty	2,957	19 angered	326	36 torso	212
3 minefield	113	20 campaigner	285	37 reputedly	177
4 spotlight	317	21 merger	1,276	38 dizziness	89
5 priceless	206	22 shortages	621	39 steward	593
6 voluptuous	89	23 devastated	570	40 luxe	50
7 chamberlain	74	24 empathy	246	41 restful	120
8 elbowed	73	25 singles	767	42 inventive	226
9 bosses	746	26 na	15,391	43 appreciating	15

Figure 27 Trends in Sketch Engine

At the time of this writing, the interface of Sketch Engine has recently been redone. It is evident from the screen captures that a registered user could go back to the original interface. Unfortunately, this researcher had a trial version, and this feature was not available for the trial version; as such, the researcher could not explore the original interface. However, the researcher could see from the user manuals and training guides available on the web that in the original interface, a user could search in text types to filter a search or create subcorpora. Sketch Engine uses the term *text type* to refer to the values (metadata) assigned to structures of the text, for example, source, medium or time (https://www.sketchengine.eu/my_keywords/text-type/). This could then be used to divide corpora into subcorpora. The text types that a user can use to filter is dependent on the text types that were used to encode the corpus. In this example, a user could, for example, choose the publication date and that only texts marked as spoken demographic must be used (Figure 28).

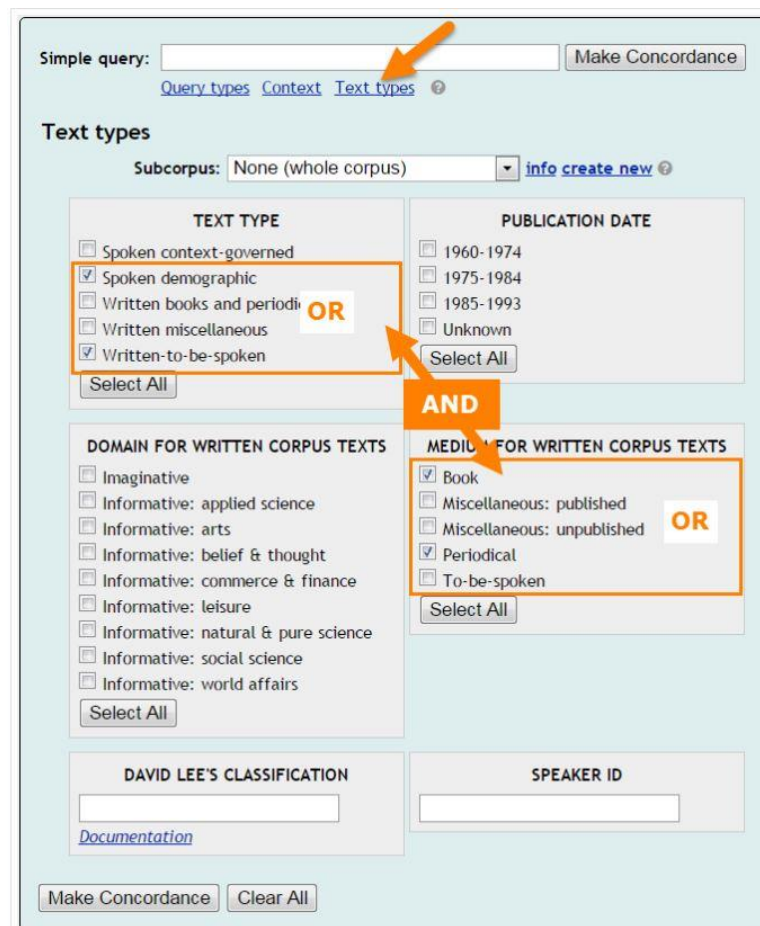


Figure 28 Using text types in Sketch Engine

Though it is evident from the training manuals that the old interface supports filtering on text types, the new interface does not offer this option to a user. The researcher cannot make any comments on personal experience with the old interface, but will make some observations regarding what is evident in the training material. Similar to the other tools from the field of linguistics, SketchEngine could make use of the metadata in a text and allow a user to filter to some extent. However, it seems that the focus of the tool is not on filtering, but analysing the corpus as a whole, as is seen in the new interface.

It is not clear why the option to filter or create a subcorpus was removed. It could be that there are not enough corpora encoded with metadata (text types in SketchEngine), that users did not use this option, or that it was too confusing to use.

SketchEngine can be used effectively to study collocations.

SketchEngine does not provide a graph to visualise the frequency of use of a word over time. This is probably due to the fact that it is not the primary purpose of SketchEngine.

EEBO

Early English Books Online (EEBO) is a collection of early printed works in English and contains more than 125 000 titles (EEBO, n.d.). Users can view the digitised images of the items in this collection, which includes books, play scripts, sermons, public and legal documents, religious material and other special items (Meyer & Eccles, 2016). This collection is subscription-based.

As a separate initiative, the Text Creation Partnership (TCP) created transcriptions of thousands of these items. The EEBO-TCP project has made about 25 000 of the texts (not images) in the EEBO collection available freely (Meyer & Eccles, 2016). These texts are SGML/XML-encoded and fully searchable (Text Creation Partnership, 2019).

This corpus is searchable through some of the tools discussed in this study (e.g. CQPweb and BYU Corpora). In BYU Corpora it is mentioned that the version used by them contains 755 million words in 25 368 texts from the 1470s to the 1690s (BYU Corpora – EEBO, n.d.; Davies, n.d.).

BNC2014

The successor to the BNC, the BNC2014, is developed by Lancaster University and Cambridge University Press (Love et al., 2017). The spoken part of the corpus has been released and, at the time of writing, the written component was still a work in progress (ESRC Centre for Corpus Approaches to Social Science (CASS), n.d.). The BNC2014 is accessible via the CQPweb. The BNC2014 is encoded in XML and follows the recommendations by Hardie (2014) and is explained in the user guide (BNC, 2018). It seems that the latter BNC has moved to a lighter encoding, with fewer tags.

2.7. Evaluation of existing tools

In this section, the tools examined in the previous section will be discussed under the categories of interface design, metadata, search options, filtering, search results, complexity of use, help files and the corpus design.

2.7.1. Interface design

The interface design of each tool will be discussed here, specifically in terms of clarity and how easy it is for a user to understand how to use the tool.

Google Books Ngram Viewer

The interface of the Google Books Ngram Viewer is very typical of a product developed by Google. The screen is simple and clean as seen in Figure 29. There is a prominent search field. Furthermore, there is an example prepopulated in the search field that shows the result of that search in a graph below the search field. Filtering options are displayed below the search field and will be discussed in section 2.7.4.

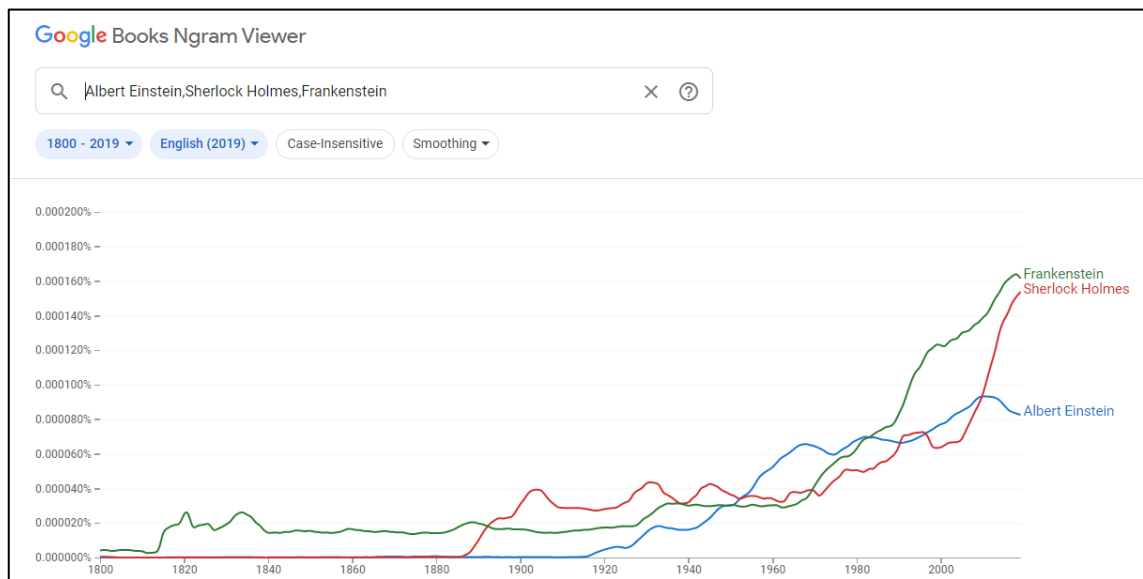


Figure 29 The Google Books Ngram Viewer interface

The interface is simple and the example makes it easy to understand how to use the tool.

HathiTrust+Bookworm

HathiTrust+Bookworm has a simple and clear interface. The interface is shown in Figure 30. The tool opens with two search fields that are prepopulated with an example that searches for usage trends in a large volume of texts. The results of this example query are illustrated in a graph below the search fields. The example query makes it clear how to use this tool. Next to each search field is a filter with a dropdown menu.

This can be used to apply filters to the search. In front of the search fields are a plus and minus sign, to add or remove search fields. Other filtering options are available to the top right of the screen.

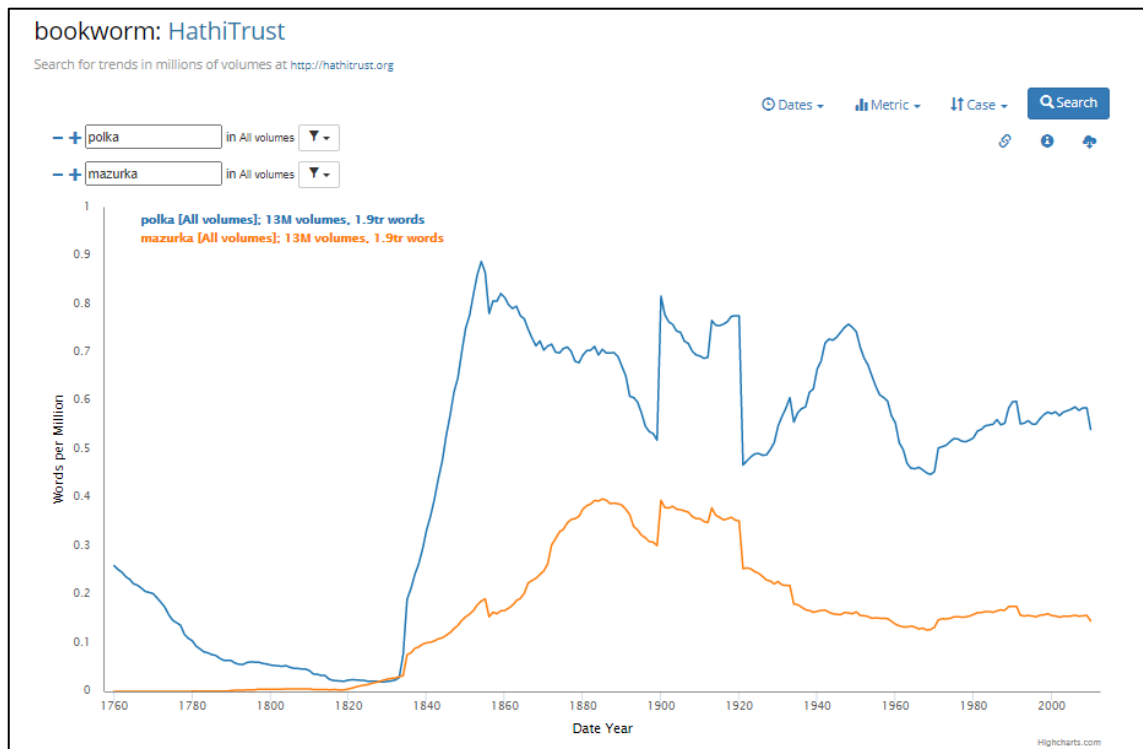


Figure 30 The HathiTrust+Bookworm interface

Perseus Project

The home page of the Perseus Digital Library is fairly clean and can be regarded as a simple web page (Figure 31). There is a menu bar under the title and a search field in the top right-hand corner. News about the project dominates the rest of the screen, with quick links to popular items on the left. From here a user can either use the menu bar to navigate to collections and then to a specific item in a collection or search for an item.

PERSEUS DIGITAL LIBRARY
GREGORY R. CRANE, EDITOR-IN-CHIEF
TUFTS UNIVERSITY

Home Collections/Texts Perseus Catalog Research Grants Open Source About Help

Welcome to Perseus 4.0, also known as the Perseus Hopper.
Read more on the [Perseus version history](#).
New to Perseus? [Click here](#) for a short tutorial.

Perseus Updates

- **May 1, 2018: Individual Developments and Systematic Change in Philology**
At the end of March 2018, my collaborators and I finished enjoying five years of support... [Read more.](#)
- **March 15, 2018: First Version of the Scaife Digital Library Viewer goes live: building the future while remembering a friend**
Announcing the first release of the [Scaife Digital Library Viewer](#), a reading environment for source texts that follow the Canonical Text Services (CTS) data model... [Read more.](#)
- **February 24, 2018: Using Clarin or Dariah to work with historical languages?** [Read more](#)
- **September 11, 2017 (update): Design Sprint for Perseus 5.0/Open Greek and Latin**
Eldarion.com has received the lead contract to develop the new Scaife DL Viewer. In this we build upon work already underway... [Read more](#)

For more read the [full Perseus blog](#)

Release Announcements

- October 2013

Popular Texts

- Caesar, *Gallic War* (English, Latin)
- Catullus, *Carmina* (English, Latin)
- Cicero, *In Catilinam I* (English, Latin)
- Vergil, *Aeneid* (English, Latin)
- Herodotus, *Histories* (English, Greek)
- Homer, *Odyssey* (English, Greek)
- Plato, *Republic* (English, Greek)
- Tom Martin, *Overview of Classical Greek History from Mycenae to Alexander* (English)

Art and Archaeology

Aegina, Temple of Aphaia Silver obol from Athens

Satyr on Attic red figure vase The Bartlett Head

Figure 31 Perseus Digital Library home page

Once a text has been selected, a user can read the text and use the tools provided to help with the study of the text. Examples of search queries are given under the search field.

Voyant Tools

Voyant Tools has a simple and clear interface. First, the user must specify the text(s) to work with. Then the homepage with the different tools opens, each in a separate panel (see Figure 16). The default tools are Cirrus – the word cloud generator, Reader – which allows a user to read the selected text, Trends – which shows the frequency of words in the text, Summary – the tool that provides a synopsis of the text and Contexts – the tool that shows words in context. A user can replace a tool in a panel with a different tool (see Figure 32).

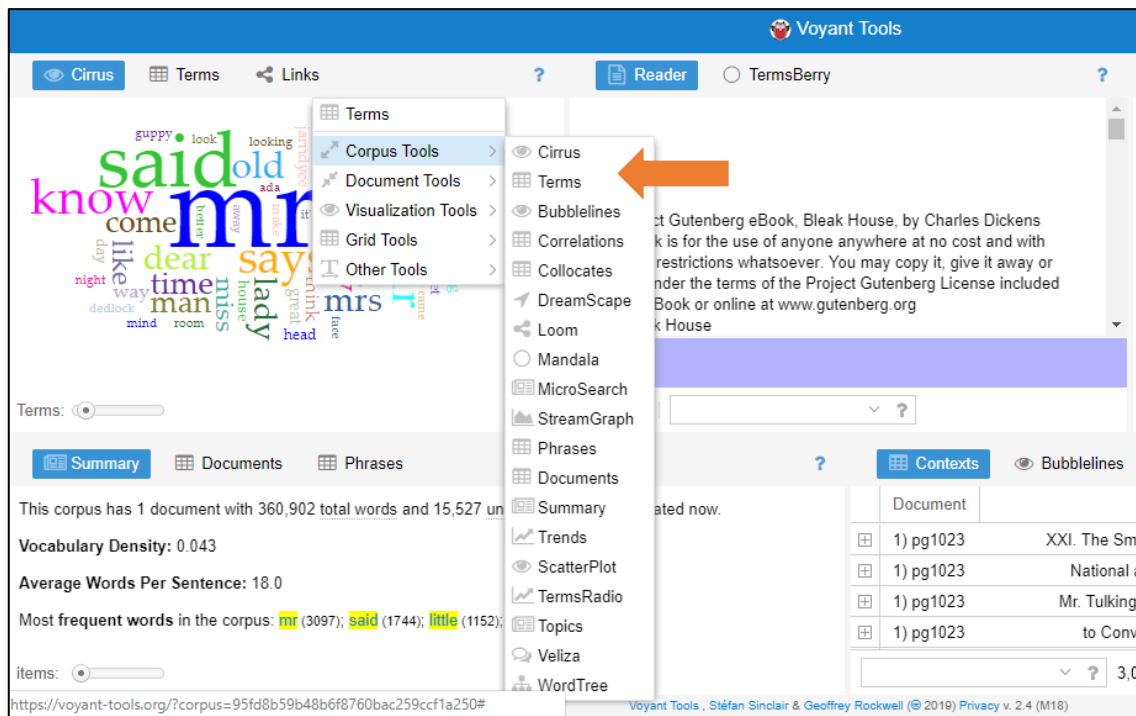


Figure 32 Different tools in Voyant Tools

TXM

TXM is a very powerful tool for analysing corpora and includes numerous functions. As a result, the tool is slightly daunting when a novice opens it for the first time (see Figure 33). It is doubtful whether this tool can be used effectively (if at all) without consulting the help documentation or user guide.

The interface is divided into four sections. The browser panel allows a user to access objects (e.g. corpora). The commands panel gives access to the menu and buttons that can be used to launch commands. The results panel displays the output and the message panel displays relevant messages.

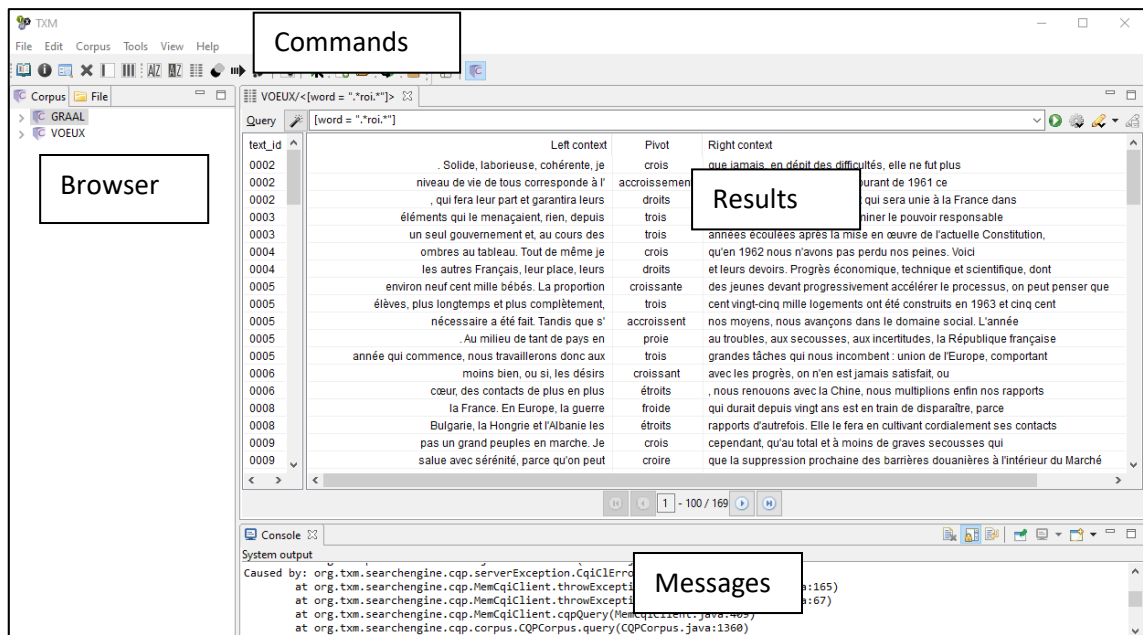


Figure 33 The TXM interface

Commands that can be performed on the corpus are executed through menus and buttons and will be discussed in section 2.7.3.

BNCweb (CQP-edition)

The BNCweb has a simple and intuitive interface. The main menu is on the left of the screen and the search field is large and in the centre of the screen. A user can perform a simple query with no further instructions. For example, in Figure 34, the query *mountain* was entered in the main search field. The ability of the user to enter a simple search query was specifically added to this tool. Due to the complexity of the CQP language, a simplified query language had to be developed so that simple queries could be executed (Hoffmann & Evert, 2006). As such, a user can choose to write a simple query or use CQP syntax.

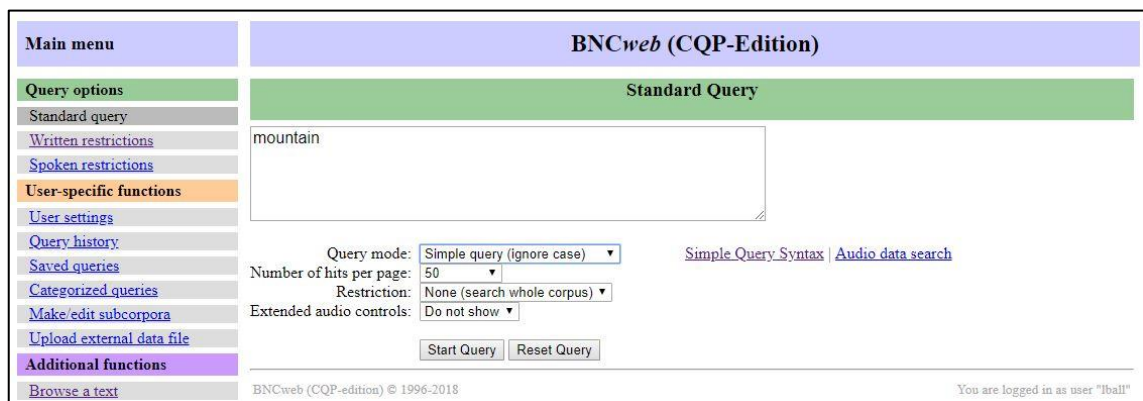


Figure 34 Interface of BNCweb (CQP-edition)

There is no pre-defined example as a user opens the tool, which could have helped novices to learn how to use the tool.

BYU Corpora

The interface to the corpora at corpus.byu.edu is fairly intuitive (Figure 35). However, a user must get used to the layout, options and flow of information. The tabs in the panel at the top of the page (1) allow a user to search, see the frequency counts of the results, the results in context and an overview of the corpus. The search field is clearly visible. The different search options are above the search field (2) and are corpus dependent.

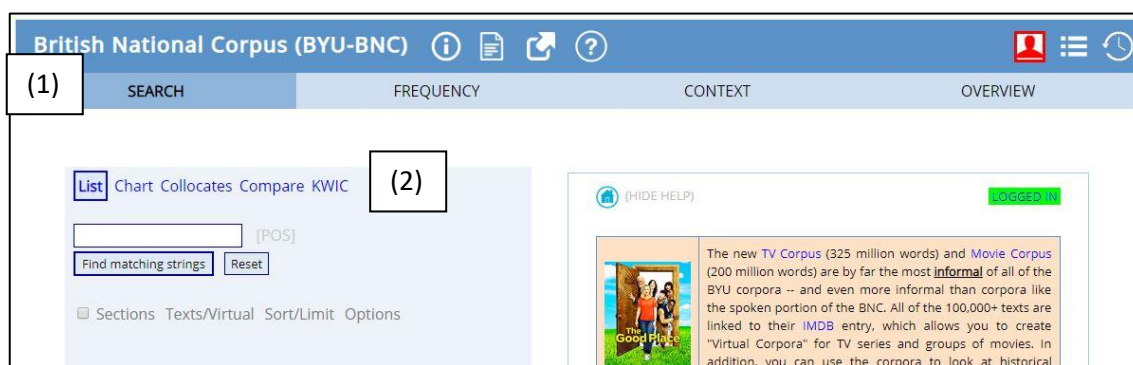


Figure 35 Interface of corpus.byu.edu

A user will typically start a search on the Search tab, proceed to view the frequency counts on the Frequency tab and see more detail on the Context tab.

2.7.2. Metadata

In this section, the use of metadata by the various tools will be discussed. The metadata available should be discussed before the search options, as the metadata will influence the type of searching and filtering that can be done.

Metadata are essentially information about an item, which, amongst other things, enables the item to be found (Riley, 2017: 2). Cultural heritage institutions, such as libraries, are known for creating good, structured metadata (Riley, 2017: 5). Libraries in particular focus on bibliographic metadata, which is detailed information about an item (Riley, 2017: 5).

Bibliographic information is very important; however, as was explained earlier, more detailed information (on a fine-grained level) can be given about an item. As such, this research will also consider the metadata at all the levels as discussed in the previous section.

Google Books Ngram Viewer

One of the main problems with the corpus used in the Google Books Ngram Viewer is the lack of metadata, which has been highlighted by numerous researchers (e.g. Jockers, 2010; Koplenig, 2017: 170). The developers of the Google Books Ngram Viewer do acknowledge that not releasing the bibliography of the corpus is a problem, however they do not have the permission to release that information as that would breach copyright agreements (Culturomics, 2017).

Koplenig (2017: 171) explains that the representativeness of a corpus is problematic and often highly subjective. However, using metadata about the items selected for the corpus can at least help to understand how the corpus was compiled. Metadata can also help to determine if the samples of items at different points in time are the same kinds of things (for example genre and text types) (Koplenig, 2017: 170).

As a result of the lacking metadata in the Google Books Ngram Viewer, Koplenig (2017: 183) questions the results from the study by Michel et al. (2011) in which they investigated the censorship during the Nazi regime. He states that the findings could be as a result of a change in the composition of the underlying data (Koplenig, 2017: 183) and concludes that "the availability of metadata is not just a nice add-on, but a powerful source of information for the digital humanities ... size cannot make up for lack of metadata" (Koplenig, 2017: 183, 184).

The main problem with the metadata in the Google Books Ngram Viewer is its absence. It is also worthwhile to consider the criticism against the quality of metadata in Google Books, because the Google Books Ngram Viewer gets its data (content) from Google Books.

Hitchcock (2013: 17) illustrates the problems of the poor quality of metadata in Google Books by searching for books according to subject, specifically Medieval History. He remarks that his search did not return any books that are related to medieval history. The researcher of this study repeated this experiment and searched for all books with the subject Medieval History. Figure 36 shows this search, as well as the first four results returned by this search. Of the first four results, only the first had a specific subject field and was marked as being about medieval history (Figure 37).

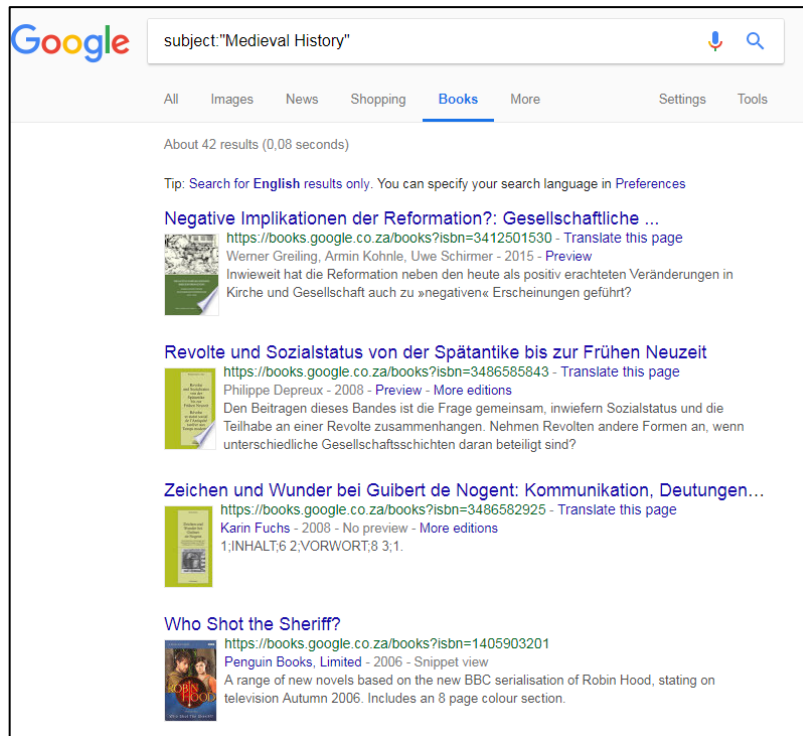


Figure 36 Searching on Google Books by subject

Bibliographic information	
Title	Negative Implikationen der Reformation?: Gesellschaftliche Transformationsprozesse 1470–1620 <i>Volume 4 of Quellen und Forschungen zu Thüringen im Zeitalter der Reformation</i>
Editors	Werner Greiling, Armin Kohnle, Uwe Schirmer
Contributor	Uwe Schirmer
Publisher	Böhlau Verlag Köln Weimar, 2015
ISBN	3412501530, 9783412501532
Length	438 pages
Subjects	History > Europe > Medieval History / Europe / Medieval History / Medieval

Figure 37 Subject metadata for the first search result

It seems that, in Google Books, a simple keyword search that searches on the content as opposed to the metadata returns more relevant results (Figure 38).

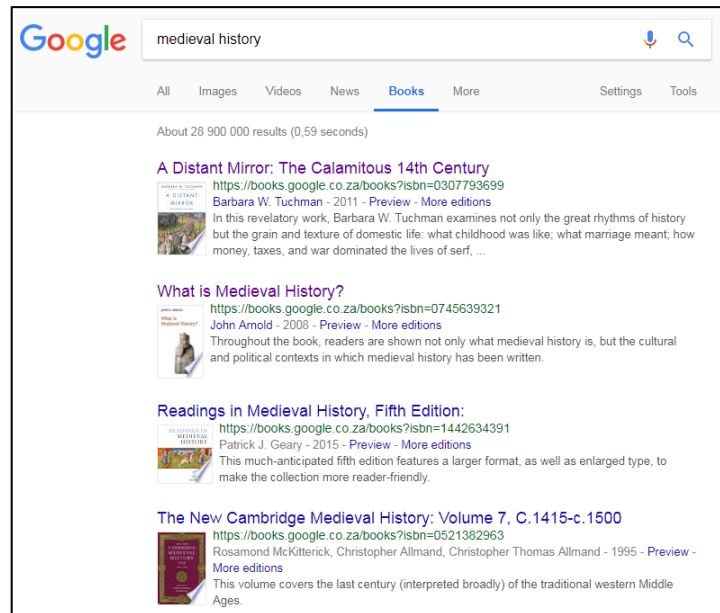


Figure 38 Searching on Google Books through keywords

Although Google Books has been criticised for the quality of the data and metadata in its collection, others have argued that the data are of sufficient quality to be used for meaningful linguistic research (Davies, 2014: 402).

It should be noted that the corpus of the Google Books Ngram Viewer does not work on the entire Google Books, as some of the books have been excluded (Culturomics, 2017). This means that although criticism of the metadata of Google Books is valid, the criticism cannot be applied directly to the Ngram Viewer corpus; it should just be noted and considered when using Google Books Ngram Viewer.

The Google Books Ngram Viewer team also acknowledge that metadata dates used in the corpus are not perfect, especially when they started with the project. However, they explain that they have worked on this problem and systematically excluded books whose metadata seemed incorrect, leaving them with an error rate of 1 in 20 (Culturomics, 2017). This should be kept in mind when using Google Books Ngram Viewer.

The bibliographic metadata used in this corpus are the year that the work has been published as well as the language that the book is in so that the different language corpora could be created.

Although no bibliographic metadata are given in the Google Books Ngram Viewer, there are metadata on a morphological and syntactic level. In addition to raw n-grams, morphologically and syntactically annotated n-grams can be extracted, as part-of-speech tagging is performed and head-modifier dependencies are noted (Lin et al.,

2012: 171-173). The tags used in the corpus are `_NOUN_`, `_VERB_`, `_ADJ_`, `_ADV_`, `_PRON_`, `_DET_`, `_ADP_`, `_NIM_`, `_CONJ_`, `_PRT_`, `_ROOT_`, `_START_`, `_END_`. These annotations allow for advanced search options and will be discussed in the next section.

According to the information given about the Google Books Ngram Viewer, the part-of-speech tags and dependency relations are generated automatically (Google Books Ngram Viewer Info, 2020). The estimation is that there should be about a 95% accuracy rate for part-of-speech tagging and around 75% accuracy for dependencies and that this should be taken into account when drawing conclusions from the data.

As was noted previously, from the second version, n-grams do not span sentence boundaries. It is assumed that sentences are determined automatically. The accuracy of the algorithm used to determine sentence boundaries should also be taken into consideration when analysing results from this tool.

Another point of criticism is that no distinction is made between different parts of a book. For example, Underwood (2015b: 6) points out that late-nineteenth-century novels often end with advertisements and as such, there is a peak in the appearance of certain words, such as *cloth*, during that period.

There are no semantic annotations and there is no functional encoding.

HathiTrust+Bookworm

As was explained earlier, this tool searches in the HathiTrust Digital Library. This library contains a significant amount of bibliographic metadata per volume. The bibliographic data in the dataset itself uses MARC data from HathiTrust but gets most of the bibliographic metadata from Hathifiles (tab-delimited text files that describe items in HathiTrust) (Kinnaman & Koehl, 2018).

Due to the extensive metadata available, the tool allows a user to filter the data to search according to certain metadata fields. This will be discussed in depth in the next section. However, although the tool allows a user to filter according to metadata, one could question the strength of the underlying metadata. As was mentioned in section 2.2, the HathiTrust Digital Library includes items that were digitised by the Google Books project. The same criticism that applies to the metadata in Google Books will then apply to the items from Google Books in HathiTrust. One example will be used here to explore the issue of metadata.

The example here is to search for *polka* where the class (subject) is Language and Literature, compared to *polka* where the class is general works. A user can click on a point on the diagram to link to examples of items that were used in the analysis. A list of items in which the term “polka” appears in 1848 is shown in this example (Figure 39). The item “Old heads...” was selected by the researcher and is shown in Figure 40. This item has been digitised by Google. The full catalogue record for this item can be viewed (Figure 41), but there is no indication in the displayed metadata that this item indeed has a subject term related to Language and Literature. This information could be obtained from a more extensive metadata record, for example, the associated MARC record. Alternatively, there are some aspects which could be worked out by an algorithm.

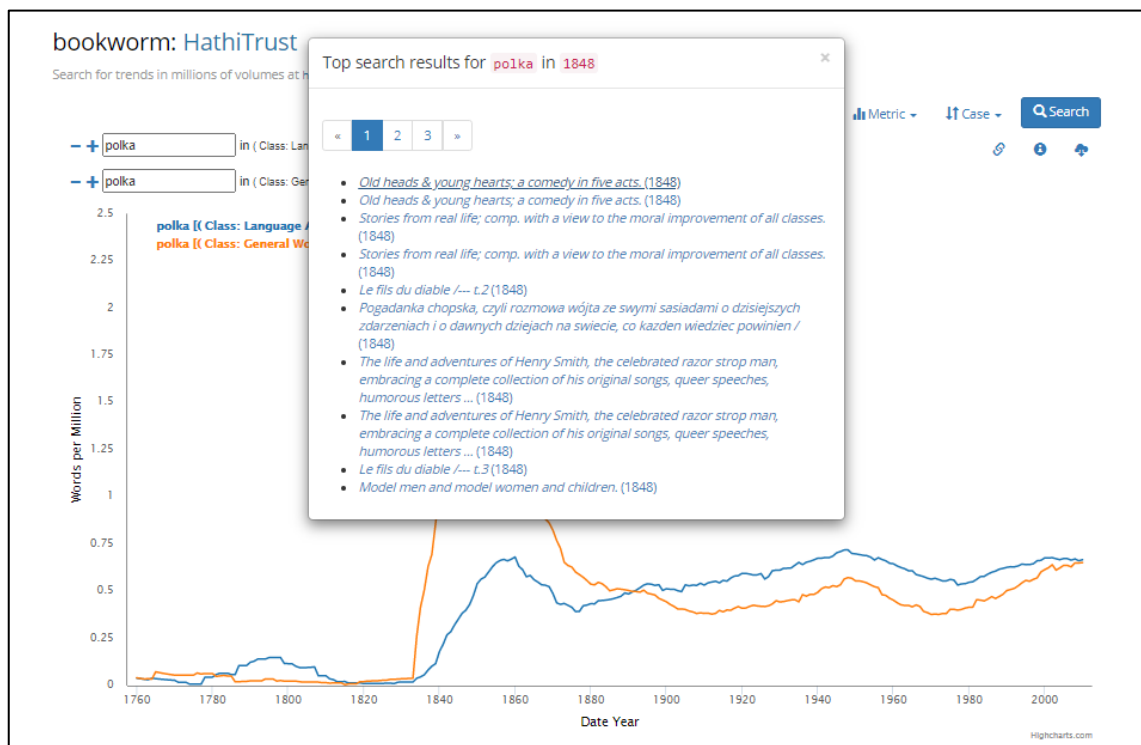


Figure 39 A list of items where "polka" appears in the HathiTrust+Bookworm

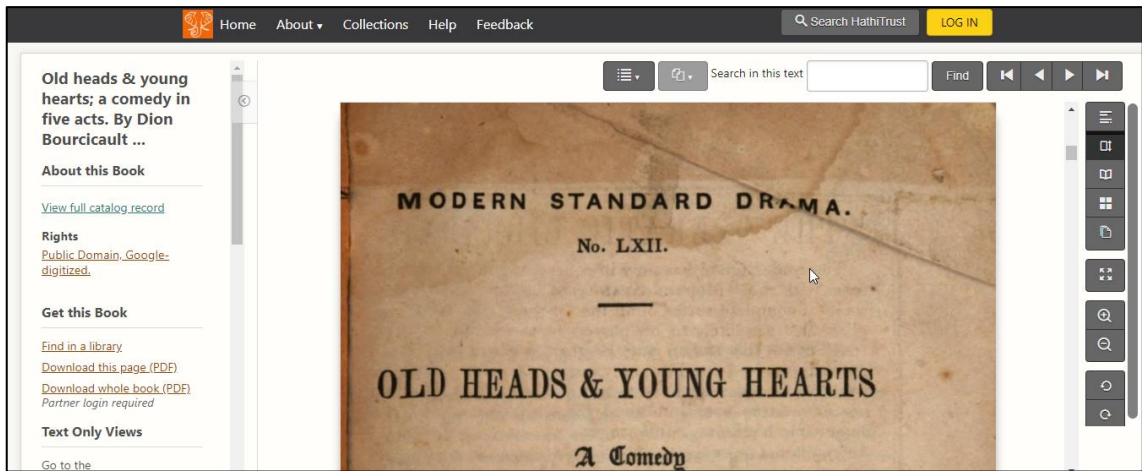


Figure 40 The item "Old heads..." in the HathiTrust Digital Library

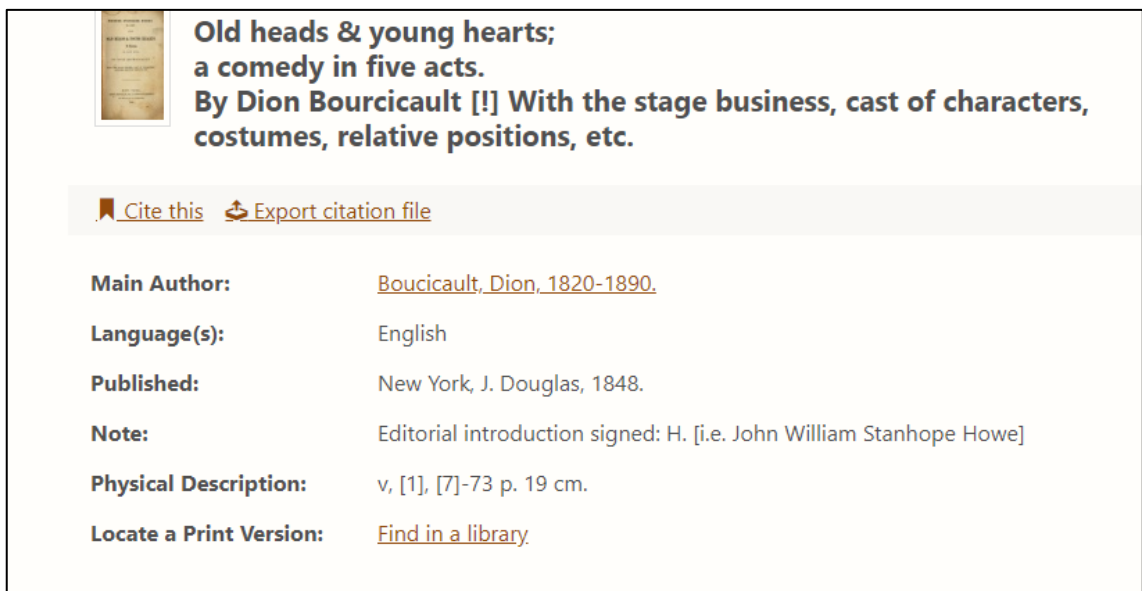


Figure 41 The catalogue record for "Old heads..."

Although the tool allows a user to filter according to subject, if the underlying metadata are incorrect, the results will not be reliable.

Furthermore, the fields according to which a user can filter often have dropdown menu options, as seen in Figure 42, for the field "Literary Form". However, the data in the dropdown menu are sometimes strange or out of place. The dropdown menu shows different types of literary forms, such as short stories and letters, but it also shows "University of Michigan" as an option, which is not typically recognised as a literary form.

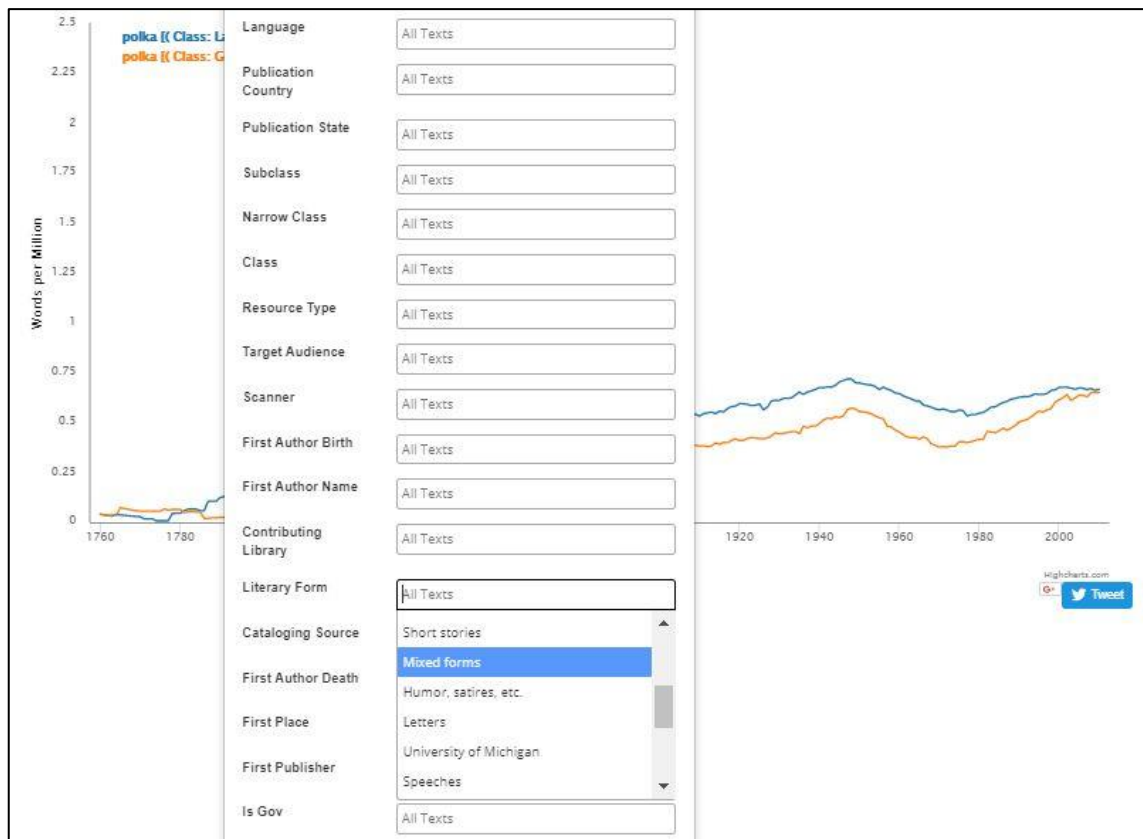


Figure 42 Dropdown in HathiTrust+Bookworm

It should be noted that the tool itself includes the ability to filter according to certain fields. From the literature it is apparent that a tool designed to search in textual corpora should include some mechanism to filter according to metadata. The stronger and cleaner the underlying metadata are, the more reliable and powerful this tool will be.

In terms of bibliographic metadata, it is apparent that this tool engages with bibliographic metadata and offers filtering options according to these fields. However, the filtering in this tool is limited to bibliographic data and does not offer filtering on a text-level (functional encoding). This is probably due to the fact that the HathiTrust Research Center supports non-consumptive research, that is work that is in copyright is not consumed by the user but is used in computational analysis.

This online tool does not support searching according to morphological, syntactic or semantic metadata.

It should be noted that the Extracted Features dataset from the HathiTrust Research Center and code to set up a bookworm using this dataset is available on GitHub. This dataset includes page-level and word-level information. For example, part-of-speech tags are assigned to words and the frequency counts are recorded. Apache OpenNLP was used for sentence segmentation, tokenisation and part-of-speech tagging, using

the Penn Treebank POS tags (Kinnaman & Koehl, 2018). This could open up searching and filtering options.

Perseus Project

Texts in the Perseus Digital Library have bibliographic data, functional metadata and some morphological metadata.

Each item in the library is described with a certain amount of bibliographic data in order to identify the item. It is not clear if a specific metadata standard was followed, but given that the focus is on classical texts, it might not be appropriate to follow a specific standard.

Most of the texts are encoded in TEI (see Figure 43) (Rydberg-Cox et al., 2000). The encoding is used to describe the text, for example, what parts of the text are notes, what are highlights, where are paragraphs. The schema differentiates between form and function. For example, sections of texts that are highlights are marked. However, the style in which these sections should be highlighted is noted separately as attributes (e.g. italics).

```
<TEI.2>
<text>
  <group>
    <text n="Catil.">
      <body>
        <div1 type="Speech" n="1" org="uniform" sample="complete">
          <milestone n="1" unit="chapter"/>
          <milestone n="1" unit="section"/>
          <p>
            <reg>quo</reg>
            usque tandem abutere, Catilina, patientia nostra? quam diu etiam furor iste tuus nos
            <note place="unspecified" anchored="yes">
              nos
              <hi rend="italics">om. A et Iulius Victor</hi>
              (
              <hi rend="italics">Rhet. M. p.</hi>
              439):
              <hi rend="italics">post</hi>
              diu
              <hi rend="italics">hab. bs</hi>
            </note>
            eludet? quem ad finem sese effrenata iactabit audacia?
            <reg>nihilne</reg>
            te nocturnum praesidium Palati, nihil urbis vigiliae, nihil timor populi, nihil concursus bonorum
            omnium, nihil hic munitissimus habendi senatus locus, nihil horum ora voltusque moverunt?
            <reg>patere</reg>
            tua consilia non sentis, constrictam iam horum omnium scientia teneri coniurationem tuam
            <note place="unspecified" anchored="yes">
              tuam
              <hi rend="italics">om. CAV</hi>
            </note>
            non vides?
            <reg>quid</reg>
            proxima, quid superiore nocte egeris, ubi fueris, quos convocaveris, quid consili ceperis quem
            nostrum ignorare arbitraris?
            <milestone n="2" unit="section"/>
            O tempora, o mores!
            <reg>senatus</reg>
            haec intellegit, consul videt; hic tamen vivit.
            <reg>vivit</reg>
            ? immo vero etiam in senatum venit, fit publici consili particeps, notat et designat oculis ad
            caedem unum quemque nostrum.
            <reg>nos</reg>
            autem fortes viri satis facere rei publicae videmur, si istius furorem ac tela vitamus
            <note place="unspecified" anchored="yes">
              vitamus
```

Figure 43 TEI encoding of *Against Cataline* by Cicero in the Perseus Project

An interesting observation in Figure 44 is that the text of this sample is in Latin, but the author has used a Greek word, which is encoded as a foreign language.

```
▼<TEI.2>
  ▼<text>
    ▼<body>
      ▼<div1 type="Book" n="1" org="uniform" sample="complete">
        ▼<div2 type="letter" n="1" org="uniform" sample="complete">
          <milestone n="1" unit="section"/>
          ▼<p>
            L. Clodius, tribunus plebis designatus, valde me diligit vel, ut
            <foreign lang="greek">ἐμφοτικώτερον</foreign>
            dicam, valde me amat.
            <reg>quod</reg>
            cum mihi ita persuasum sit, non dubito (bene enim me nosti) quin i
            <reg>nihil</reg>
```

Figure 44 Encoding of a foreign language

One of the problems readers of a text may experience is that words in a text can be used in an inflected form. The inflected form is not typically the headword in a dictionary, for example, *taught* as an inflected form of *teach* will not be the headword in a dictionary. The Perseus Project addresses this problem. Latin or Greek words are parsed, and the resulting data are stored in a database (Rydberg-Cox et al., 2000). The words in Greek or Latin texts are compared with words in the database and links are formed between words and data in the database. As a result, a user can click on a word in the text to get more information about that word, including a suggestion of the most likely meaning in that context, definitions, links to full dictionary entries, as well as other grammatical information such as the part-of-speech and inflection. Figure 45 shows a section from a letter by Cicero and the word *notum* that the user wishes to know the meaning of. Figure 46 shows the page for this word and it is suggested by the tool that the most likely meaning in this context is *known*. The tool uses statistical methods to determine the most likely meaning in the context, as such, it is not necessarily correct. (See the message from the tool in Figure 46.) In this example, the correct option is more likely an adjective. The tool does provide an option for users to vote for the correct meaning for a word in context, but there are no user votes in this example. This could be beneficial in a system where there is extensive user participation.

Notus2 the south wind
 (Show lexicon entry in Lewis & Short Elem. Lewis) (search)

notum	noun sg masc acc	no user votes	7.6%
notum	noun pl masc gen poetic	no user votes	5.6%
notum	adj sg neut voc	no user votes	6.3%
notum	adj sg neut nom	no user votes	4.7%
notum	adj sg neut acc	no user votes	4.7%
notum	adj sg masc acc	no user votes	3.7%
notum	adj pl neut gen poetic	no user votes	3.5%
notum	adj pl masc gen poetic	no user votes	3.5%

Word Frequency Statistics ([more statistics](#))

Words in Corpus	Max	Max/10k	Min	Min/10k	Corpus Name
9,789	26	26.56	0	0	M. Tullius Cicero, Letters to and from Brutus

nosco to get knowledge of, become acquainted with, come to know, learn, discern
 (Show lexicon entry in Lewis & Short Elem. Lewis) (search)

notum	part sg perf pass neut voc	no user votes	3.3%
notum	part sg perf pass neut nom	no user votes	2.9%
notum	part sg perf pass neut acc	no user votes	2.9%
notum	part sg perf pass masc acc	no user votes	2.9%
notum	part pl perf pass neut gen poetic	no user votes	3%
notum	part pl perf pass masc gen poetic	no user votes	3%
notum	noun sg supine neut nom	no user votes	2.9%

Word Frequency Statistics ([more statistics](#))

Words in Corpus	Max	Max/10k	Min	Min/10k	Corpus Name
9,789	38	38.819	1	1.022	M. Tullius Cicero, Letters to and from Brutus

notus known
 (Show lexicon entry in Lewis & Short Elem. Lewis) (search)

notum †	noun sg masc acc	no user votes	7.6%
notum	noun pl masc gen poetic	no user votes	5.6%
notum	adj sg neut voc	no user votes	6.3%
notum	adj sg neut nom	no user votes	4.7%
notum	adj sg neut acc	no user votes	4.7%
notum	adj sg masc acc	no user votes	3.7%
notum	adj pl neut gen poetic	no user votes	3.5%
notum	adj pl masc gen poetic	no user votes	3.5%

† This form has been selected using statistical methods as the most likely one in this context. It may or may not be the correct form. ([More info](#))

Word Frequency Statistics ([more statistics](#))

Words in Corpus	Max	Max/10k	Min	Min/10k	Corpus Name
9,789	26	26.56	0	0	M. Tullius Cicero, Letters to and from Brutus

Figure 46 The suggested meaning of a word in the Perseus Project

Voyant Tools

In general, the tool seems to focus on the analysis of plain text and there is very little interaction with metadata. As a user will select the texts to upload, the user will presumably be familiar with the bibliographic detail of the selected texts. The tool itself does not interact with bibliographic data as such. Nor are there options to search using morphological data, such as part-of-speech.

However, Voyant Tools can import texts in various formats, including XML and TEI. This means that the tool does accommodate texts where certain textual features have been encoded.

TXM

TXM was developed to be able to create and analyse tagged and structured corpora. In other words, texts that contain additional data about the texts (e.g. data about the morphological properties of the words in the text, or data about the structure of the text or bibliographic data) can be analysed. The metadata applied to the texts in the corpora are used by TXM when searching and analysing the corpora.

The design of the corpora that can be used in TXM is explained in the TXM processing model, as discussed in TXM User Manual (2018). The TXM processing model consists of several components. Each corpus may contain texts, which have associated bibliographic metadata. Each text may contain internally nested structures that describe the structure of the text. The smallest unit in each text is a word that can be in the text structures and can have properties of its own. (The TXM documentation refers to the encoding of in-text features as structural information, but not all features that are encoded at this level are strictly structural.)

Detailed guidelines specify the format of the data that can serve as input to TXM. This refers to how the additional data, as explained in the processing model, can be encoded. For example, texts can be encoded in XML or TEI (TXM User Manual, 2018). TXM offers specific support for TEI encoded texts, for example, the "TEI P5 BFM" TXM import module contains scripts that can be adapted to specific TEI encoding usage (TEI wiki – TXM, 2016).

Texts where the properties of words have been pre-encoded can be imported, but TXM can tag each word with morphological data (TXM User Manual, 2018).

As each corpus can have its own metadata, some examples of metadata will be given by looking at the two corpora that are included with TXM, namely, the GRAAL corpus and the VOEUX corpus.

The GRAAL corpus is encoded in XML to describe properties of the text and words are tagged with morphological information.

The GRAAL corpus consists of only one manuscript, so the bibliographic information is limited. The text in the GRAAL corpus is encoded with 20 structural units. For example, body, paragraph, sentence or direct speech.

Each word (lexical unit) is tagged with additional data, namely, the word as it was written, the diplomatic form, the order number of the word in the text, part-of-speech, the concordance's reference and the nesting level of direct speech. The part-of-speech tags used are from the CATTEX2009 tag set (TXM User Manual, 2018). Two more attributes were added in the 2019 corpus, namely, *aggl* and *facs*. These attributes were not discussed in the documentation at the time of writing this study.

The information regarding the encoding is shown in Figure 47.

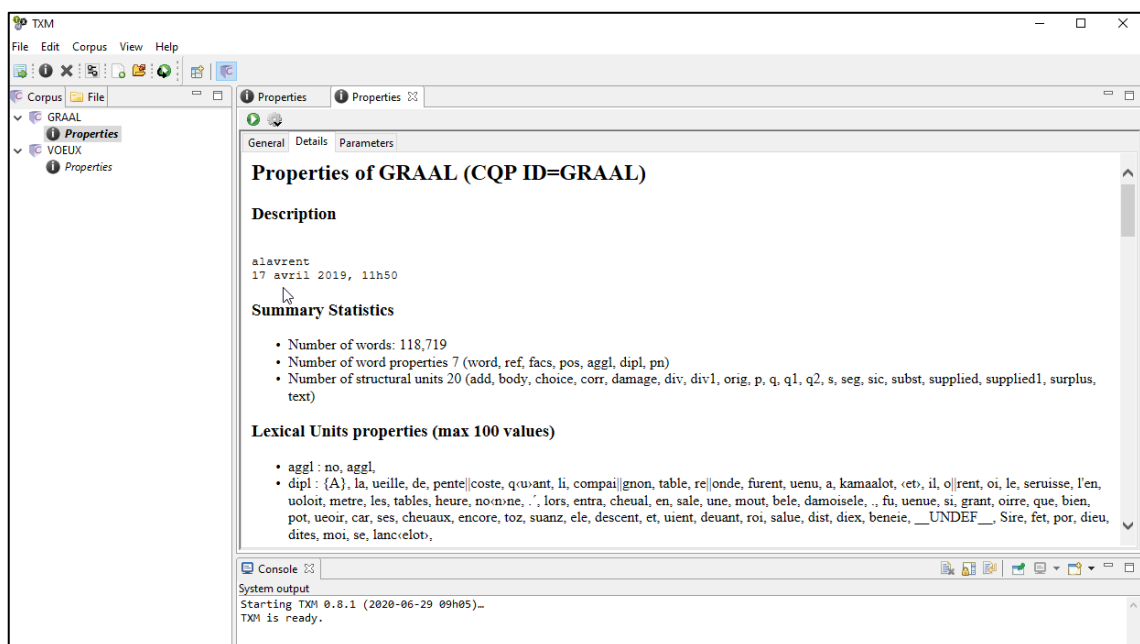


Figure 47 Information about the encoding in the GRAAL corpus

In the VOEUX corpus, each text has as bibliographic data the year of the address as well as the president giving the address. The structural information for each text includes the text itself, paragraphs, sentences and line beginnings. The words (lexical units) contain information regarding the written form of the word, the part-of-speech category, the lemma, the row number in which the word appears, the sentence,

paragraph and line beginning number. The part-of-speech category and lemma were assigned using the program included in TXM, using the fr.par model as tagset.

The information regarding the encoding is shown in Figure 48.

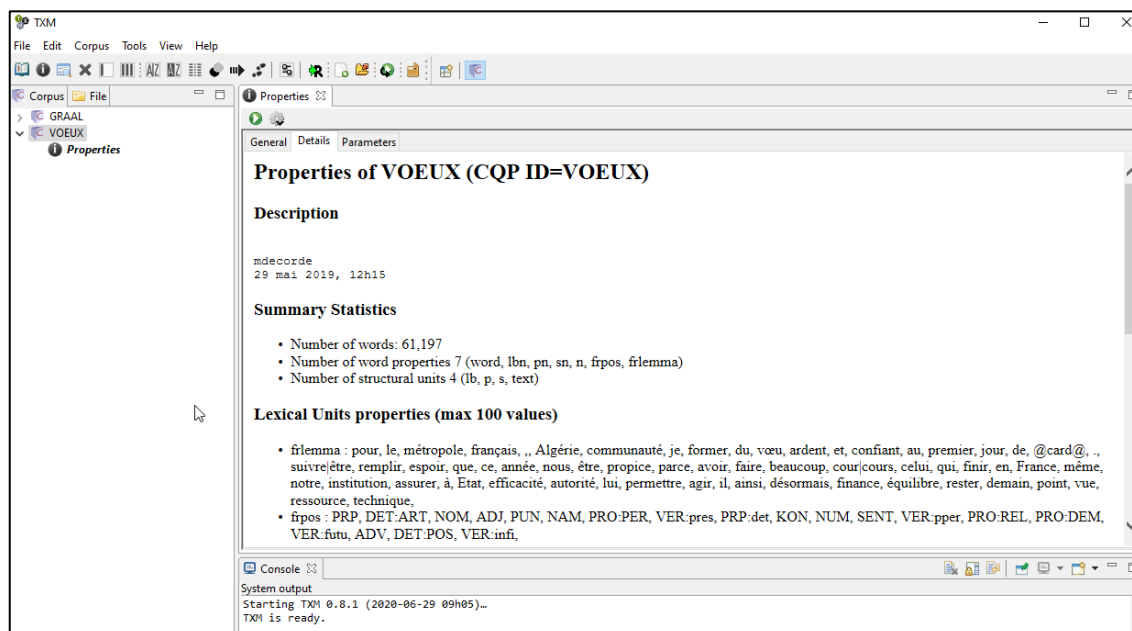


Figure 48 Information about the encoding in the VOEUX corpus

Section 2.7.3 will look at how this metadata are used by TXM to retrieve and analyse data.

BNCweb (CQP-edition)

As the BNCweb is a tool that uses the BNC as corpus, the metadata of the BNC will be discussed here. The BNC XML edition includes grammatical, structural as well as bibliographic data and is encoded in TEI (Burnard, 2009).

CLAWS4, an automatic tagger, is used to divide the text into individual words (tokenise) and assign part-of-speech tags to the words (Burnard, 2007). The Template Tagger is used to enhance the output of CLAWS4 (Burnard, 2007). The BNC is tagged with the C5 tagset, also known as the BNC Basic Tagset (Burnard, 2007). Though a very successful tagger, some tagging errors have been pointed out by Mitton et al. (2007).

Various textual structures or phenomena are encoded in the BNC according to the TEI specification. Different guidelines for the written and spoken texts are given (Burnard, 2007). It is beyond the scope of this research to discuss all the elements used in the BNC, but some structures or phenomena that are encoded and widely used in analysis

are: sentences, paragraphs, speaker, heading, quotation, list, highlighted text and multiword units. An example of how encoding is done in the BNC is given by Burnard (2007) (Figure 49). It is not a complete example, amongst other things the header is not included, but it gives an indication of how grammatical information (e.g. part-of-speech tags) and structural units (e.g. headings, sentences) are encoded.

```

<wtext type="FICTION">
<pb n="5"/>
<div level="1">
<head>
<s n="1">
<w c5="NN1" hw="chapter" pos="SUBST">CHAPTER </w>
<w c5="CRD" hw="1" pos="ADJ">1</w>
</s>
</head>
<p>
<s n="2">
<c c5="PUQ">'</c>
<w c5="CJC" hw="but" pos="CONJ">But</w>
<c c5="PUN">,</c>
<c c5="PUQ">' </c>
<w c5="VVD" hw="say" pos="VERB">said </w>
<w c5="NP0" hw="owen" pos="SUBST">Owen</w>
<c c5="PUN">,</c>
<c c5="PUQ">'</c>
<w c5="AVQ" hw="where" pos="ADV">where </w>
<w c5="VBZ" hw="be" pos="VERB">is </w>
<w c5="AT0" hw="the" pos="ART">the </w>
<w c5="NN1" hw="body" pos="SUBST">body</w>
<c c5="PUN">?</c>
<c c5="PUQ">'</c>
</s>
</p>
.....
</div>
</wtext>

```

Figure 49 Example of encoding in the BNC (Burnard, 2007)

Typical bibliographic information for each text included in the corpus is stored in the header section of that text (Burnard, 2007). It is beyond the scope of this study to discuss the detail of the headers in this corpus. Detailed information about the header and its structure is given by Burnard (2007). An example of the bibliographic information for one of the texts as displayed in the BNCweb tool is shown in Figure 50.

BNC header information for file A04	
Title:	Art criticism: a user's guide. Sample
Spoken or Written:	Written
Number of Words (tagged items):	39,163
Average sentence length (<w>-tags per <s>-unit):	24.1598
Derived text type:	Academic prose
Genre:	W:ac:humanities_arts
Text type:	Written books and periodicals
Publication date:	1985-1993
Age of Author:	unknown
Domicile of Author:	UK and Ireland
Sex of Author:	Male
Type of Author:	Sole
Age of Audience:	Adult
Text Domain:	Informative: Arts
Perceived level of difficulty:	High

Figure 50 Header information for a text in the BNC

One of the criticisms of the BNC was that it had “no text categorisation for written texts beyond that of domain, and no categorisation for spoken texts except by context and demographic/socio-economic classes” (Lee, 2001: 51). (An example of domain in the BNC is “written > informative > pure science”. See Lee (2001) for more information.) As such, an enhancement was the development of an index that contains 70 genres, for example, “academic prose: humanities”, “school essays”, “drama” (Lee, 2001). The BNC therefore includes information about the genre and domain of each text. This is valuable metadata when filtering.

BYU Corpora

The metadata will depend on the corpora used. The metadata for the BNC has already been discussed; however, the way it is used in searching and filtering in corpus.byu.edu will be discussed in the next section.

The iWeb corpus has little metadata. As it consists of webpages, it does not have bibliographic metadata about the texts (webpages) that make up the corpus. It also does not contain structural metadata of the texts. It is possible to search according to certain morphological properties, such as lemmas and parts-of-speech, but it is not evident from the tool itself how the annotation was done or what tagset was applied.

2.7.3. Search options

In this section the different search options that a tool offers will be discussed. This includes the use of fields to enter search terms, the ability to search for a word or a phrase, the ability to search for words or sections of texts with specific properties and the ability to construct complex queries using a query language or features of a query language (e.g. truncation).

Google Books Ngram Viewer

As was seen in Figure 29, the Google Books Ngram Viewer offers a simple search field, where a user can enter comma-separated phrases. The separated phrases are treated as different n-grams to search for and the results are drawn as separate lines on the graph below the search field. N-grams consisting of five tokens are supported, enabling phrase searching to a limited extent. Filtering by date and language will be discussed in the next section. A user can also use the graphical user interface to specify that a search must be case-insensitive.

Apart from these simple search options, Google Books Ngram Viewer offers some advanced search options through commands that can be entered in the search field. These advanced options include using wildcards in queries, searching for inflections, specifying part-of-speech tags or dependency relations. Some examples of their use will be discussed here. Other functions of the Google Books Ngram Viewer, such as adding n-grams, are not relevant to this study and will not be discussed.

An asterisk is used as a wildcard to search for substitutes in a specific phrase. For example, *bread and ** will return the top ten words used with *bread and* ____, such as *bread and wine*, *bread and butter* and *bread and water* (see Figure 51). Wildcard characters cannot be used to replace a part of a word, for example, to search for all words ending in *ly*.

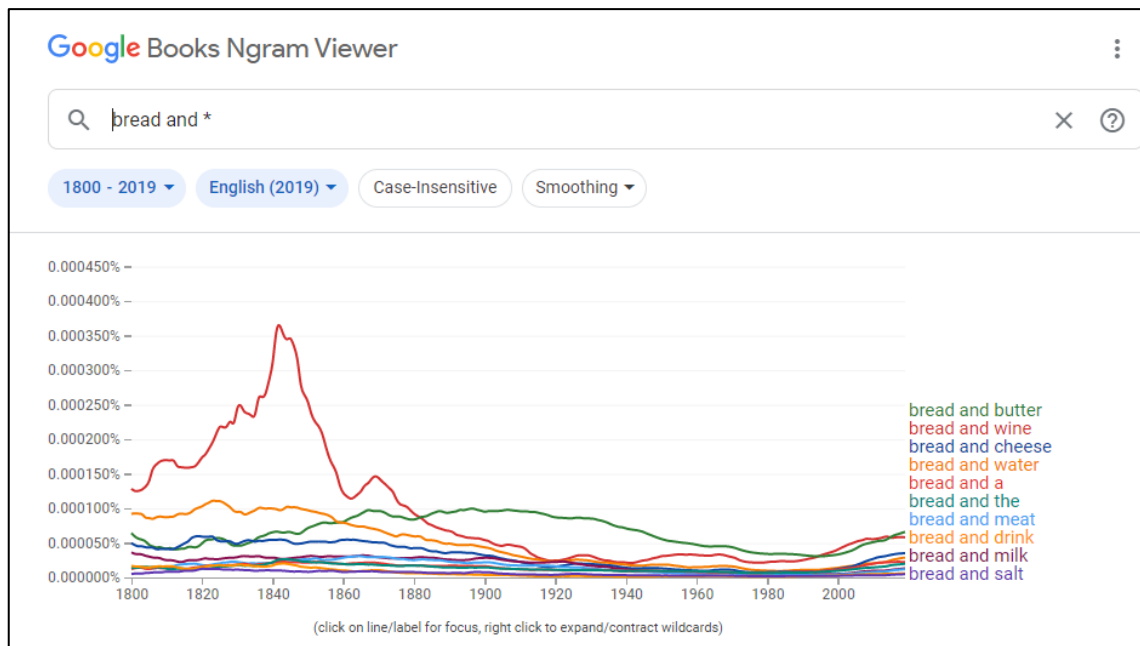


Figure 51 The use of wildcards in Google Books Ngram Viewer

It is possible to search for inflected forms by using *_INF* in a query. For example, *run_INF* will search for *run*, *runs*, *running*, *ran*.

The part-of-speech tags (e.g. *_NOUN_*) applied to the corpus can be used to search in different ways. Firstly, it can be used to indicate that a word must be in a certain word class, for example, a user can search for *play* as a verb by using *play_VERB*.

Secondly, it can be used to indicate that a certain word class must appear somewhere in a phrase, for example, *lost a_NOUN_*, will search for instances where *lost a* is followed by a noun (e.g. *lost a leg*, *lost a battle*, *lost a friend*) and exclude instances where *lost a* is followed by other word classes (e.g. *lost a lot*, *lost a great*, *lost a large*).

A user can search for modifier dependencies, for example, searching for *hair=>long* which will reveal how often hair is described as long.

A user can mix wildcards, searching for inflections, parts-of-speech or dependencies to a limited degree. There is a note from the developers that states that a user cannot freely mix wildcard searches, inflections and case-insensitive searches for one particular n-gram (Google Books Ngram Viewer Info, 2020). One combination that is allowed is to combine a wildcard with a part-of-speech tag, for example, to know what the most common nouns after *lost a* are, the query *lost a *_NOUN*. However, a combination such as *run_INF * race* is not valid.

The search can be case sensitive or case insensitive.

HathiTrust+Bookworm

Figure 30 shows the two search fields on the main page of the HathiTrust+Bookworm. A user enters search terms in these fields and can add more fields as desired. This tool offers comprehensive filtering, which will be discussed in the next section.

The HathiTrust+Bookworm only allows a user to search and compare single words and does not support phrase searching. There is no option to create complex queries with any type of query language. A user cannot search using morphological or syntactic annotations. The search can be case sensitive or case insensitive.

Perseus Project

A user can search for word(s) in texts to retrieve texts containing the desired word(s) (1 in Figure 52) or search for information for specific words (2 in Figure 52). Option 1 results in a list of texts that a user can choose from. A user can refine the results by searching for other word(s) in the results. Searching in option 2 results in a list of the different meanings of the word, as well as other grammatical information. Other search tools are also included, namely, searching in the English definition, searching in specific dictionaries, searching for artefacts or images, searching for specific places, people or dates.

The screenshot displays the Perseus Search Tools interface. The top navigation bar includes links for Home, Collections/Texts, Perseus Catalog, Research, Grants, Open Source, About, and Help. The main content area is divided into two columns. The left column, labeled '1', is titled 'General Search Tools' and contains a search form for 'Search the collections'. It includes a dropdown for 'Search in' (set to 'Latin'), a text input for 'containing all of the words' (with 'suis' entered), and a 'Search' button. Below this are sections for 'English-to-[Language] lookup', 'Dictionary Entry Lookup', 'Art & Archaeology Search', and 'Named Entity Search Tools' (with 'Search places' and 'Search people' options). The right column, labeled '2', is titled 'How to enter text in Greek:' and shows a grid of Greek letters and their codes. Below this is a 'Word Study Tool' with a text input and a 'Go' button, and a 'Vocabulary Tool' link.

Figure 52 Search options in Perseus Project

One of the advantages of this tool is that it allows the user to search for all forms of an inflected word. Perseus Project does not allow the user to search in sections of the texts (as encoded in TEI), nor is it possible to search for a word with a specific part-of-speech tag or with specific syntactic properties. There is no query language.

Voyant Tools

Practically, Voyant Tools is a platform of various tools, each displayed in its own panel. Tools where a search option is applicable (e.g. to search for trends of words) has a search field at the bottom of the panel (see Figure 53). A user can search for a single word or a phrase and there are some features that can be used to construct a query. An asterisk can be used as a wildcard and proximity operators can be used to specify the distance between words. For example, Figure 54 shows the occurrences of the words that start with *murder-* in the text.

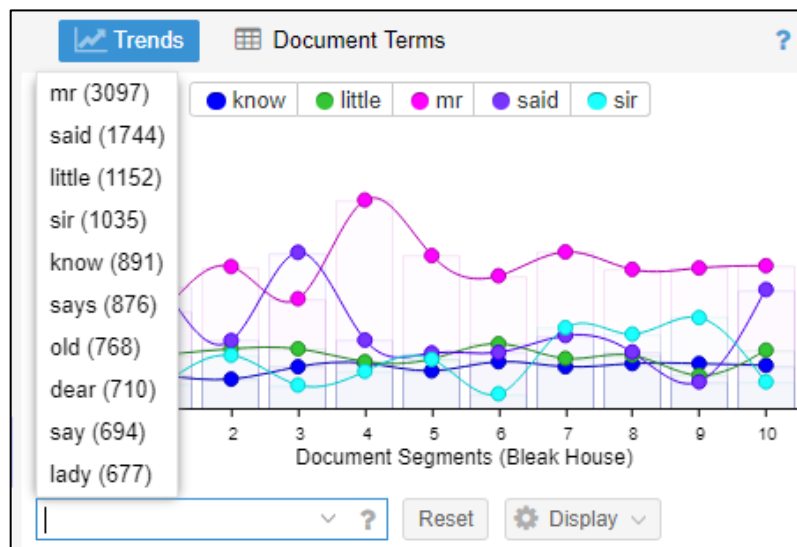


Figure 53 A search field in Voyant Tools

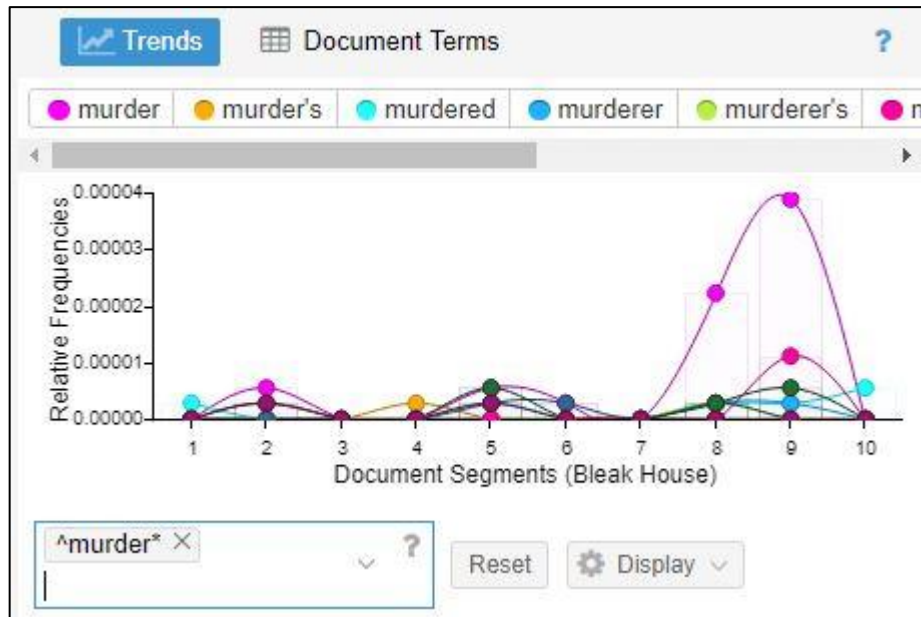


Figure 54 Searching using truncation in Voyant Tools

Voyant Tools can accept text that has been encoded in XML or TEI to denote certain textual phenomena. As an example, *Romeo and Juliet* by William Shakespeare has been encoded. Amongst other things, the encoding shows who the speaker is in each instance (Figure 55). By using XPath expressions, a user can specify which sections of the document(s) should be used for analysis when uploading the text. For example, the expression “//SPEECH[contains(SPEAKER,"ROMEO")]/LINE” will select the lines where Romeo is the speaker and so allow the user to analyse the words spoken by Romeo (Figure 56).

```

<SCENE>
<TITLE>SCENE I. Verona. A public place.</TITLE>
<STAGEDIR>
Enter SAMPSON and GREGORY, of the house of Capulet, armed with swords and bucklers
</STAGEDIR>
<SPEECH>
<SPEAKER>SAMPSON</SPEAKER>
<LINE>Gregory, o' my word, we'll not carry coals.</LINE>
</SPEECH>
<SPEECH>
<SPEAKER>GREGORY</SPEAKER>
<LINE>No, for then we should be colliers.</LINE>
</SPEECH>
<SPEECH>
<SPEAKER>SAMPSON</SPEAKER>
<LINE>I mean, an we be in choler, we'll draw.</LINE>
</SPEECH>
<SPEECH>
<SPEAKER>GREGORY</SPEAKER>
<LINE>
Ay, while you live, draw your neck out o' the collar.
</LINE>

```

Figure 55 *Romeo and Juliet* encoded to show speakers

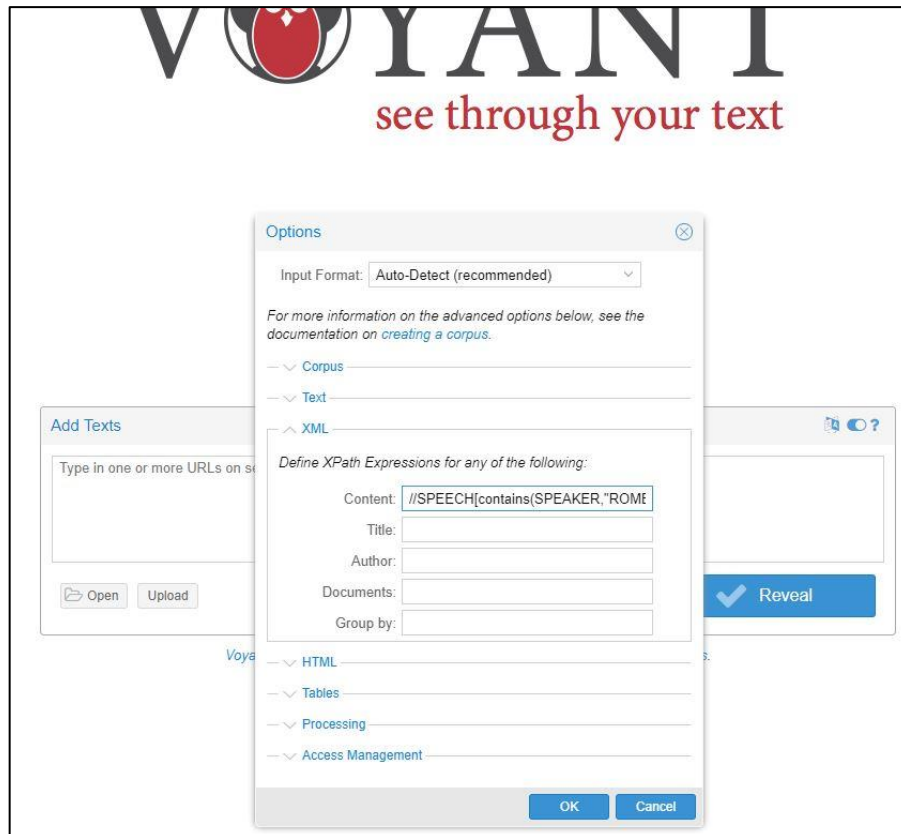


Figure 56 Selecting sections of text in Voyant Tools

This does mean that the user will need to be familiar with the encoding standard of XML, the query language XPath, as well as know how the text has been encoded in order to select sections of text.

TXM

TXM is based on the CQP search engine, which is software that processes queries written in CQL, a corpus query language. As such, a user can write complex and advanced search queries using CQL. However, TXM does not require a user to know CQL to use the software. TXM offers query assistants that help a user to construct queries. It is beyond the scope of this research to discuss CQL as a query language in depth, but the examples given to demonstrate how searches in TXM can be done will also illustrate some of the functionality of CQL.

A user can search using bibliographic, functional and morphological data. The following examples will illustrate how simple searching, as well as searching using metadata, can be done on TXM.

One can simply type a word to search for the occurrence of this word in the corpus. For example, in Figure 57 the user searched for the word *toutes*.

ref	Left context	Pivot	Right context
qgraal_cm, c	l'enfant, si le voit garni de	toutes	biautez si merveilleusement qu'i...
qgraal_cm, c	li palais raempliz de si bones o...	toutes	les espices terriennes i fussent ...
qgraal_cm, c	ja pueploiee. Si fu maintenant d...	toutes	les chambres de laiencz comme...
qgraal_cm, c	a plorer mout tendrement, et au...	toutes	les dames, et les damoiseles q...
qgraal_cm, c	chevaliers dou monde, et est es...	toutes	parz de rois et de reines, et dou ...
qgraal_cm, c	, et li plus desirrez a veoir de	toutes	genz, et li mielz amez qui onque...
qgraal_cm, c	le me dit pas qui me met en	toutes	les mesaises dou monde et en ...
qgraal_cm, c	en toutes les mesaises dou mo...	toutes	les poors ou onques gentil fam...
qgraal_cm, c	chastel si fist maintenant les po...	toutes	parz, et dist puis que Diex li avoit...
qgraal_cm, c	veu, et feroit autresi soef com se	toutes	les espices dou monde fussent ...
qgraal_cm, c	bien se je estoie ilec que l'en m...	toutes	les peines que l'en porroit en m...
qgraal_cm, c	l'ordre de chevalerie nez et espu...	toutes	ordures et de toz pechiez donc v...
qgraal_cm, c	ne l'abatent. A cel encontrer furent	toutes	lor lances brisiees si en a Gala...

Figure 57 Searching for a single word in TXM

Truncation and wildcard characters can be used to search for variations. A question mark indicates that the preceding character is optional, and an asterisk replaces multiple characters. For example, *rois?* searches for *roi* and *rois*; *.*tre* searches for all words that end with *-tre*. It is also possible to search for a sequence of words, for example, *"la" "nation" "française"* searches for the phrase *la nation française*.

One can use a query assistant or CQL commands to search for the properties of words that are available in the corpus. For example, to search for nouns in the VOEUX corpus one can use `[frpos="NOM"]`. An example of the TXM query assistant to create this search query is shown in Figure 58.

Figure 58 TXM query assistant

In Figure 59 a more complex query is constructed. The user is looking for the word *leurs*, followed by a noun, within a paragraph.

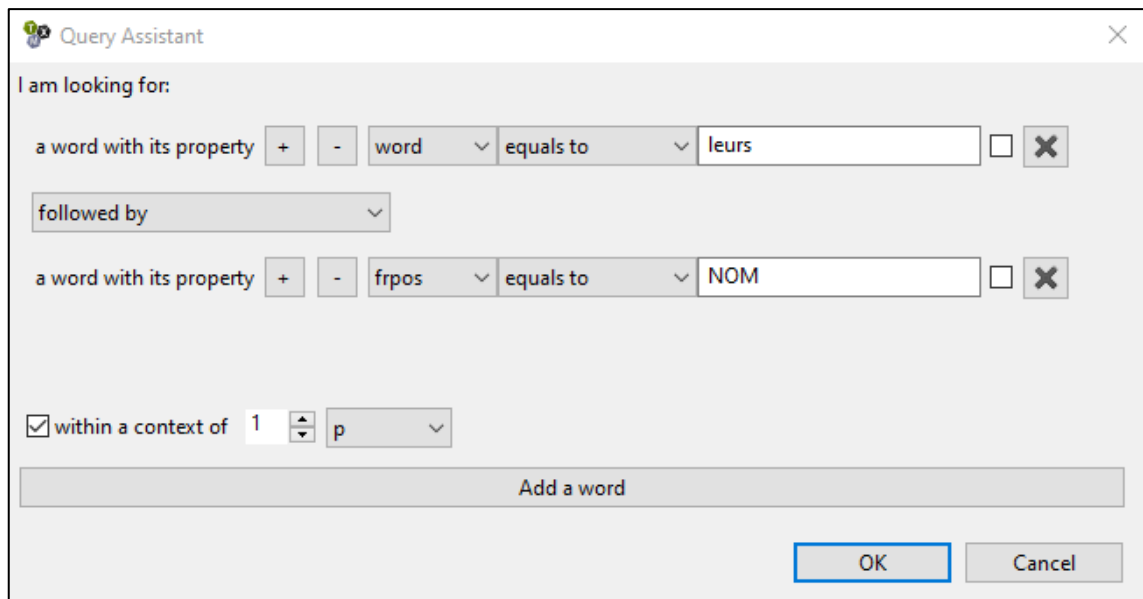


Figure 59 Using the query assistant in TXM to create a complex query

It is to be noted that though CQL is used throughout, the metadata that can be searched will be specific to each corpus. For example, searching for all nouns in the GRAAL corpus is done by searching [pos="NOMcom" | pos="NOMpro"].

Truncation and wildcard characters can be combined when searching in metadata, as is demonstrated in the following query that searches for a sequence of four words. The first word must be a verb but can be any type of verb. The second word must be "le". The third word must be a noun. The last word in the sequence must end in an "e".

```
[frpos="VER.*"]le[frpos="NOM"]*.e"
```

This query as executed on the VOEUX corpus to create a concordance is shown in Figure 60.

text_id	Left context	Pivot	Right context
0001	qui vont marquer 1960. Le franc nouveau	est le signe de	cette féconde solidité. Dans les domaines politique, s
0002	à réorganiser leur alliance en vue de mieux	défendre le monde libre	et d'agir en commun sur toute la terre. Aider à
0003	autant plus portés à la démagogie xénophobe,	remplit le monde de	tumultes bruyants. D'autre part, à l'intérieur de nous
0003	ans et sept mois, n'a pu	déterminer le pouvoir responsable	à changer de route. Assurément le caractère qu'ont pu
0004	qui a, dans le bon sens,	marqué le destin de	la France. Certes ne nous y ont manqué ni les épreuve
0005	, notamment, améliorer toutes les rémunérations ;	réaliser le reclassement de	deux cent mille chefs de famille rapatriés d'Algérie ; cr
0006	nationale bénéficier de l'avance solide que lui	permettent le plan de	développement économique et social actuellement er
0008	chacun de ses voisins et en travaillant à	bâtir le groupement économique	et peut-être un jour politique, des six Occidentaux. Le t
0010	enfin, que notre activité vigoureusement relancée,	dépasse le taux le	plus élevé qu'elle ait jamais atteint, que notre monnaie
0010	laisser à d'autres l'admirable mérite de	réussir le tour de	la lune, nous n'en avons pas moins à assumer dans
0011	l'hiver glacé de la libération, j'	accompagnais le général de	Gaulle qui rendait visite sous la neige aux villes de la t
0013	a pas déchu du rang où l'avait	placé le général de	Gaulle. Il y a un an, je vous disais encore
0016	pas célébrer la fin de l'année ou	apercevoir le début de	l'année nouvelle sans ressentir la misère du monde q
0017	Caire et à Ismaïlia, partout nous avons	rencontré le rayonnement de	la France. Puisse-t-elle, dans notre univers tourmenté
0017	, dans notre univers tourmenté et violent,	rester le phare de	la liberté et du rapprochement entre les hommes. Je l
0017	qui peut alléger l'effort, améliorer ou	préserver le cadre de	vie et faciliter les tâches quotidiennes, notamment cell
0018	avec ceux que vous aimez, ce qui	est le bonheur le	plus important. Je souhaite que vous gardiez la santé
0020	sens venu du fond des âges et qui	est le certificat de	naissance des nations, - le sens de l'unité l'a
0023	bon, à la condition de ne jamais	confondre le désir que	nous en avons et la réalité d'aujourd'hui. Le drame pol
0028	souhaite qu'elle gagne les enjeux que lui	propose le monde moderne	. Qu'elle sache s'unir quand il le faut. Les

Figure 60 Query executed in TXM

TXM allows a user to create a subset of data using the bibliographic and structural data. For example, a selection based on a specific text structure can be made, such as <q> for everything that has been encoded as citations (see Figure 61). A selection based on bibliographic data can be made, such as the author of a text. In Figure 62 all texts by a certain author (Chirac) is selected.

Create partition ✕

Name:

Simple **Assisted** Advanced

Structure: Property

Select the values to assign:

citation

written

Title:

Figure 61 All direct speech is selected in TXM

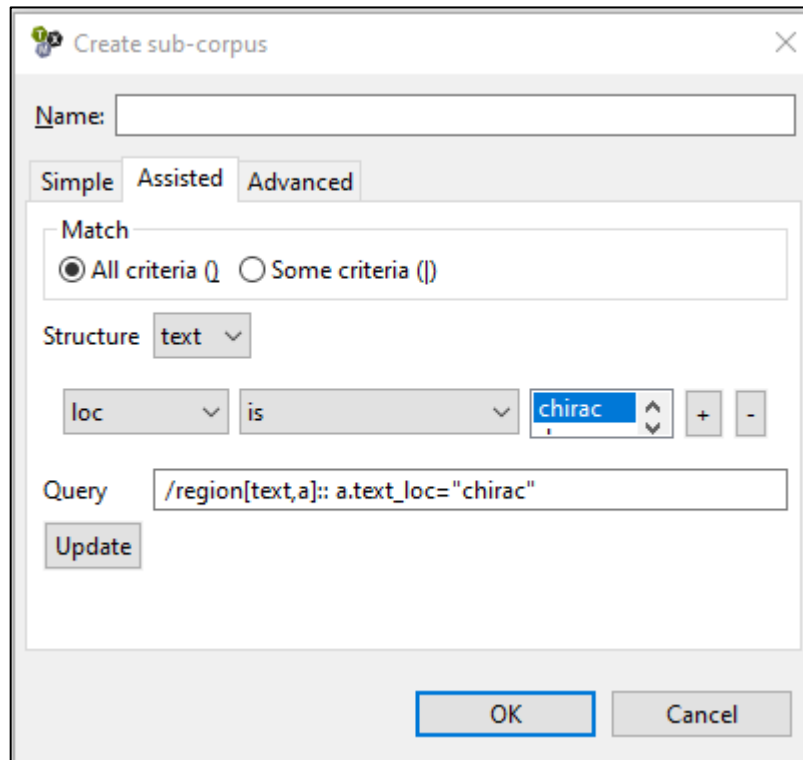


Figure 62 All texts by a certain author is selected in TDM

From the figures it is clear the TDM provides three modes for a user to make a selection of data, namely, a simple, assisted and advanced mode. This allows the user to use the simple interface, a combination of graphical user interface and CQL commands, and pure CQL commands. Once a selection of data from the main corpus has been made, some commands can be executed on the selection.

However, making a selection can be cumbersome and complex and the functionality for a selection can be limited. More regarding the complexity of this tool will be said in section 2.7.6.

BNCweb (CQP-edition)

As was mentioned previously, there are two modes that can be used to search on the BNCweb, namely, the simple query mode and the CQP mode. (The simple query mode offers two options, namely one that is case sensitive and one where case is ignored.) Due to the complexity of the CQP language, a simplified, intuitive query language was designed, allowing simple queries to be executed, and allowing even novices to easily and intuitively create queries of considerable complexity (Hoffmann & Evert, 2006). However, the CQP mode is still available for advanced users due to the powerful queries one can create.

In both modes there are various advanced search options. For example, one can use wildcard characters to search for patterns or variations of words, search for a word form with a certain part-of-speech tag, search for lemmas, word sequences and within XML tags (BNCweb (CQP-edition), n.d.; Evert, 2005).

Some examples will be given to demonstrate some of the search features of these two modes.

1) *y_AJ0 s?ng

The first query is written in the simplified query language and searches for any adjective that ends in a y, followed by the token *s?ng*, where the question mark can be replaced by any character (Figure 63). The asterisk is used to match zero to several consecutive characters and the question mark is used for a single arbitrary character. Part-of-speech tags are written in capital letters and preceded by an underscore.

2) <quote> (*)* [good] (*)* </quote>

The second query is written in the simplified query language and searches for the lemma *good* appearing in quoted text, where there are zero or more words or tokens before *good* and zero or more words or tokens after *good*.

3) first:[pos = "NN.*"] [pos = "VB.*"] [pos = "NN.*" & word = first.word]

The third query (an example from Hoffmann and Evert (2006)) is written in CQP and searches for a noun, followed by a verb, followed by a noun that is the same word as the first noun, retrieving instances such as *rules are rules* or *bygones be bygones*.

Main menu	BNCweb (CQP-Edition)	
Query options	Standard Query	
Standard query		
Written restrictions	*y_AJ0 s?ng	
Spoken restrictions		
User-specific functions		
User settings		
Query history		
Saved queries		
Categorized queries		
Make/edit subcorpora		
Upload external data file		
Additional functions		
Browse a text		
	Query mode: <input type="text" value="Simple query (ignore case)"/>	Simple Query Syntax Audio data search
	Number of hits per page: <input type="text" value="50"/>	
	Restriction: <input type="text" value="None (search whole corpus)"/>	
	Extended audio controls: <input type="text" value="Do not show"/>	
	<input type="button" value="Start Query"/> <input type="button" value="Reset Query"/>	
	BNCweb (CQP-edition) © 1996-2018 You are logged in as user "Iball"	

Figure 63 Standard query in BNCweb

BYU Corpora

BYU Corpora offers a search field, where a user enters a simple word or phrase to search for. A user can include some of the advanced search features available on the platform. A user can use wildcard characters to search for substrings, search for lemmas or a specific part-of-speech, or search for synonyms.

The example query “*y_j* s?ng” will search for all adjectives ending on a y followed by a word s?ng where the question mark can be any character.

A user can type the part-of-speech tag or use the dropdown menu to select a part-of-speech (Figure 64).

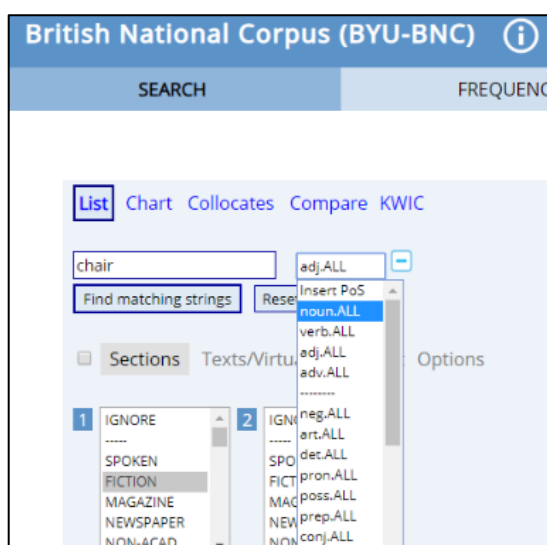


Figure 64 List of part-of-speech tags in corpus.byu.edu

Furthermore, as a user starts a search, there are various search options to select from, listed above the search field (depending on the corpus that has been selected). In the BNC, a user can choose to list the results, to see a chart, to search for collocates, to compare words or see patterns of words in KWIC format (in Figure 64 “List” has been selected). In the iWeb, a user can also choose to list the results, to search for collocates, to compare words or see patterns of words in KWIC format. A user cannot see the results in a chart or compare words, but there is the option to see more information about a specific word or browse according to various options. The results retrieved by selecting different options will be discussed in section 2.7.5.

Although some corpora included by corpus.byu.edu do include structural encoding, such as the BNC, it is not possible to search in these metadata.

2.7.4. Filtering

Filters allow a user to apply certain constraints on a dataset (Wilson, 2011: 150).

Applying filters on a dataset creates a subset of the larger dataset. Filters are found in many applications. For example, in Google a user can filter the main results to only see images, or videos. In database search services, such as EbscoHost, filters such as date or publication type can be applied to create a subset of the main results.

In this study, filtering will focus on the use of bibliographic data to create a subset, for example, creating a subset of texts that were published in a specific year or are of certain genre.

Google Books Ngram Viewer

Due to the lack of available bibliographic metadata there is little filtering that can be done in the Google Books Ngram Viewer. However, some filtering can be done on date, language and to a limited extent genre (refer to the English Fiction corpus). A user filters the language by choosing a specific corpus. In Figure 65, British English has been selected.

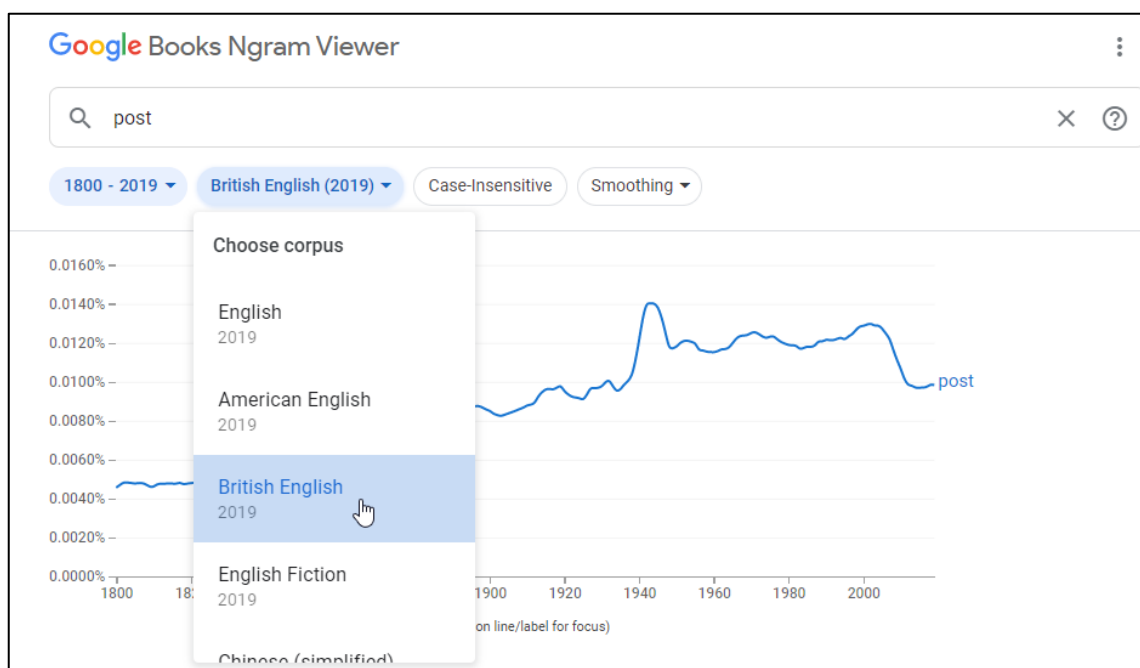


Figure 65 Filtering according to language in Google Books Ngram Viewer

HathiTrust+Bookworm

As was seen in section 2.6.2, the HathiTrust+Bookworm makes use of metadata fields to allow a user to filter according to specific fields, such as class.

The HathiTrust+Bookworm allows a user to filter according to the following fields:

- Language
- Publication country
- Publication state
- Subclass
- Narrow class
- Class
- Resource type
- Target audience
- Scanner
- First author birth
- First author name
- Contributing library
- Literary form
- Cataloguing source
- First author death
- First place
- First publisher
- Is Gov
- Subject places

Unfortunately, the HathiTrust+Bookworm itself does not provide documentation to explain the fields that are available. It might be obvious for someone familiar with cataloguing and records, but for an interested lay person, or a researcher in humanities not familiar with cataloguing practices, it could be a barrier. For example, what the field “Is Gov” is might not be clear to everyone.

In addition to filtering according to these fields, the user can also adjust the dates to search by and indicate if the search should be case sensitive or not.

In Figure 66 the frequency of the term *umbrella* from 1802 to 1901 is shown, where the following filters have been applied: the language is English, publication country is United Kingdom, the class is Language and Literature, subclass is English Literature, the resource type is book and the literary form is fiction.

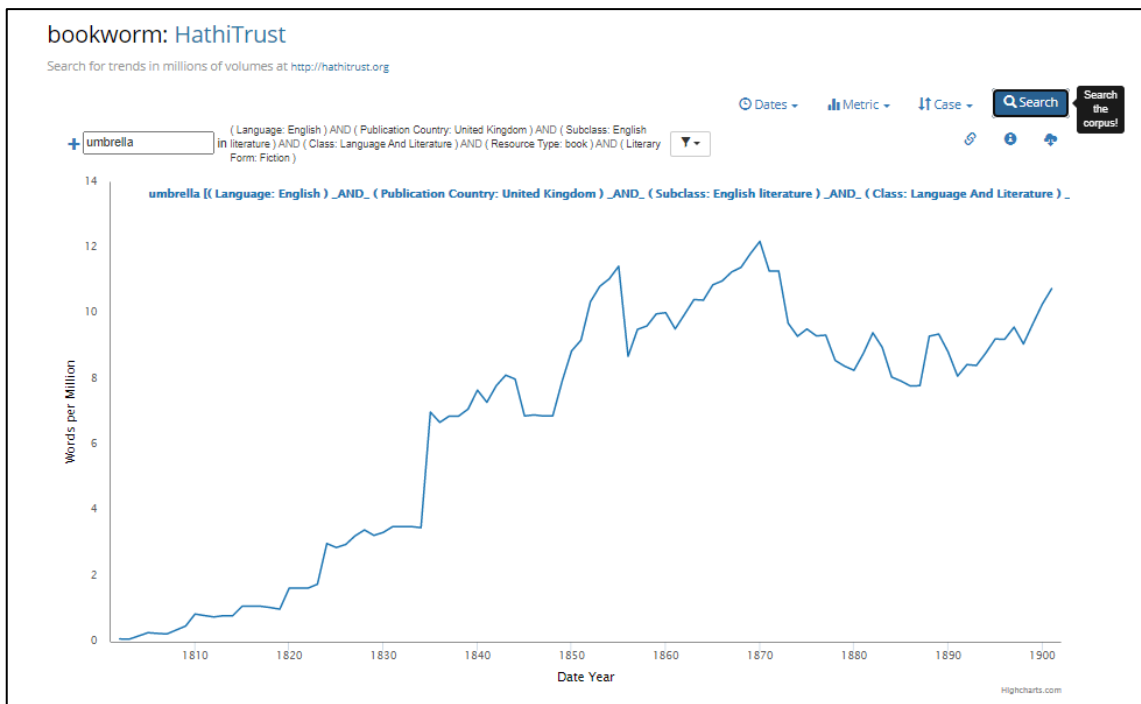


Figure 66 Filtering in HathiTrust+Bookworm

Furthermore, as was explained in the previous section, as there is some concern over the quality of the metadata, there is some doubt about the reliability of the results obtained when applying filters.

Perseus Project

There are no specific filtering options in the Perseus Project, for example, filtering according to bibliographic data such as genre or date. However, all the texts are divided into collections and a user can browse to a specific collection.

Voyant Tools

The Voyant Tools do not specifically engage with bibliographic data. However, as a user adds texts (s)he could make some selections through XPath expressions if the corpus is encoded in such a way as to allow it. This could allow some kind of filtering, but is not straightforward in the tool itself.

TXM

TXM does not offer filtering through dropdown menus as is the case in Google Books Ngram Viewer or the HathiTrust+Bookworm. Filtering is achieved through the search options that were discussed in the previous section.

BNCweb (CQP-edition)

The BNCweb offers various filtering options. The first option is to search only in the written section or only in the spoken section of the corpus (Figure 67).

Main menu	BNCweb (CQP-Edition)
Query options	Standard Query
Standard query	
Written restrictions	
Spoken restrictions	
User-specific functions	
User settings	
Query history	
Saved queries	
Categorized queries	
Make/edit subcorpora	
Upload external data file	
Additional functions	
Browse a text	

Query mode:	Simple query (ignore case) ▾	Simple Query Syntax Audio data search
Number of hits per page:	50 ▾	
Restriction:	None (search whole corpus) ▾	
Extended audio controls:	None (search whole corpus)	
	Written Texts	
	Spoken Texts	

BNCweb (CQP-edition) © 1996-2018 You are logged in as user "Iball"

Figure 67 Filtering according to written or spoken texts in the BNCweb

Furthermore, there are specific filtering options for both the written and spoken texts in the corpus. The filtering options for the written section of the corpus are shown in Figure 68. These are typical bibliographic data, for example, publication date, and data that are relevant for linguists, such as the perceived level of difficulty or where in the text the sample was taken from. One can also filter according to the genres used in the BNC (Figure 69).

Publication Date:	Medium of Text:	Text Sample:
<input type="checkbox"/> 1960-1974 <input type="checkbox"/> 1975-1984 <input type="checkbox"/> 1985-1993	<input type="checkbox"/> Book <input type="checkbox"/> Periodical <input type="checkbox"/> Miscellaneous: published <input type="checkbox"/> Miscellaneous: unpublished <input type="checkbox"/> To-be-spoken	<input type="checkbox"/> Whole text <input type="checkbox"/> Beginning sample <input type="checkbox"/> Middle sample <input type="checkbox"/> End sample <input type="checkbox"/> Composite
Domain:		Derived text type:
<input type="checkbox"/> Imaginative prose <input type="checkbox"/> Informative: Natural and pure sciences <input type="checkbox"/> Informative: Applied science <input type="checkbox"/> Informative: Social science <input type="checkbox"/> Informative: World affairs	<input type="checkbox"/> Informative: Commerce and finance <input type="checkbox"/> Informative: Arts <input type="checkbox"/> Informative: Belief and thought <input type="checkbox"/> Informative: Leisure	<input type="checkbox"/> Academic prose <input type="checkbox"/> Fiction and verse <input type="checkbox"/> Non-academic prose and biography <input type="checkbox"/> Newspapers <input type="checkbox"/> Other published written material <input type="checkbox"/> Unpublished written material
Estimated Circulation Size:	Perceived Level of Difficulty:	Domicile of Author:
<input type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High	<input type="checkbox"/> Low <input type="checkbox"/> Medium <input type="checkbox"/> High	<input type="checkbox"/> UK and Ireland <input type="checkbox"/> Commonwealth <input type="checkbox"/> Continental Europe <input type="checkbox"/> USA <input type="checkbox"/> Elsewhere
Age of Author:	Sex of Author:	Type of Author:
<input type="checkbox"/> 0-14 <input type="checkbox"/> 15-24 <input type="checkbox"/> 25-34 <input type="checkbox"/> 35-44 <input type="checkbox"/> 45-59 <input type="checkbox"/> 60+	<input type="checkbox"/> Male <input type="checkbox"/> Female <input type="checkbox"/> Mixed	<input type="checkbox"/> Corporate <input type="checkbox"/> Multiple <input type="checkbox"/> Sole
Target Audience Age:	Target Audience Sex:	
<input type="checkbox"/> Child <input type="checkbox"/> Teenager <input type="checkbox"/> Adult <input type="checkbox"/> Any	<input type="checkbox"/> Male <input type="checkbox"/> Female <input type="checkbox"/> Mixed	

Figure 68 Filtering options on the BNCweb (CQP-edition)

Genre (description of codes):		
<input type="checkbox"/> W:ac:humanities_arts <input type="checkbox"/> W:ac:medicine <input type="checkbox"/> W:ac:nat_science <input type="checkbox"/> W:ac:polit_law_edu <input type="checkbox"/> W:ac:soc_science <input type="checkbox"/> W:ac:tech_engin <input type="checkbox"/> W:admin <input type="checkbox"/> W:advert <input type="checkbox"/> W:biography <input type="checkbox"/> W:commerce <input type="checkbox"/> W:email <input type="checkbox"/> W:essay:school <input type="checkbox"/> W:essay:univ <input type="checkbox"/> W:fict:drama <input type="checkbox"/> W:fict:poetry <input type="checkbox"/> W:fict:prose	<input type="checkbox"/> W:hansard <input type="checkbox"/> W:institut_doc <input type="checkbox"/> W:instructional <input type="checkbox"/> W:letters:personal <input type="checkbox"/> W:letters:prof <input type="checkbox"/> W:misc <input type="checkbox"/> W:news_script <input type="checkbox"/> W:newsp:brdsh_t:arts <input type="checkbox"/> W:newsp:brdsh_t:commerce <input type="checkbox"/> W:newsp:brdsh_t:editorial <input type="checkbox"/> W:newsp:brdsh_t:misc <input type="checkbox"/> W:newsp:brdsh_t:report <input type="checkbox"/> W:newsp:brdsh_t:science <input type="checkbox"/> W:newsp:brdsh_t:social <input type="checkbox"/> W:newsp:brdsh_t:sports	<input type="checkbox"/> W:newsp:other:arts <input type="checkbox"/> W:newsp:other:commerce <input type="checkbox"/> W:newsp:other:report <input type="checkbox"/> W:newsp:other:science <input type="checkbox"/> W:newsp:other:social <input type="checkbox"/> W:newsp:other:sports <input type="checkbox"/> W:newsp:tabloid <input type="checkbox"/> W:non_ac:humanities_arts <input type="checkbox"/> W:non_ac:medicine <input type="checkbox"/> W:non_ac:nat_science <input type="checkbox"/> W:non_ac:polit_law_edu <input type="checkbox"/> W:non_ac:soc_science <input type="checkbox"/> W:non_ac:tech_engin <input type="checkbox"/> W:pop_lore <input type="checkbox"/> W:religion

Figure 69 Genres of the BNC (written section)

A combination of these filters can be used to search in a very specific subset of the corpus. For example, one can search in stories (W:fict:prose in genre) that were written for children (target audience age) by women (sex of author) between 1960-1974 (publication date).

BYU Corpora

Some filtering is available through this interface. Firstly, a user can create virtual corpora (subcorpora). Virtual corpora can be created based on keywords in the text or information about the text. There is very limited information available about texts from which a selection can be made. Working with the BNC, a user can create a virtual corpus on the title, source, keywords, genre, individual texts or words in the texts. There are more bibliographic data in the corpus itself (as is evident from the discussion on the BNCweb), but it cannot be utilised through this tool. In iWeb, a user can create virtual corpora according to a topic or a web domain. A user can then search in these virtual corpora, which is a filtered subset of the main corpus.

Apart from virtual corpora, a user can search according to sections in the corpora that have such metadata. For example, the genres in the BNC are used to divide the results into sections. Figure 70 shows the results for the search “a most ADJECTIVE NOUN” according to sections (e.g. spoken, fiction, magazine). A user can then select to see the examples of an item in a specific section (Figure 71).

British National Corpus (BYU-BNC)										
SEARCH			FREQUENCY		CONTEXT					ACCOUNT
SEE CONTEXT: CLICK ON WORD (ALL SECTIONS), NUMBER (ONE SECTION), OR [CONTEXT] (SELECT) [HELP...]										
CONTEXT			ALL	SPOKEN	FICTION	MAGAZINE	NEWSPAPER	NON-ACAD	ACADEMIC	MISC
1	<input type="checkbox"/>	A MOST ENJOYABLE EVENING	4					3		1
2	<input type="checkbox"/>	A MOST IMPORTANT FACTOR	4				1			3
3	<input type="checkbox"/>	A MOST UNUSUAL THING	4						1	
4	<input type="checkbox"/>	A MOST EXTRAORDINARY THING	3	2	1					
5	<input type="checkbox"/>	A MOST INTERESTING DEBATE	3	1						2
6	<input type="checkbox"/>	A MOST INTERESTING TALK	3					2		1
7	<input type="checkbox"/>	A MOST PECULIAR WAY	3		1			1		1
8	<input type="checkbox"/>	A MOST REWARDING EXPERIENCE	3							3
9	<input type="checkbox"/>	A MOST USEFUL ADDITION	3			1			1	1
10	<input type="checkbox"/>	A MOST ATTRACTIVE POSITION	2							2

Figure 70 Searching in corpus.byu.edu according to sections

British National Corpus (BYU-BNC)										
SEARCH			FREQUENCY		CONTEXT					ACCOUNT
SECTION: NON-ACAD (3) (SHUFFLE)										
CLICK FOR MORE CONTEXT <input type="checkbox"/> [?] SAVE LIST CHOOSE LIST CREATE NEW LIST <input type="text"/> [?]										
SHOW DUPLICATES										
1	GXH	W_non_ac_soc_science	A	B	C	13th February 1993. A poor response, only 43 children participating. Nevertheless a most enjoyable evening. Clown entertainment. # 3 # Quiz Night Saturday 20th March				
2	HD0	W_non_ac_soc_science	A	B	C	adverse weather conditions. Entertainment provided by City By Pass and child instrumentalists. A most enjoyable evening. # 3. # Pre-school Children's Party, Thursday, 2				
3	HD1	W_non_ac_soc_science	A	B	C	13th February 1993. A poor response, only 43 children participating. Nevertheless a most enjoyable evening. Clown entertainment. # 3 # Quiz Night Saturday 20th March				

Figure 71 Viewing results according to section

2.7.5. Search results

In this section, the results that the user retrieves after a search will be considered. If the user searches for a word or phrase in the collection, in what way are the results presented to the user? Are results available as a list of words in context (KWIC)? Are they displayed in a graph to show frequency over time? Furthermore, it should be considered if the data are removed from context, or if a user can link to see the context of an example.

Google Books Ngram Viewer

The Google Books Ngram Viewer searches for n-grams and retrieves the frequency of their usage in the corpus over a period of time. This is then displayed in a graph. Below the graph are predetermined searches for Google Books.

The example in Figure 72 shows the frequency of the terms Cupid and Psyche over the period from 1800 to 2000. Predetermined searches for Google Books are displayed at the bottom of the page. For example, the first link leads to a Google Books page where the search is for Cupid and limited to books published between 1800 and 1806 (Figure 73).

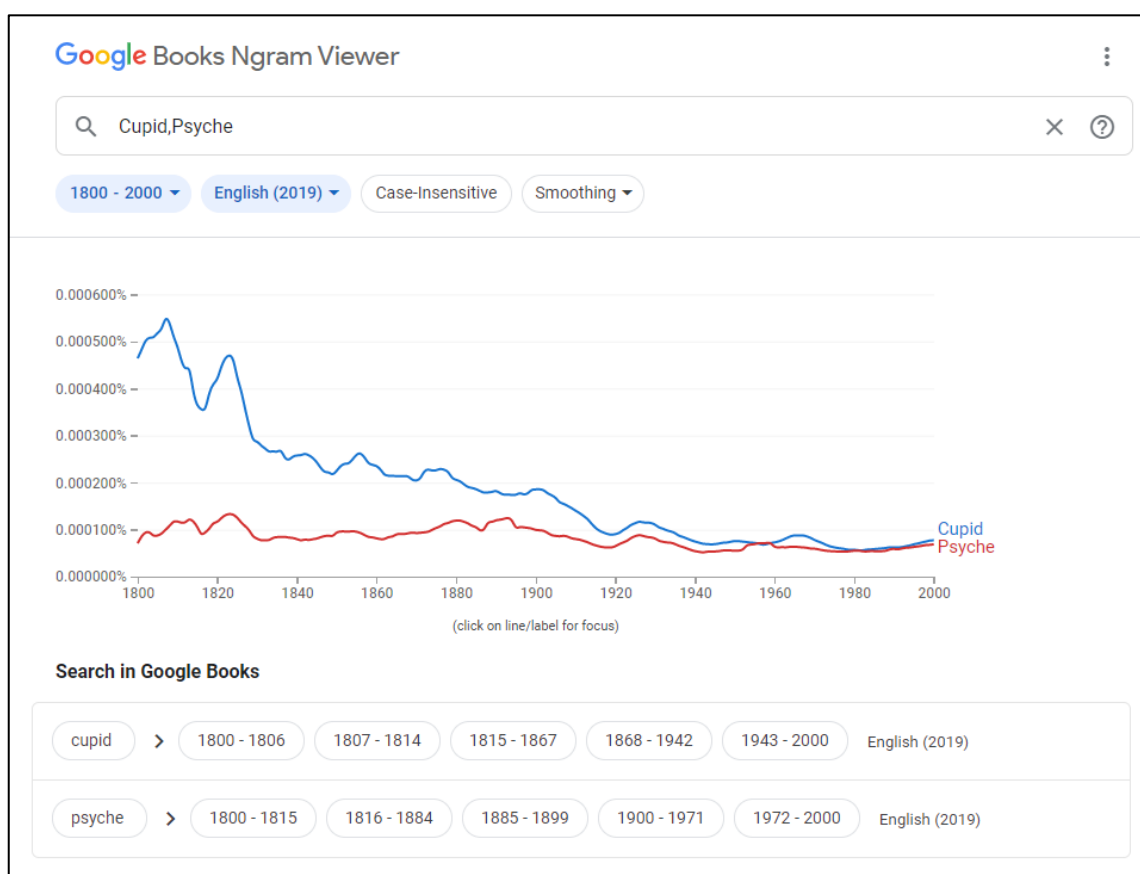


Figure 72 Searching for Cupid and Psyche in Google Books Ngram Viewer

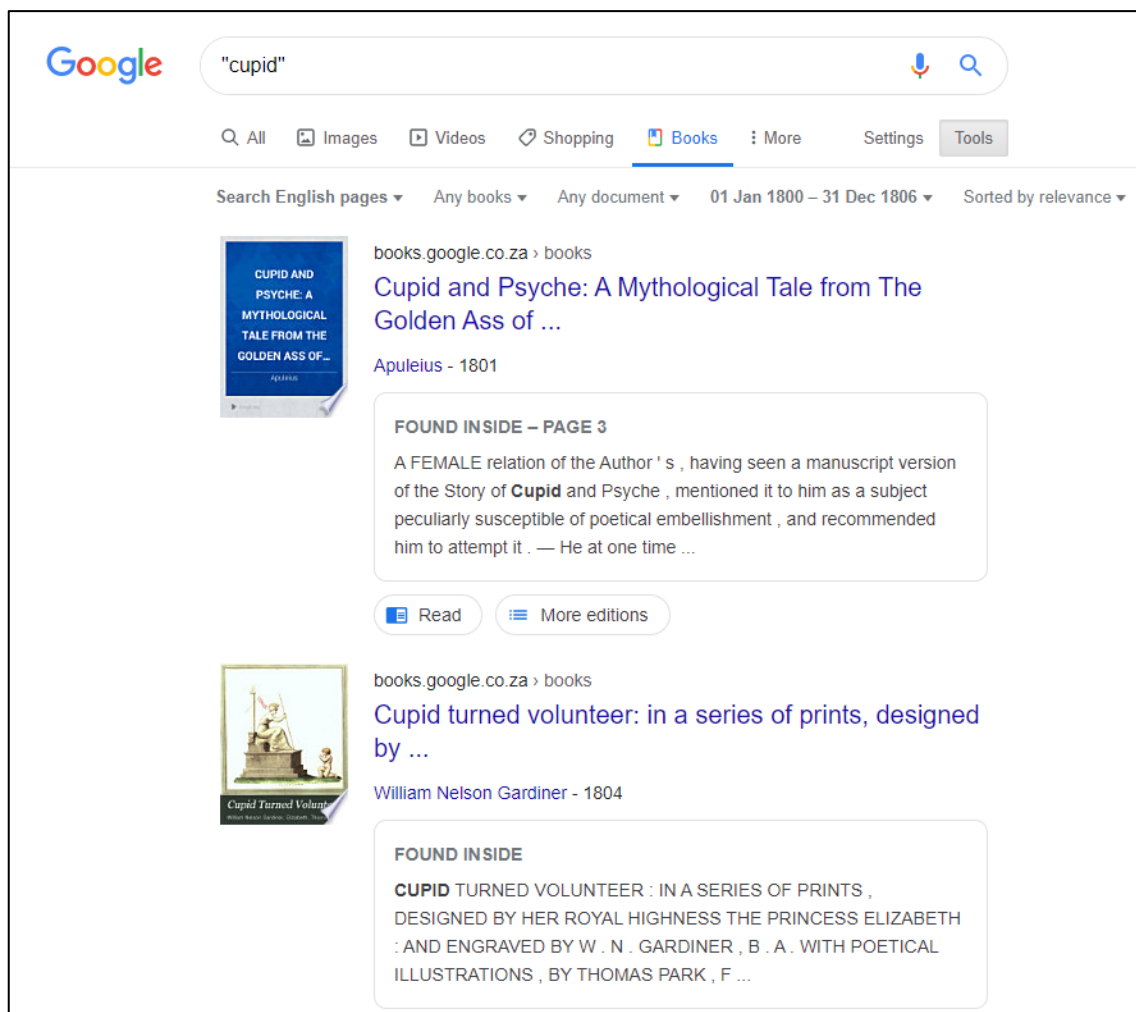


Figure 73 A search done in Google Books

It should be noted that the corpora used in the Google Books Ngram Viewer were created on specific dates and as such have dates attached to them (e.g. 2009 or 2012 version). However, Google Books could be growing and the search results in Google Books do not necessarily reflect the data in the Ngram Viewer. Furthermore, the developers of the Ngram Viewer specifically state that some books from Google Books were filtered out when creating the corpora so the data in the corpora and Google Books are not the same (Culturomics, 2017).

The user cannot see or link to the data that are used to create the graph from the Ngram Viewer. It is possible to download the raw data and do experiments; however, the n-grams are not linked to the source from which they were taken due to copyright reasons.

HathiTrust+Bookworm

Similar to the Google Books Ngram Viewer, the HathiTrust+Bookworm searches for words and retrieves the frequency of their usage in the corpus over a period of time. This is then displayed in a graph.

However, as opposed to Google Books Ngram Viewer where the corpus is not linked to data of the digital library from which the corpus was created, HathiTrust+Bookworm does link to the data that a user searches in. One can click on a point in the graph to see the top search results in the library for the term in that year (see Figure 74 and Figure 75). The search results link to the corresponding items in the HathiTrust Digital Library (see Figure 76).

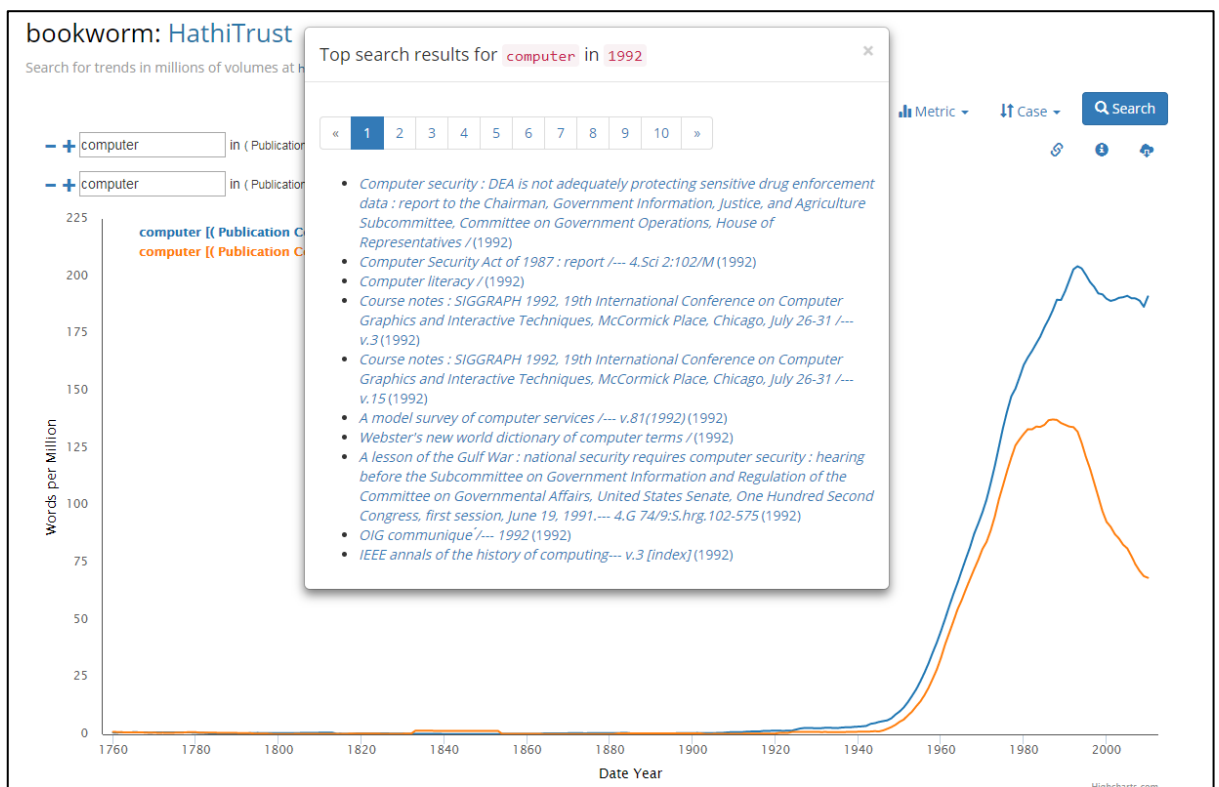


Figure 74 The top search results for a term in a specific year



Figure 75 Enlargement of Figure 74

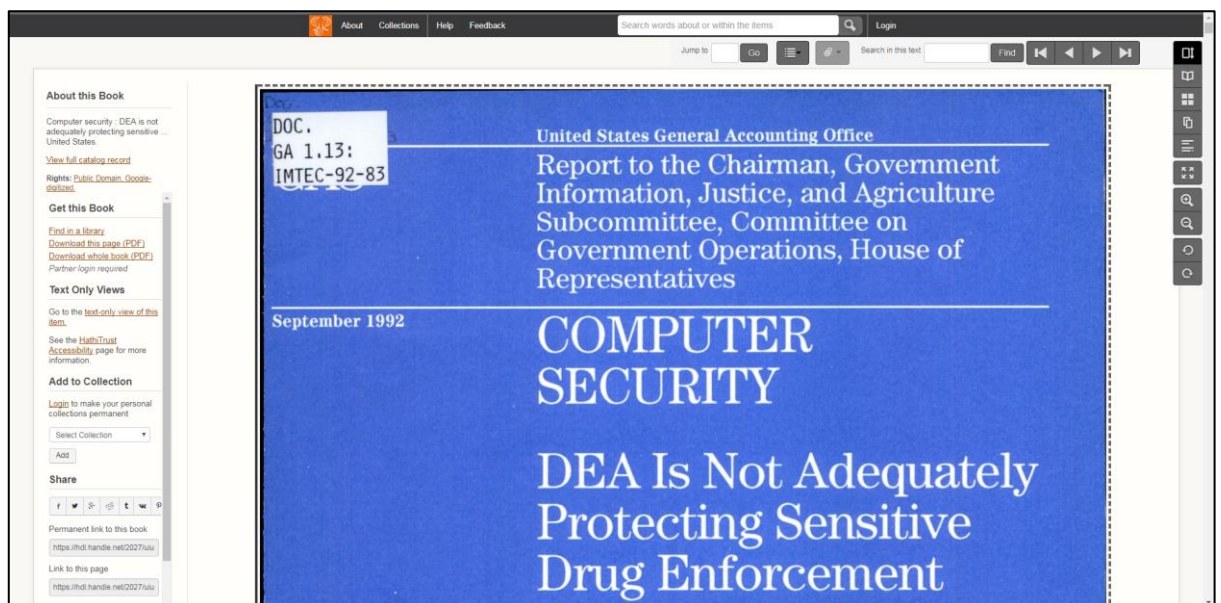


Figure 76 An item from the search results in the digital library

HathiTrust+Bookworm protects items that are still in copyright. If a user follows a link in the top search results that leads to an item that is still in copyright, a message will be shown to indicate that the item is not available (see Figure 77).

Home About ▾ Collections Help Feedback Search HathiTrust [LOG IN](#)

Options

This item is **not available online** (🔒 Limited - search only) due to copyright restrictions. [Learn More »](#)

Search in this text

You can try to [find this item in a library](#) or [search in this text](#) to find the frequency and page number of specific words and phrases. This can be especially useful to help you decide if the book is worth buying, checking out from a library, etc.

Figure 77 An item in the HathiTrust Digital Library is not available due to copyright restrictions

Perseus Project

A search in the Perseus Project can return a list of texts (Figure 78) or information about a word (Figure 79). A user can either search for more information about a word in general, or as previously explained, a user can simply click on a word in a text to retrieve information about the word in context.

Search Results

("Agamemnon", "Hom. Od. 9.1", "denarius")
All Search Options [view abbreviations]

Home Collections/Texts Perseus Catalog Research Grants Open Source About Help

Showing 1 - 10 of 985 document results in Latin. 1 2 3 4 5 6 ... ▶ ⌂

Aristotle, *Economics* [More\(2\)](#)
(Greek) (English)
book 3, section 2: ... corporis esca, ad quam anime semen consumitur quid si pro **suis** liberis matre atque nutrice nonne omne studium est faciendum?

Callimachus, *Hymns and Epigrams* [More\(4\)](#)
(Greek)
text intro, section 5: ... principe hymni quinti editio reddidit, quam a. 1489 primum Miscellaneis **suis** inseruit. Cf. Nigra p. 42. qui codex epigramma

C. Julius Caesar, *De bello Gallico* [More\(71\)](#)
(Latin) (English)
book 1, chapter 1: ... praecedunt, quod fere cotidianis proeliis cum Germanis contendunt, cum aut **suis** finibus eos prohibent aut ipsi in eorum finibus bellum

C. Valerius Catullus, *Carmina* [More\(5\)](#)
(Latin) (English, ed. Leonard C. Smithers) (English, ed. Sir Richard Francis Burton)
poem 3: ... passer, deliciae meae puellae, quem plus illa oculis **suis** amabat; nam mellitus erat, suamque norat ipsa

M. Tullius Cicero, *Letters to and from Brutus* [More\(5\)](#)
(Latin) (English, ed. Evelyn Shuckburgh, Evelyn S. Shuckburgh)
book 1, letter 1: ... visus est suspicari nec sine magno quidem dolore aliquid a **suis** vel per suos potius iniquos ad te esse delatum

Refine This Search [hide](#)

Language: Latin ▾

Required words: Expand

Required phrase:

Allowed words: Expand

Excluded words: Expand

(This searches within the currently selected documents. To search within all documents, use the form below.)

Relevant Works (1) [show](#)

All Matching Documents (985) [show](#)

Matching Lemmas (4) [hide](#)

- sus: "a swine, hog, pig, boar, sow" (entry in Lewis & Short Elem. Lewis)
- suo: "to sew, stitch, sew up, sew together" (entry in Lewis & Short Elem. Lewis)
- suis: "of oneself, belonging to oneself, his own, her own, his, her, its, their" (entry in Lewis & Short Elem. Lewis)
- Sue: "a town in Assyria" (entry in Lewis & Short)

Figure 78 Results of a search in Perseus Project

Figure 79 More information about a specific word in the Perseus Project

Another tool in the Perseus Project is the Vocabulary tool that allows a user to explore the vocabulary of the non-English texts in the collection. A user makes a selection of texts to use in the analysis and can then get a view of the vocabulary in the selection. In Figure 80 five texts have been selected. The most frequently used words are displayed in table format.

Figure 80 Vocabulary tool in Perseus Project

There are two further points worth noting. It is not possible to view trends of words over time. The full texts of the items are accessible.

Voyant Tools

Each tool in Voyant Tools will have its own type of output. It is beyond the scope of this research to discuss each tool in depth. Some of the tools pertinent to this study will be discussed.

Figure 81 shows the Trends tool, the Contexts tool and the Reader tool. The Trends tool shows the frequency of terms across a text. It is not a timeline as in the case of Google Books Ngram Viewer or HathiTrust+Bookworm, but shows the occurrence of the words through the document. If more than one document is analysed, it will show the frequency of terms in each document. The Contexts tool shows the search terms in a KWIC format, with context to the left and to the right. The Reader tool shows the full-text of the document(s) being analysed.

The tools are linked. A user can click on a term in the Reader to change the output in the Trends and Contexts tools. For example, in Figure 81, the user has clicked in the word *air* in the Reader tool and the other tools are now considering the word *air* to be the search term.

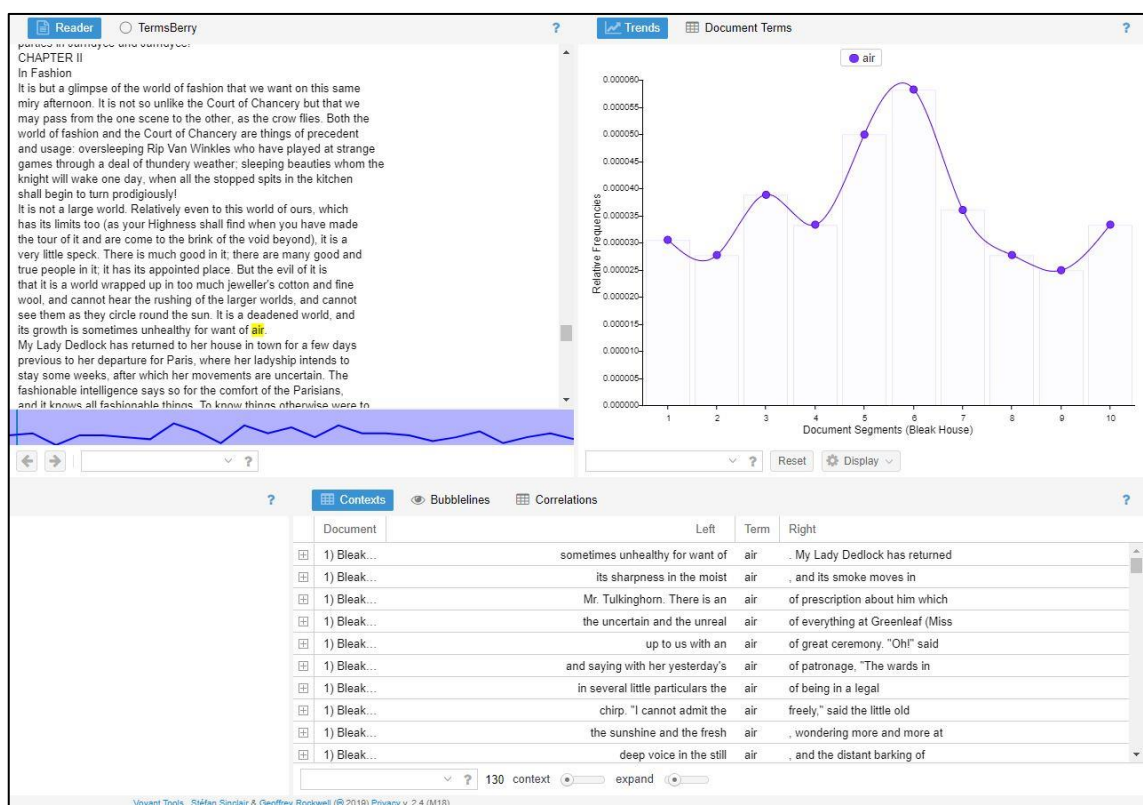


Figure 81 Search results in Voyant Tools

It is also worth noting that there are very interesting visualisations that can be used. For example, the Bubblelines tool, shows the frequency of words in a text as bubbles along a line (Figure 82).

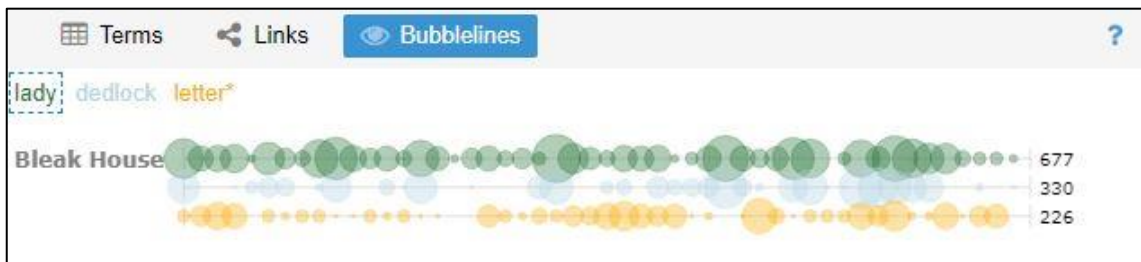


Figure 82 Bubblelines in Voyant Tools

TXM

Due to the various text analysis options that TXM offer, there are various results that a user could retrieve. The most pertinent to this study will be discussed here, namely, the concordance that can be created.

A concordance for the query “*leurs*”[fpos=NOM] (the word *leurs* (*their*) followed by a noun) in the context of one paragraph on the VOEUX corpus is shown in Figure 83. The results are displayed as keywords in context. There are various display settings. In this example only the words are displayed, but more information can be displayed, for example the lemma and the part-of-speech category. A user can also change how the results are sorted. A user can also double-click on a word in the results and the original text with the selected word highlighted, is opened in the top panel. In this way, the words in the KWIC list can be linked back to their original context.

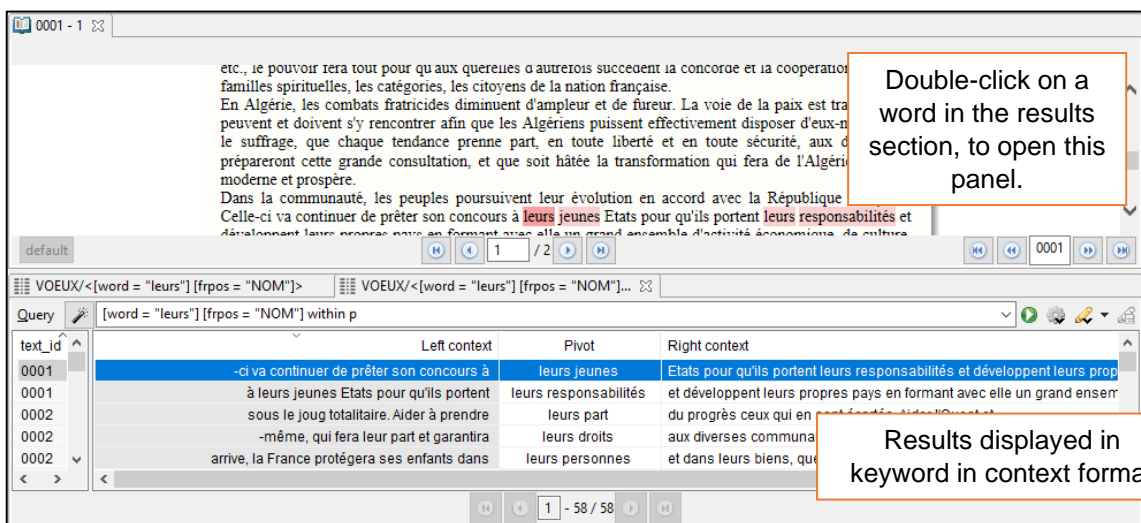


Figure 83 Concordance in TXM

A user can page through the text of the corpus as is shown in Figure 84. If the corpus allows it, as is the case with the GRAAL corpus, an image of the original manuscript can be displayed to the right of the text.

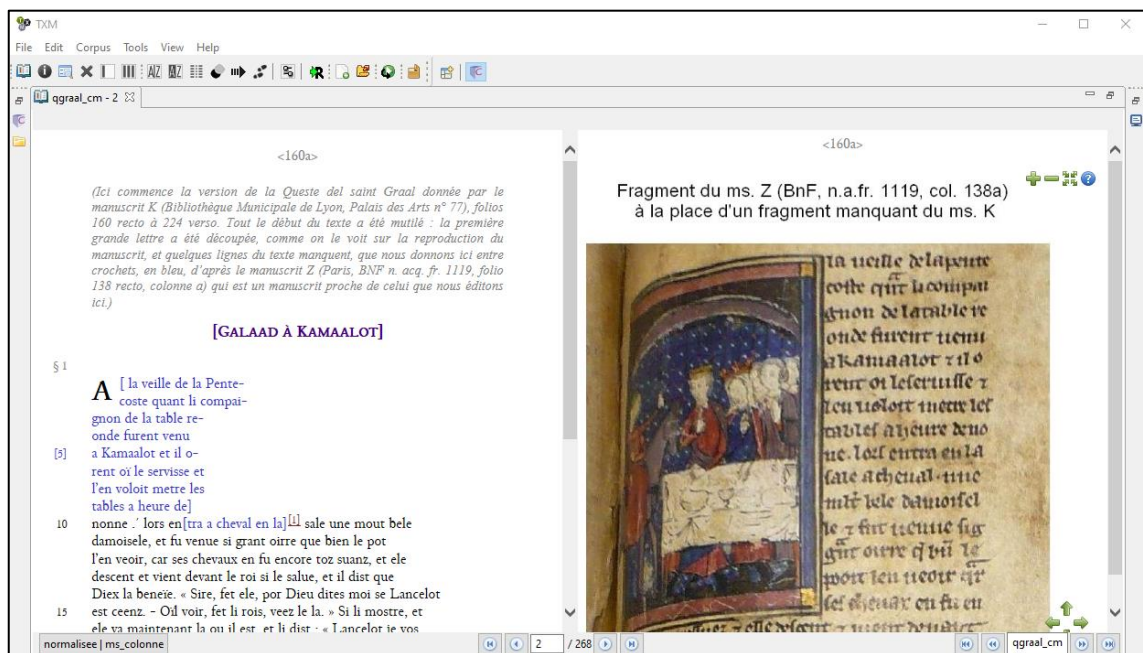


Figure 84 Pages in the GRAAL corpus

The tool does not offer the option to see a graph visualising the usage of a word over time. However, there are other visualisation options, such as a progression graph. A progression graph displays the evolution of one or several patterns throughout the corpus, producing either a cumulative or density graph (TXM User Manual, 2018).

The tool does offer other options, as well as allow the user to program and to use scripts to get certain results. These options and the results that they can produce is beyond the scope of this study. This study does not include the type of results a person can get by programming.

BNCweb (CQP-edition)

The results for the simple query for the usage of the word *mountain* in the corpus are shown in Figure 85. The query is repeated at the top of the screen, and information regarding the total number of hits, the total number of texts that it appears in, as well as the frequency is given. The results for the query are shown in sentence view but can also be displayed as keywords in context (KWIC).

Your query "mountain" returned 3816 hits in 968 different texts (98,313,429 words [4,048 texts]; frequency: 38.81 instances per million words) (0.099 seconds - retrieved from cache)

No	Filename	Text
1	A04_1527	The third perspective is Kao yuan , in which the viewer is looking up toward a mountain scene, as William Willetts puts it, 'through successively receding heights represented by flat parallel planes, each with its own horizon'.
2	A05_1092	Where can Jenny have been, in the course of her adolescence, to be willing, if only out of nervousness, to accept that the Reds in Spain have been swept out from under the bed and up into mountain caves?
3	A06_957	Your cheeks like damask, the soft white loveliness of your breasts, leading to the firm dark mountain peaks of your, Laura, now I'm dreading which part of my body he will choose next on which to turn the great white beam of his fucking sincerity.
4	A08_990	Genius is the bust of Beethoven and Keats dying and Shelley dying and the size of War and Peace and poor old Sartre banging away at his trilogy and Hemingway paring it down to its essence and Monet unable to distinguish colours any more and Picasso staring out at the camera with his chest bare and his eyes blazing and Cézanne snarling like a dog and then walking out of Aix with his canvas and paints on his back to paint that mountain and Byron dying and Pushkin dying and all the rest of it.
5	A08_1661	Now planning huge work to take place simultaneously in every town in Greece and on every mountain .
6	A0C_852	The wines include Le Bonheur Blanc Fume (Sauvignon Blanc) from Stellenbosch which is unwooded with a fresh, grassy character; Fleur du Cap Chenin Blanc Sec (crisp and fruity); Witzenberg Emerald Stein (semi-sweet Fleur du Cap), and Roodebloem from the Bergkelder or mountain cellars' of Stellenbosch.
7	A0F_117	You're making a mountain out of a molehill, Dorothy.
8	A0L_272	Go to Woodstock, the sea, the top of a mountain , a river, go forever from the flat respectability of home and market town.
9	A0N_1972	He had started out to make a rough count of the houses to be visited and then let his thoughts drift into a reverie of his own old home, the far tropical look of the mountain skyline beyond Loch Arkaig on the rare hot days.
10	A0P_408	Leonard recently referred to the memory of his father as 'a dark mass or mountain ,' of which, clearly, the details were too painful for the young boy to register or the adult to express.
11	A0P_409	(The image actually appeared in a somewhat different way in The Favourite Game : 'Concerning the bodies Breavman lost ... a man on the mountain ,' a reference to the cemetery on Mont Royale probably.)

Figure 85 Results for a query in BNCweb

One can click on one of the highlighted instances to see the instance in context (Figure 86), as well as see more information about the source text, for example, AOF 117 (Figure 87).

A0F: <s>-units 112 to 122 (of a total of 3417 <s>-units)

<<	>>	File info for A0F	Go!	Show POS-tags	Colour wordclass	No audio available
112		'You already have.	113	The problem is you can't accept that fact.'		
114		'Don't you be patronizing with me. 115 I'll take it all the way to the Senate if I have to.'				
116		'Look, do you really think that the Senate of the University of London is going to care two hoots about a footing little first-year lecture? 117 You're making a mountain out of a molehill, Dorothy. 118 All that's happened is I decided I was going to give the Bemini lecture this year. 119 If you've got some new material on him that you want to share with us, I'm more than happy to arrange another lecture for you later in the term, but frankly, as you've apparently given the same lecture on him for the past ten years, I can hardly be accused of interfering with academic freedom, can I?'				
120		'Who on earth told you that?'	121	(It wasn't quite true.)		
122		I can't remember now.				

Figure 86 A search result shown in more context in the BNCweb

BNC header information for file A0F	
Title:	Part of the furniture. Sample containing about 39211 words from a book (domain: imaginative)
Spoken or Written:	Written
Number of Words (tagged items):	40,478
Average sentence length (<w>-tags per <s>-unit):	11.8461
Derived text type:	Fiction and verse
Genre:	W:fict:prose
Text type:	Written books and periodicals
Publication date:	1985-1993
Age of Author:	15-24
Domicile of Author:	UK and Ireland
Sex of Author:	Male

Figure 87 Details for a file in the BNC

Apart from simply displaying the results, the BNCweb has various processing options for the results of a query. A user can filter the results to work with a smaller, randomly created, subset. A user can see the frequency breakdown, sort the results or search for collocations. The results can be downloaded, saved or categorised. The distribution across different types of texts (as specified in the bibliographic metadata) can be viewed (Figure 88).

Text Domain:				
Category	No. of words	No. of hits	Dispersion (over files)	Frequency per million words
Informative: Leisure	12,191,902	1,571	207/437	128.86
Informative: Natural and pure sciences	3,818,803	256	32/146	67.04
Informative: Applied science	7,173,003	313	130/370	43.64
Informative: Belief and thought	3,037,532	108	40/146	35.56
Imaginative prose	16,496,408	556	174/476	33.7
Informative: Arts	6,574,853	177	85/261	26.92
Informative: World affairs	17,244,523	459	144/483	26.62
Informative: Commerce and finance	7,341,009	82	43/295	11.17
Informative: Social science	14,025,338	144	62/526	10.27
total	87,903,571	3,666	917/3,140	41.7

Figure 88 Distribution of results in the BNCweb

Although technically probably a minor challenge, the tool does not include an option to visualise the frequency of words over a period of time, as is seen in many other tools for digital humanities, such as the Google Books Ngram Viewer and the HathiTrust+Bookworm.

BYU Corpora

As was explained earlier, there are various search options above the search fields from which a user can select to determine the type of results the user will retrieve. The BYU-BNC has the following options: “List”, “Chart”, “Collocates”, “Compare” and “KWIC”. The iWeb corpus has the options: “List”, “Word”, “Browse”, “Collocates” and “KWIC”.

The list option from both corpora shows the frequency of matching items (Figure 89). A user can click on a group to see those items in context (Figure 90). In the BYU-BNC a user can then click on a specific item to see more detail about the file (e.g. date of publication and title) and an expanded context for the sample text (Figure 91). In the iWeb corpus, however, instead of clicking on an item to see information about the file where this item is from, a user can link directly to the webpage where this item was found. Unfortunately, as the web is ever-changing, it can lead to dead links.



Figure 89 Results listed in corpus.byu.edu

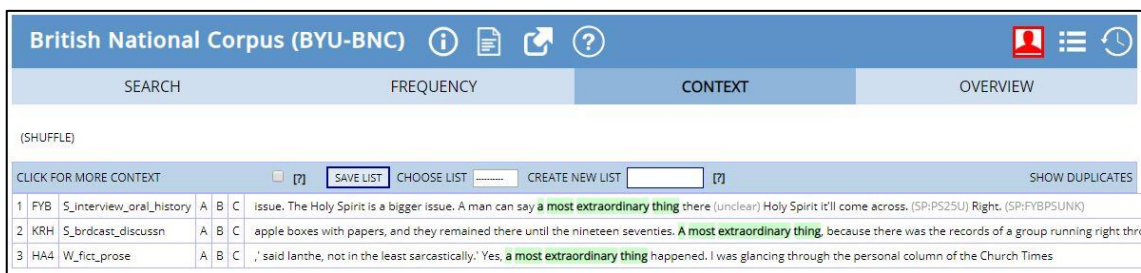


Figure 90 Items in more detail in the corpus.byu.edu

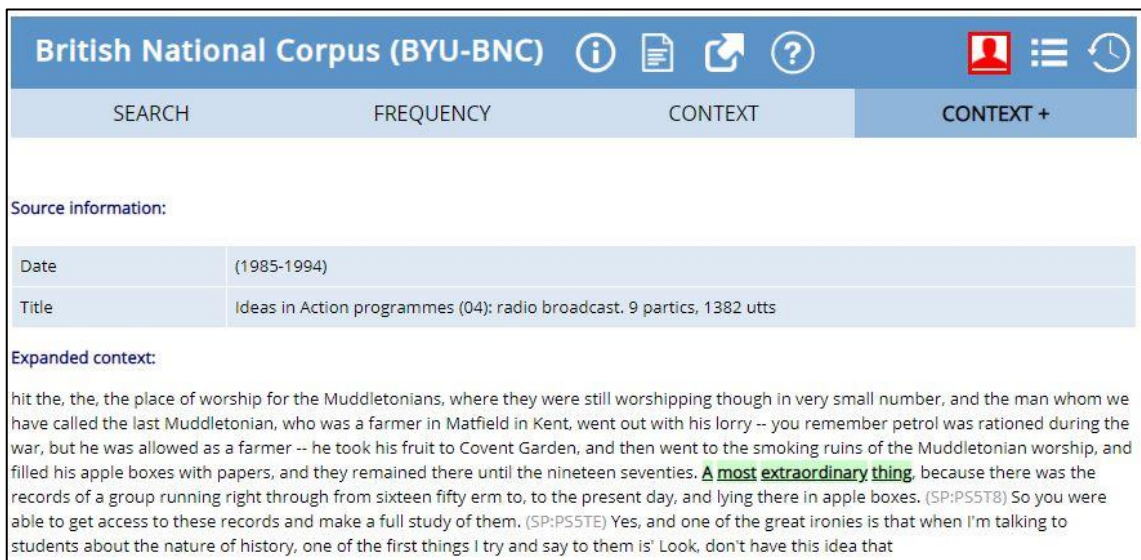


Figure 91 Expanded context in corpus.byu.edu

The chart option in the BYU-BNC shows the frequency of the results according to genre, as specified by the BNC. As there are no genres in the iWeb corpus, there is no chart option. Both the BNC and iWeb have a collocates option, which allows a user to see what words occur near other words. The BNC includes a compare option, which allows a user to compare the collocates of two words. Both corpora have a KWIC

option so that a user can see the patterns in which a word occurs, by sorting words to the left and right.

The iWeb has a word option that gives more information about a specific word and includes links to sites with more information about the word, for example, definition, synonyms, a link to images, collocates, clusters and concordance lines (Figure 92). The browse option for the iWeb corpus, is a different interface to search for words.

The screenshot shows the iWeb interface for the word "evening". The top navigation bar includes "SEARCH", "WORD", "CONTEXT", and "HISTORY". The "WORD" tab is active, displaying the following information:

- evening** (NOUN) #1508
- DEFINITION:** 1. the latter part of the day (the period of decreasing daylight from late afternoon until nightfall) 2. the early part of night (from dinner until bedtime) spent in a special way 3. a later concluding time period 1 2 3 4 5
- SYNONYMS:** twilight, sunset, twilight, dusk, nightfall, sundown, even
- COLLOCATES:** morning, weekend, afternoon, meal, dinner, summer, entertainment, dress
- CLUSTERS:** evening with, evening at, evening in, evening for, evening news, evening meal, evening when, evening i, evening, friday evening, saturday evening, one evening, sunday evening, thursday evening, tuesday evening, wednesday evening, monday evening, evenings and weekends, evening and weekend, evening primrose oil, evening of music, evening and i, evening when i, evening of fun, evening of july, in the evening, in the evenings, for the evening, on the evening, for an evening, morning and evening, during the evening, on friday evening, evening and weekend hours, evenings and on weekends, evening and weekend work, evening and the morning, evening under the stars, evening and weekend classes, evening and early morning, evening of the day, later in the evening, late in the evening, in the early evening, end of the evening, earlier in the evening, rest of the evening, in the late evening, part of the evening
- WEBSITES / VIRTUAL CORPORA:** parentseveningsystem.co.uk, willyweather.com, grubclub.com, simplydresses.com, eldersweather.com.au, es-static.us, parisciel.com, WeatherWatch.co.nz-Feb 20, 2016, tucsonweather.us, teranicouture.com
- CONCORDANCE LINES:** 1 sitcomsonline.com But when they do finally take the time for a **sun** **evening** **alone** **they** miss all the chaos. # Written by Rich

Figure 92 Page for a word in corpus.byu.edu

It is also possible to specify how the results must be sorted (e.g. frequency) and specify some display preferences, such as how many hits to be displayed.

2.7.6. Complexity of use

In this section the ease with which a user can use the tool will be considered. The more complex a tool is the more difficult it is to use, and conversely, the simpler a tool, the easier it will be to use. When thinking about the complexity of a tool, one should consider how hard it is to learn how to use the tool. The complexity of the tool will have a bearing on the audience that will use the tool. A very complex tool that takes time and effort to master is probably aimed at scholars or researchers. On the other hand, a tool

that can be used easily is probably suitable to be used by interested laypersons or researchers from fields other than linguistics or related fields.

Google Books Ngram Viewer

To perform a simple search on the Google Books Ngram Viewer is very easy. A user can open the tool and get going almost immediately, either by changing the sample query or entering his/her own query. In order to perform more complex queries, for example to search for a word that is in a certain word class, a user will have to consult the help file.

HathiTrust+Bookworm

It is very straightforward to use this tool and the options that it provides are self-evident. However, it does not offer very advanced options.

Perseus Project

The Perseus Project is relatively easy to use. The intuitive graphical user interface uses search fields, check boxes and links to enable the user to search in the library and move through the library. Descriptive labels are used for most items in the project which makes it easy to use. As the focus of the library is on texts from the Greco-Roman world one would assume that the users will be fairly educated, and if they are not scholars in this area, that they would have enough knowledge to know what texts to search for or explore. However, one of the aims of the tool is to provide access to classical texts and resources to interpret classical texts to anyone in the world who is interested (Crane, 1998). The project indeed succeeds in fulfilling this aim.

Voyant tools

One of the biggest advantages of Voyant Tools is its ease of use. One does not need specialist knowledge from either the field of linguistics or digital humanities. The tool is intuitive and can be used by researchers or lay persons.

TXM

TXM demonstrates how powerful a tool can be if it can use the metadata about the text when filtering or analysing. TXM presents a steep learning curve. It is a powerful tool for corpus linguists or other researchers who want to dedicate time to analyse texts encoded with TEI, yet, possibly making it less accessible for other researchers or interested lay persons. It is highly improbable that a lay person will open the tool and be able to use it. It is designed for skilled users who are experienced in the field of corpus analytics and those who want to take time to learn how to use the tool.

It is also evident that a user needs to understand the structure and encoding of each corpus, the tags used in each corpus, the meaning of the tags and possibly something of the encoding format (e.g. XML) used. For example, a user needs to know that the GRAAL corpus uses the tag *pos* and the VOEUX uses the tag *frpos* to annotate part-of-speech tags, or that the VOEUX corpus uses the tag *loc* to store the author of the text, or that the GRAAL corpus uses levels of direct speech with the tags *q*, *q1* and *q2*. Furthermore, although not technically required to know CQL, the tool can be used more effectively if a user is familiar with CQL. This can clearly be seen in the query assistant when a list of options is given to the user (such as *q*, *div1*, *edville*) without an explanation of what these options might mean.

It has been mentioned that creating a selection (subset of data) can be complex and have limited functionality. This will be illustrated by examples. It is possible to create a partition where all direct speech has been selected, but the functionality from the tool on a partition is limited. There are more functions available for a subcorpus, but it is difficult to create a subcorpus of all direct speech, as each number must be selected individually.

TXM provides powerful search options, and other analysis options for texts that are out of scope of this research, as it requires significant effort or training from the user to gain knowledge about the corpus, encoding, search language and the tool itself. It can be recommended to linguists and other scholars who wish to analyse annotated corpora but is not suitable for interested lay people or scholars whose main interest and knowledge is not in the encoding of texts or a complex search language.

BNCweb (CQP-edition)

The BNCweb (CQP-edition) is fairly user-friendly and intuitive for simple queries. A user can open the tool, immediately see the search field and type a word or phrase to search for. Even most of the filtering options are self-evident. The genres of the corpus might not be as clear, as only the codes for the genres are given, but a link to the descriptions of the genres is readily available. One could assume that this tool is for users who are familiar with the BNC and wish to analyse this corpus; as such, they are probably familiar with the design of the corpus and the use of the codes in this corpus.

For more complicated queries, a user would need some training. Linguists using this tool might already be familiar with advanced query languages for corpora and would therefore find it easy to use. It is therefore probably not ideal for a researcher in humanities not familiar with corpus linguistics, nor an interested lay person.

BYU Corpora

The corpus.byu.edu tool is not particularly difficult to start using, as the search field is clearly visible, and the part-of-speech tags can be selected from a dropdown box. Yet, a user must pay attention to the fact that there is a type of workflow or progress from one tab to the next. A user first searches and is then taken to the next tab, which shows the frequency. To see context, the user either has to select the next tab (Context) or has to click the link of the item (s)he wishes to see.

A user will have to consult the help files to understand how to construct a virtual corpus, and if (s)he is not a linguist, will probably need to consult the help files for some more advanced options, such as collocates or comparisons for corpora that have such options.

2.7.7. Help files

The help documentation that is available for each tool will be discussed in this section.

Google Books Ngram Viewer

The help file for the Google Books Ngram Viewer is found by following a link at the bottom of the main Ngram Viewer page. The help file itself is clearly structured, with many helpful examples.

HathiTrust+Bookworm

The HathiTrust+Bookworm tool itself does not provide any help documentation. As the tool is not difficult to use and does not present complex searching options, it can be used without instructions. However, as has been noted, there are some places where extra information could be useful, for example the meaning of the metadata fields.

There have been some academics or researchers involved in the development of the tool, and as such there are numerous articles written about the tool or where the tool has been used. Some extra information can therefore be obtained from external sources, but none from the site itself.

Perseus Project

The Perseus Project contains a well-developed help section. The instructions are clear, and screenshots are included to aid with understanding.

Voyant Tools

There is a very well developed help guide for Voyant Tools, explaining general actions, such as uploading a corpus, to specific details, such as how each tool works.

TXM

There are various help files on various platforms available for TXM. In the first place, the tool itself has a Help menu, with links to various sources. The Textométrie website also has information about TXM, as well as links to information sources, such as manuals. The project page for TXM on SourceForge.net also hosts help documentation. The information sources include manuals, brochures, a wiki, information on shortcuts, amongst others.

Unfortunately, it seems that most of the documentation is in French, for example the wiki and the most up-to-date user manual. There is an option to translate the wiki automatically to English and Russian.

The most recent English manual does seem to be clear and effective. It describes the functionality of the tool fairly well. However, the screenshots of this manual have not been translated.

The purpose of the documentation is to describe the tool; it does not contain much information about corpus analysis and contains limited information about the individual corpora. A user will need prior knowledge about those aspects.

BNCweb (CQP-edition)

A user does not need much help to use this tool for a simple search. However, help is needed for more complicated search queries. There is a link to a cheat sheet for the simple query syntax on the home page of the tool (BNCweb (CQP-edition), n.d.). The cheat sheet is a summary of the information in the book *Corpus Linguistics with BNCweb – a Practical Guide*. The cheat sheet is clearly written, contains some commonly used features and is enough to get a user started. However, a user who wishes to utilise the tool with maximum efficiency and effectiveness should consult the book.

There is no link to information about the CQP syntax on the tool itself. Presumably a user should be familiar with the syntax already or learn about the syntax from other sources.

There are links to information about the corpus and the tagset used in the corpus.

BYU Corpora

The corpus.byu.edu has well-developed help documentation which is context sensitive (see Figure 93). For example, as the user clicks on the option to create virtual corpora, the help changes to display information about virtual corpora. This is helpful as it

provides the right information at the right time; however, it can be confusing as well. For example, if a user is looking for something specific in the help documentation, but does not know what option to click on to get to that section of the help documentation, it can be frustrating.

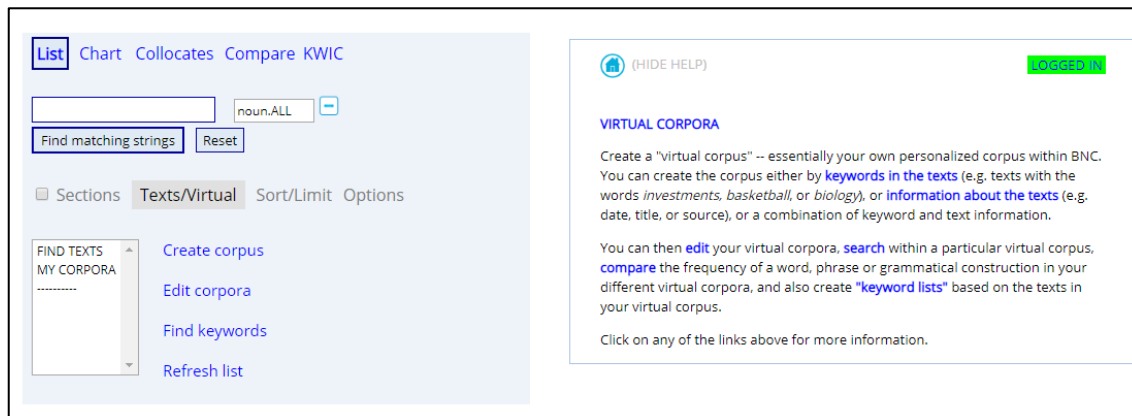


Figure 93 Help documentation in corpus.byu.edu

2.7.8. Corpus design

The discussion on metadata has already touched on the importance of the design of a corpus. It has been mentioned that metadata can allow a user to understand what material is in a corpus and exactly what is being studied. Representativeness and balance in a corpus are topics that are beyond the scope of this research. However, they are acknowledged as important.

Some criticism regarding corpus design will be discussed by considering the corpus design of one of the tools examined in this study, the Google Books Ngram Viewer.

It has been argued that the corpus used for the Google Books Ngram Viewer is similar to a library in which only one of each book is available (Pechenick et al., 2015: 2). New editions or reprints of books already in the corpus also have an influence on the resulting corpus (Pechenick et al., 2015: 2). To counter the library-like nature of the Google Books Ngram Corpus where there is only one copy of each book and each is treated as equally important, it has been suggested that popularity filters should be applied to the corpus (Pechenick et al., 2015: 2). For example, frequencies could be adjusted according to book sales, library usage or how often each page has been read on Amazon's Kindle (Pechenick et al., 2015: 2).

Research suggests that there is a delay of about ten years before historical events find prominence in literature, which limits the usefulness of the Google Books Ngram Corpus as an exact indicator of culture at a specific time (Pechenick et al., 2015: 4).

From the 1900s to the present there is an increased use of scientific publications in the general corpus (Pechenick et al., 2015: 23). Therefore Pechenick et al. (2015: 19) question the conclusion reached by Michel et al. (2011) that the current culture forgets more quickly than before. This is because their study was performed on the general English corpus which includes numerous scientific works and as such would have been influenced by the practice to cite more recent authors. As there is a marked difference between the two versions of the corpus used in the Google Books Ngram Viewer, a reader should be aware of what version has been used in the research as that will influence the results of the study. Pechenick et al. (2015: 12) have shown that the first version of the English fiction corpus was not effectively filtered and therefore still contained many scientific terms.

2.8. Conclusion

In this chapter, the growing number of digital collections were discussed. From mass-digitisation efforts, such as Google Books or HathiTrust, to smaller digitisation efforts, all projects contribute to the large amount of digital items available to users.

Not only are these digital items used in traditional ways, such as access and in-depth study, digital collections are being used in new and innovative ways, specifically using computational methods to do research on large collections. Various interesting research studies were discussed.

Despite the growing collections and increased use of collections, there are some problems with research on large digital text collections. This chapter specifically considered the low quality of some mass-digitised collections; the lack of metadata (on a bibliographic level and a deeper level); poor quality of metadata; the uncertainty regarding the composition of some large collections; the issue of copyright; the value of computational methods; as well as the problems with the software that is available to explore text collections.

Of specific interest to this study, is the relationship between the collection and the software that can be used to explore and analyse the collection. The quality of software available to search and analyse digital text collections will have a direct influence on the research that can be done on digital text collections. Similarly, the metadata assigned to a collection will influence what functions the software can offer.

As the metadata applied to text has such an influence on retrieval, this chapter explored the types of metadata that can be assigned to words and a text, and these were categorised as various levels of metadata, namely, morphological, syntactic,

semantic, functional and bibliographic. On the morphological level, the metadata that were identified are the lemma of a word and the part-of-speech category of a word. These metadata are relevant for each word in a text. If the lemma of a word is known, then a user can broaden their search and search for all variants of a lemma. Similarly, knowing the part-of-speech category of a word can improve retrieval by allowing the user to filter according to specific categories of words. Various standards that can be used to encode these metadata were discussed, for example, the Penn Treebank tagset and the Universal POS tags from Universal Dependencies. On the syntactic level two main types of grammars were discussed. The type of grammar that is relevant to this study is dependency grammar. This encoding makes the relationship between words clear. These metadata allow a user to search for words that modify or are modified by specific words, or search for words with specific relationships between each other. The syntactic relationships proposed by the Universal Dependencies were shown. The next level, the semantic level, considers the meaning of each word. This is useful as a word can have several meanings and if the meaning of a word is encoded then a user can search for a word with a specific meaning and exclude other homographs. WordNet as a database of senses was discussed. On the functional level, the concept of encoding structures in a text was discussed and the well-known encoding standard for such structures, TEI, was discussed. Encoding this information is beneficial for retrieval. A user can filter to only search for information in specific structural sections. The last level that was discussed is the bibliographic level. This level contains standard bibliographic metadata and there are various standards in which this type of data can be captured. These standard metadata are necessary to do basic retrieval, for example limiting to search from certain author or subject.

After considering the metadata that can be applied to texts, the ability to retrieve words or phrases was explored. In order to determine to what extent current tools (software) allowed a user to search and explore a text collection, a number of tools were examined.

Key points from the discussion about the various tools are summarised in Table 5.

Table 5 Comparison of tools

	Google Books Ngram Viewer	HathiTrust+ Bookworm	Perseus Project	Voyant Tools	TXM	BNCweb (CQP-edition)	BYU Corpora
Corpus							
Name of corpus	Google Books	HathiTrust Digital Library	Perseus Digital Library (divided into collections)	Corpus independent Examples: Bleak House Romeo and Juliet	Corpus independent Examples: Graal Voeux	BNC (British National Corpus)	Corpus independent Examples: BNC iWeb
Size	Total: Over 8 million volumes Over 800 billion words English: Over 4 million volumes Over 400 billion words	Over 17 million volumes	Per collection Greek and Roman collection English: 44 462 693 words Greek: 13 507 448 words Latin: 10 525 338 words	Corpus independent Bleak House 361 153 words Romeo and Juliet 27 463 total words	Corpus independent Graal: 118 719 words Voeux: 61 197 words	100 million words	Corpus independent BNC: 100 million words iWeb: 14 billion words
Availability of corpora	N-grams are available for download.	Items in copyright are not accessible, others are accessible through the	Items in the public domain are available for free download in XML format	Corpora are imported by the user.	Corpora are imported by the user.	Available for download (users agree to the associated licence)	BNC Corpus is open and was imported into this tool

	Full-text and metadata are not available.	HathiTrust Digital Library.					iWeb Not available for download, but opens as a user searches through it.
	Google Books Ngram Viewer	HathiTrust+ Bookworm	Perseus Project	Voyant Tools	TXM	BNCweb (CQP-edition)	BYU Corpora
Interface design							
	Simple	Simple	Simple	Simple	Complex	Intermediate	Simple to intermediate
	Google Books Ngram Viewer	HathiTrust+ Bookworm	Perseus Project	Voyant Tools	TXM	BNCweb (CQP-edition)	BYU Corpora
Metadata							
Morphological data	Yes (Universal POS tagset)	No	Yes	Corpus independent Bleak House: No Romeo and Juliet: No	Corpus independent Gaal: Yes (CATTEX2009 tag set) Voeux: Yes (fr.par model)	Yes (C5 tagset)	Corpus independent BNC: Yes (C5 tagset) iWeb: Yes (tagset not provided)
Syntactic data	Yes (dependency syntax representation)	No	No	Corpus independent Bleak House: No	Corpus dependent Gaal: No Voeux: No	No	Corpus dependent BNC: No iWeb: No

				Romeo and Juliet: No			
Semantic data	No	No	No	No	No	No	No
Functional data	No	No	Yes	Corpus independent Bleak House: No Romeo and Juliet: Yes	Corpus dependent Gaal: Yes Voeux: Yes	Yes	Corpus dependent BNC: Yes iWeb: No
Bibliographic data	No	Yes	Yes	Limited	Corpus dependent Gaal: Yes (limited) Voeux: Yes	Yes	Corpus dependent BNC: Yes iWeb: No
	Google Books Ngram Viewer	HathiTrust+ Bookworm	Perseus Project	Voyant Tools	TXM	BNCweb (CQP-edition)	BYU Corpora
Search options							
Simple word search	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Phrase searching	Yes	No	Yes	Yes	Yes	Yes	Yes
Truncation, wildcards or other command line options	Limited	No	No	Limited	Yes	Yes	Yes
Part-of-speech	Yes	No	No	No	Yes	Yes	Yes
Other grammatical features, e.g. syntax	Yes	No	No	No	No	No	No
Functional searching	No	No	No	Limited	Yes	Yes	No

	Google Books Ngram Viewer	HathiTrust+ Bookworm	Perseus Project	Voyant Tools	TXM	BNCweb (CQP-edition)	BYU Corpora
Filtering							
Bibliographic data	Limited (date and language)	Yes	No (except browsing)	No	No	Yes	Limited
	Google Books Ngram Viewer	HathiTrust+ Bookworm	Perseus Project	Voyant Tools	TXM	BNCweb (CQP-edition)	BYU Corpora
Search results							
Graph that plots frequency over time	Yes	Yes	No	Limited	No	No	No
KWIC view	No	No	No	Yes	Yes	Yes	Yes
Link to more context	No (only indirectly)	Yes (for out of copyright options)	Yes	Yes	Yes	Yes	Yes
	Google Books Ngram Viewer	HathiTrust+ Bookworm	Perseus Project	Voyant Tools	TXM	BNCweb (CQP-edition)	BYU Corpora
Complexity of use							
Effort to learn how to use	Easy	Easy	Easy	Easy	Difficult	Moderate	Moderate
Presumed audience	Laypersons to scholars	Laypersons to scholars	Laypersons to scholars	Laypersons to scholars	Scholars	Scholars	Laypersons to scholars

	Google Books Ngram Viewer	HathiTrust+ Bookworm	Perseus Project	Voyant Tools	TXM	BNCweb (CQP-edition)	BYU Corpora
Help files							
Availability	Yes	No	Yes	Yes	Yes	Yes	Yes
Clarity	Good	N/A	Good	Good	Moderate	Good	Good

Though the power of searching in and analysing encoded texts has been demonstrated in the tools that were examined in the previous sections, there are clearly several limitations in these tools.

There is not one tool that would allow a researcher with little prior training or knowledge of XML, or knowledge of the corpus structure, to search in a large body of texts filtered to specific properties (e.g. direct speech), and then visualise the frequency of a certain word over time.

The tools from corpus linguistics come the closest to enabling a user to do this type of filtering; however, a user would need to know XML or TEI to some extent, as well as the query language used in these tools. It also seems that a user needs knowledge of the underlying corpus to use the query language effectively.

TXM does well to inform a user about the structure of the underlying corpus, and a user can filter effectively using the available TEI tags as the tags are parsed and presented to the user through the interface. Yet, it is very technical and requires an understanding of XML and knowledge of the software.

The corpus linguistic tools also do not typically have a visualisation function included to observe trends and are more aimed at traditional language research.

It seems the tools with simple interfaces, such as Google Books Ngram Viewer, are more readily used in research in the humanities (as seen from the references from articles). It will also be too time intensive for an interested lay person who wishes to confirm the use of a word in a particular type of text to learn XML and a query language before being able to use the tool to answer their question.

Yet, the tools with simpler interfaces lack the power of tools such as TXM or BNCweb that can filter on a fine-grained level. Google Books Ngram Viewer allows searching on morphological and syntactic properties but does not include filtering. HathiTrust+Bookworm does not allow searching on morphological or syntactic level nor on a functional level.

Furthermore, there has been little focus on how metadata should be used to extend bibliographic detail to a lower level. For example, if a text is written in English, but there is a quote in French, then the language of that section should be marked as foreign and then the tool should allow a user to filter on that. In the Perseus Project foreign words are encoded as such, but there is no filtering or searching option that takes it into consideration.

Apart from limitations in the tools, there are also problems with the encoding of the underlying data and how it is used by tools to enable retrieval. Though many texts have been encoded in detail (e.g. the British National Corpus, EEBO, Perseus Project) more research can be done on how encoding can be used to improve the retrieval of text on a fine-grained level for research in the digital humanities as well as use by lay persons.

In the light of this discussion, it is recommended that more work should be done to enable general users to be able to do advanced searching in a text collection. The following is suggested to achieve this goal:

- A metadata framework should be developed that allows for the encoding of metadata on different levels, from bibliographic metadata to fine-grained in-text metadata.
- This metadata framework could make use of existing standards.
- A tool should be developed that can be used to search in this encoded collection.
- The tool should allow people to search for words or phrases with certain properties, such as part-of-speech, direct speech, time period or language.
- The tool should be usable by people with little or no programming experience, and little or no knowledge of encoding standards.
- Through the tool, a user should be able to retrieve examples of a specific word or phrase.
- The examples of a specific search should link to more context, in other words, the context in which the item was used.
- The tool could include advanced display options, such as displaying results in context and provide a visualisation function, so that a user can observe trends of usage over time.

Before looking at how existing standards can be used, or modified, to encode texts for enhanced retrieval, the next chapter will first consider the research methodology of this study.

3. Research methodology

'Begin at the beginning,' the King said gravely, 'and go on till you come to the end: then stop.'

Alice in Wonderland by Lewis Carroll

3.1. Introduction

In the preceding chapters, the researcher has argued that retrieval of words or phrases from a large text collection could be improved by encoding the texts in the collection with detailed metadata. These metadata could allow a user to specify different properties of words or phrases that should be retrieved. The purpose of this study is to investigate this suggestion.

In order to answer the research question of this study the following need to be done:

- the current landscape should be reviewed to see how words and phrases are currently retrieved from text collections;
- a metadata data schema that can improve retrieval should be proposed;
- selected texts should be encoded with the suggested metadata schema;
- a system should be built to test whether retrieval is improved.

Work from different fields had to be considered in order to address the question posed by this study. This includes, but is not limited to, work from information science, computer science, corpus linguistics, digital humanities and natural language processing. The result would be an interdisciplinary study drawing from insights and research from diverse fields. No single, traditional method would be suitable to answer the question posed by this study. As such, a combination of methodologies had to be employed, namely, grounded theory research, phenomenological research, literature review, action research, case study research, heuristic evaluation and prototyping. For each approach or method, the researcher will first discuss the concept and subsequently how it will be used in this study.

Important concepts to consider in a research study are whether the study is a qualitative or quantitative study, the credibility of the study and ethical issues regarding the study. These concepts will also be discussed in this chapter.

3.2. Grounded theory study

In a grounded theory study, the researcher is unlikely to start with a theory or a preconceived framework (Salkind, 2010), rather the approach is to “begin with the data and use them to develop a grand theory that might be generalised to other settings, groups, and processes” (Leedy & Ormrod, 2020: 163). In this type of study, the theory emerges from the simultaneous collection and analysis of data rather than the research literature (Leedy & Ormrod, 2020; Salkind, 2010). There is little consensus among researchers about whether literature about the topic should be consulted before or after collecting and analysing the data (Leedy & Ormrod, 2020; Salkind, 2010).

The focus in grounded theory studies is typically on a process, specifically the actions and interactions of humans (Leedy & Ormrod, 2020; Nolas, 2011). The roots of this research approach are in sociology but have been used broadly (Salkind, 2010).

Data collection in a grounded theory study is typically very flexible and is likely to change during the course of the study (Leedy & Ormrod, 2020). In grounded theory research the collection of data and analysis of data typically occurs simultaneously (Pickard, 2017). Salkind (2010) explains that data collection generally starts with purposive sampling and ends with theoretical sampling.

An important part of grounded theory research is the development of categories to classify data and develop theory (Leedy & Ormrod, 2020; Pickard, 2017; Salkind, 2010). Participants are chosen carefully, as these participants should have certain attributes that can refine the theory or categories (Leedy & Ormrod, 2020).

When conducting grounded theory research, it is important that the researcher remains open to different options and possibilities and does not allow previous knowledge and theory to drive the data collection and analysis (Pickard, 2017).

Though not a typical grounded theory study, this study is influenced by the grounded theory approach and principles from this research design will be applied. It is not a traditional grounded theory study, because the actions and interactions of people are not studied. However, in this study categories of metadata are developed and refined as literature, tools and text samples are studied. As content and samples are studied, it will influence the categories and schema development. In the same way, the criteria for evaluating tools emerge from examining the tools and literature. Furthermore, as a contribution to theory development, a decision support system is developed to formalise the process in this study. In this way, no existing theoretical framework is used, but data collection and analysis occur simultaneously to inform the study.

3.3. Phenomenological study

In a phenomenological study the researcher explores people's experiences of a specific situation in depth (Leedy & Ormrod, 2020; Miller & Salkind, 2002). It could be that the researcher him-/herself has been in the situation in question and would like to understand what other people's perspectives and perceptions are (Leedy & Ormrod, 2020). An example would be to study what it is like to have a certain illness. In contrast to narrative research where a single case is studied in depth, the experiences of several individuals of a certain phenomenon are studied (Miller & Salkind, 2002).

The most typical data collection method to use in this approach is interviews (Leedy & Ormrod, 2020; Miller & Salkind, 2002). Generally a small sample is carefully selected (Leedy & Ormrod, 2020). When collecting data for this type of study, it is important that the researcher suspends any of his/her own preconceived notions to truly understand the phenomenon from the point of view of the participants (Leedy & Ormrod, 2020; Miller & Salkind, 2002).

After reading a phenomenological study the reader should have a deep understanding of what it is to live through a certain phenomenon without going through the experience him-/herself (Miller & Salkind, 2002; Pickard, 2017).

This study is not a phenomenological study per se, but again, uses some of the principles of this research approach. In this study, several tools used for searching in text collections are examined in depth to give a detailed understanding of the current possibilities in terms of retrieving words or phrases with detailed properties from a text collection.

3.4. Literature review

According to Leedy and Ormrod (2014: 51), a literature review "describes theoretical perspectives and previous research related to the problem at hand". Kumar (2014: 48) emphasises the importance of the literature review in the research process. Through analysing and discussing previous literature on the research topic, the researcher demonstrates:

- that (s)he has an in-depth understanding of the topic at hand (Hofstee, 2006: 91; Mouton, 2001: 87)
- that there is indeed a problem that exists that had not been addressed before and this research is not simply a duplication of work that has been done (Hofstee, 2006: 91; Leedy & Ormrod, 2014: 51; Mouton, 2001: 87)

- how the research fits in with the existing scholarship on the topic and can either confirm or contradict current research (Hofstee, 2006: 91; Kumar, 2014: 48)
- what methods or tools have been used previously to deal with problems in this area (Leedy & Ormrod, 2014: 51; Mouton, 2001: 87)
- the significance of the study and what contribution it hopes to make (Hofstee, 2006: 93)

In this study, the literature review will allow the researcher to explain what has been done regarding searching in large textual corpora, what tools are available to researchers in this regard and also what criticism has been found against the current tools and implementations. Furthermore, the literature review will reveal what standards are currently available to encode text to allow for retrieval on a greater level of granularity. This then will allow the researcher to answer the first three sub-questions.

The literature review in this study is not a typical literature review, because it does not only discuss what has been reported on in literature (e.g. studies in articles, books, conference proceedings). A large part of the literature review consists of a review of software programs that have been developed and are used to retrieve words or phrases from text collections. This is important as it helped the researcher understand what has been done in this field and what contribution can be made in the field. Though the review of software forms a crucial part of the review, relevant literature is also discussed.

It is also important to consider the nature of the literature review in the context of the grounded theory approach taken in this study. Based on the analysis of the data (from literature, tools and examples) a conceptual framework for evaluating search tools is developed, as well as categories for metadata. The data analysis and collection happen simultaneously and inform each other.

3.5. Action research

Reason and Bradbury (2006: 1) define action research as “a participatory, democratic process concerned with developing practical knowing in the pursuit of worthwhile human purposes, grounded in a participatory worldview which we believe to be emerging at this historical moment”. It typically aims to solve a problem through practical solutions (Hofstee, 2006: 127; Reason & Bradbury, 2006: 1) and so create practical knowledge that is useful in everyday life (Reason & Bradbury, 2006: 2).

Kumar (2014: 160) distinguishes between two traditions, namely the British tradition that sees action research as a way to improve and advance practice, and the American tradition that sees action research as a way of collecting data with the purpose of social change. Reason and Bradbury (2006: 2) identify five features that characterise action research, namely, emergent developmental form, practical issues, participation and democracy, knowledge-in-action and human flourishing.

In this study the researcher aims to address the problem of lacking granular metadata in current search and retrieval tools of large textual corpora. The researcher does this by providing a practical solution by suggesting what metadata can be used to encode text in more detail. The study is not necessarily a participatory study, as there will be no participants in the study. However, to some extent the researcher will rely on the knowledge of the community as identified in the literature review to address the problem.

3.6. Case study research

In a case study, a particular instance or a few carefully selected instances are studied in depth and the focus is on the details of the entity being studied (Kumar, 2014: 155). The case being studied is either representative of a group of entities or very atypical (Kumar, 2014: 155). If the selected case is typical of a certain group then certain insights of the group can be drawn (Kumar, 2014: 155). This type of research is often used when detailed information is required (Hofstee, 2006: 123). Case study research mostly addresses 'how' or 'why' types of question (Yin, 2009: 8, 18) and is focused on exploring and understanding (Kumar, 2014: 155).

Case study research is often difficult to generalise and the subjectivity of the researcher should be kept in consideration (Hofstee, 2006: 123).

This study can be regarded as using the case study method, because the researcher selects a few texts to use in the study. Texts that have certain characteristics or features relevant to the study are selected. For example, if direct speech should be encoded and retrieved, there must be examples in the selected texts with direct speech. These texts can thus be seen as being representative of other similar texts. These texts are encoded with the metadata that the researcher identified and then used to test whether it is possible to improve retrieval when fine-grained metadata are used.

In this study it will be beneficial to use only a selection of texts, as the purpose of the study is to prove a concept and not to create a system for public use. If the system is

able to work on a few examples, the concept has been proven and it is not necessary to use more examples to demonstrate the same concept.

It is also useful to consider the prototype that will be developed in this study as a case. It is a single instance that will be studied in-depth and considers how retrieval of words or instances with specific properties can take place.

3.7. Purposive sampling

Another concept that is relevant to this study, specifically when considering the sample of texts selected for encoding, is purposive sampling. Leedy and Ormrod (2020: 456) define purposive sampling as a “sampling selection process in which participants or other units of study are chosen on the basis of how much and what types of information they can yield about the topic under investigation”. Pickard (2017: 64) also explains that in purposive sampling, units that are rich in information are studied in depth. Purposive sampling is often used in qualitative studies and also in mixed-methods studies (Leedy & Ormrod, 2020). A researcher will choose the units of study for a particular purpose as the unit represents something that the researcher wishes to study (Leedy & Ormrod, 2020). Pickard (2017: 64) discusses purposive sampling in the context of choosing participants for a study and explains that one of the approaches to purposive sampling could be to set up criteria for selection, where the criteria are the outline of the nature of the participants that are needed for the purpose of the study. Most importantly, the purpose of the research should determine the type of sampling that is done (Pickard, 2017: 66).

The goal of this study is to demonstrate that texts can be encoded with detailed metadata, beyond the level of the volume or book and that queries can be constructed that include metadata from various levels. In order to do this, metadata appropriate for this purpose had to be identified and applied to sample texts that contain the features that are identified in the metadata. This means that if the metadata suggested in this study encode direct speech in a text, then a text that contains direct speech should be included. In other words, the units (texts) to be used in this study should be selected based on the information that they can reveal. The set of criteria for selecting the texts is based on the metadata suggested in this study. All the metadata suggested in this study had to be present in some texts. Not all texts could include all metadata, but a metadata element had to be present in at least one of the texts, ideally more.

The nature of sampling would therefore be highly selective and only a limited number of texts would be selected, and in these selected texts only relevant samples would be encoded. If there are sufficient examples of certain metadata elements to prove that

these metadata elements could be successfully combined with other levels of metadata and be used in the retrieval process, then it would not be necessary for this study to encode more examples. This means that there will not be sufficient data to conduct statistical analysis. Statistical analysis is also not relevant to the study as the purpose of the study is not to do measurements and calculate ratios.

3.8. Prototyping

A prototype is “a working model built to develop and test design ideas” (Walker et al., 2002: 661). It has also been referred to as “an initial version of a software system” (Suranto, 2015: 150). The creation of prototypes is an important step in the design process (Hanington & Martin, 2012: 138). Prototypes can serve several purposes. They can be used to communicate ideas about a design, create a proof of concept, conduct basic usability testing, determine if an idea is worth further investment (Klimczak, 2013: 117) and determine or elicit user requirements (Dhandapani, 2016; Suranto, 2015). Furthermore, by using prototypes, some problems can be identified early on in the design process before wasting resources on a flawed design (Walker et al., 2002: 661).

There are different types of prototypes, varying according to their similarity to the final product. Prototypes that are more similar to the final product are called high-fidelity prototypes and prototypes that are less like the final product are called low-fidelity prototypes (Walker et al., 2002: 661-662). Low-fidelity prototypes are often used early in the design process and can appear as sketches or storyboards on paper (Hanington & Martin, 2012: 138). High-fidelity prototypes are useful later in the design phase as they are more refined and may include basic functionality and allow for interaction (Hanington & Martin, 2012: 138; Walker et al., 2002: 662).

Walker et al. (2002: 661-662) discuss some advantages and disadvantages of the different prototyping techniques. It is quicker and cheaper to create low-fidelity prototypes, whereas high-fidelity prototypes are more time-consuming and expensive to produce. Low-fidelity prototyping allows for more iterations during development. However, low-fidelity prototypes may appear unprofessional and cannot convey the range of possibilities that high-fidelity prototyping does. Another disadvantage of high-fidelity prototyping is that designers or developers may be reluctant to change the design.

Different mediums can be used to create prototypes, most notably paper and computer, each medium presenting different advantages and disadvantages (Walker et al., 2002: 662).

The aim of this study is to determine if retrieval can be improved by encoding texts with fine-grained metadata. This will be tested by building a prototype of a retrieval tool. The tool receives some texts that have been encoded with the metadata proposed in this study. These are saved in a database. The interface of the tool allows a user to retrieve words or phrases from the database and to filter according to specific properties. The tool is regarded as a prototype as it does not have enough texts in the database to be considered a full-scale system. As there are not enough data in the database there is some functionality that is not logical to add at this stage; for example, it is not sensible to add the visualisation of trends, because there simply are not enough data. However, the purpose of the prototype, to prove the concept proposed by this study, is achieved. When a person is “not sure if something is even possible, prove it out ... [by] building a proof of concept” (Klimczak, 2013: 117). This prototype shows that what is proposed in this study is possible. Based on the results of the prototype and the experience of encoding the data, one subsequently can decide if it is worth expanding the prototype to a fully-fledged system, and make recommendations in this regard.

The type of prototype for this study is a high-fidelity prototype, as it is necessary to have interaction on the system in order to test the functionality of the system. The medium used for the prototype is a computer, as interaction is necessary.

At this point it is pertinent to discuss the concept of retrieval as used in this study. The aim of the prototype is to determine whether different levels of metadata can be combined to allow users to search and filter according to various properties. In this way, certain queries can now be executed which were not possible before. Words or phrases with different types of properties can be gathered together (filtered) programmatically and studied. In order to achieve this goal, the prototype uses exact matching of words and the properties saved for these words. This means that if a word is encoded as being direct speech, it should be returned when words in direct speech are being searched for. It should also take the hierarchy of items into consideration. If an author quotes another author, that quote will have two authors, which forms a hierarchy, and depending on whether the user searches on the in-text level or the document level, different authors in the hierarchy should be used.

The aim is not to perform retrieval experiments, where the recall and precision are measured, as are typical of experiments for TREC (TREC, 2020) or CLEF (CLEF, 2020). In other words, the aim is not to develop an advanced searching algorithm with various modern searching techniques such as fuzzy searching or semantic searching and calculate the effectiveness of the retrieval. As such, the goal of this study is not to

get statistical data to make a judgement about the effectiveness of retrieval in the prototype. The prototype uses exact matching of words and the properties saved for these words. The system will demonstrate improved retrieval if various levels of metadata can be combined, saved in a database in such a manner that a user can retrieve words or phrases with specific properties, ranging from morphological to bibliographic, and so allow the execution of queries that were not possible before.

The second aspect that will be taken into account when considering the improvement that this prototype brings is whether it is as accessible to laypersons as well as advanced users. The user should not need to understand the encoding of the underlying data or have knowledge of a query language to do this kind of retrieval.

Another important concept to discuss in this section is the quality of data. As the purpose of the study is to prove a concept in the form of a prototype, only enough data necessary to test the idea would be encoded. One encoder (the researcher) would be able to encode all the data for the project. At the very least this reduces the risk of differences between different encoders. However, it is also possible that the encoding of one person is not consistent across different items. As much as possible will be done to keep the encoding consistent, for example by referring back to previous examples and manually checking automated encoding. However, even if there were to be some inconsistencies or errors (for example an incorrectly tagged part-of-speech category) this would not influence the main findings of the study, namely that various levels of metadata can be combined to enable retrieval of words or phrases with specific properties. The study shows that various levels of metadata can be used to enable retrieval on a fine-grained level.

Furthermore, though the study recognises the importance of quality metadata, it is also important to recognise that the level of quality of metadata can depend on the requirements of the research project and be decided on when creating a collection. It could also be sufficient to be aware of the level of quality of the metadata and take that into consideration when doing research. The larger the collections are, the more difficult and expensive it will be to ensure the quality of the metadata. However, if the collections are very large, it might be acceptable to have some errors in the metadata. In other research projects it might be critical that the quality of the metadata is exemplary. In such cases, it might only be feasible to have smaller collections. The main criticism regarding Google Books Ngram Viewer, for example, is not about the morphological or syntactic metadata that were generated automatically, but is against the lack of bibliographic metadata (refer to section 2.7.2). In terms of

HathiTrust+Bookworm, the product shows that detailed bibliographic metadata can be used to filter a collection (on the document level, not, as this study suggests, on the text level), but of course it must be noted that the quality of the metadata will have an influence on the results and must be taken into consideration when interpreting results (refer to section 2.7.2).

If a fully-fledged, public system were to be developed, careful consideration should be given to the required quality of the metadata. Different techniques can be used to ensure quality, for example different encoders can be asked to check each other's work. Such an approach was followed by Finlayson (2015) when annotating a corpus of the purposes of machine learning, where an adjudicator checked the work of the annotators. However, this would increase the labour necessary to encode the collection. This means that the requirements of a specific customisation/implementation could determine the quality of the encoding and should be considered. If some errors in encoding are acceptable (specifically if a large corpus is used) then more automated tools can be used. However, if it is crucial that the encoding is of a high quality, more manual checking and more checking by other encoders will be required.

3.9. Heuristic evaluation

There are various methods that can be used to evaluate systems. One such a method is heuristic evaluation. During heuristic evaluation experts evaluate an interface systematically according to certain principles (also known as heuristics) (Shneiderman & Plaisant, 2010). Through conducting such an evaluation, problems in the system can be identified.

It is usual for the expert to examine the system several times, where the expert will get a general feel for the system during the first round and will then examine the system during the following rounds according to the set of criteria that have been determined (Barnum, 2010). It is not necessary for expert evaluators to have experience in the field or domain where the system will be used (Barnum, 2010).

Heuristics should be supported by good design principles (Preece et al., 2011). There are general design heuristics that are often used to evaluate systems, for example, the well-known heuristics developed by Nielsen (1995). However, general heuristics are not necessarily useful in all evaluations and experts may adjust heuristics or create their own (Preece et al., 2011). Nielsen (1995) differentiates between general heuristics, that can be used to evaluate any product, category-specific guidelines that can be used to evaluate products in a certain category, or product-specific guidelines

that are only relevant to a specific product. Between five and ten heuristics are sufficient for experts to conduct an evaluation (Preece et al., 2011).

Heuristic evaluation has been criticised for pointing out minor or even false problems (Molich & Dumas, 2008). However, heuristic evaluation has been used successfully in various studies (e.g. Kjeldskov et al., 2010)

Heuristic evaluation was firstly used in this study to evaluate various tools that are used to work with digital text collections. It has already been mentioned that the literature review in this study is not only a traditional review, but in order to understand the field, the researcher examined various tools. The researcher identified various criteria (heuristics), based on the relevance of the criterion to the main objectives of the study, according to which the usefulness of the tools could be judged. The criteria according to which the tools were evaluated were the interface design (in terms of ease of use by lay people and scholars), the metadata available in the tools according to which a user can search and filter, search and filtering options that the tool included to enable advanced retrieval, the complexity of the tool, the help files and the corpus design (where relevant). These are the important issues in terms of filtering search results in this study. The researcher then systematically reviewed the tools according to the criteria and commented on strengths and weaknesses of the various tools.

Heuristic evaluation is also used in this study to evaluate the prototype developed in this study to test the retrieval of words and phrases by using specific properties.

3.10. Qualitative and quantitative research

According to Leedy and Ormrod (2020: 113), the purpose of quantitative research is to seek explanations and predications that can possibly be generalised, whereas the purpose of qualitative research is to seek a better understanding of complex situations. In quantitative research, methods that allow the researcher to make objective measurements are typically used and the collected data are often analysed using statistical methods. In qualitative research the researchers typically try to keep an open mind and immerse themselves in a complex situation and the analysis of the collected data is more subjective in nature.

This study is not a quantitative study, as no statistical analysis is done. The purpose of the study is not to encode a large number of texts and calculate the relevant items that are retrieved or not retrieved and make judgements about the efficiency of the retrieval. That would not be suitable to this study, as the purpose is not to judge a new searching

algorithm, but to confirm whether layers of metadata can be used to retrieve specific items from a large collection.

Though the study is not a typical qualitative study, it is probably closer to being qualitative in nature, as the study is more descriptive and principles from grounded theory research are applied to develop categories of metadata. Furthermore, purposive sampling will be used to select a small sample of texts to test the proposal and the results will be discussed in detail in narrative form.

3.11. Credibility of the study

According to Leedy and Ormrod (2020: 117), a research study is considered credible when other researchers agree that the methods used are appropriate for the specific research project, the results seem trustworthy and the interpretation of the data seems reasonable. Important concepts when discussing credibility, are reliability and validity. Validity refers to the fact that a method used gives an accurate assessment of the object being studied and reliability refers to whether the method used consistently gives similar results (Leedy & Ormrod, 2020: 127-133).

Researchers follow different strategies to show the credibility of their research. The strategy most relevant to this study is using thick descriptions (Leedy & Ormrod, 2020: 118). By using thick descriptions, a study is discussed in such detail that other researchers can draw their own conclusions. The system developed in this study is discussed in depth and screen shots are used as illustrations.

Another aspect to increase the value of a study is to ensure that the conclusions drawn can be applied to other situations, referring to the generalisability of the study (Leedy & Ormrod, 2020: 119). Though this study will focus on a small sample of texts to prove a concept, the suggested metadata scheme is extensible and can be extended to other texts.

3.12. Ethical considerations

Ethical issues should be considered when conducting research. The following categories address most ethical issues, namely, protection from harm, voluntary and informed participation, right to privacy and honesty with professional colleagues (Leedy & Ormrod, 2020: 135). This research does not make use of human or animal participants, nor is it likely to have any impact on the environment. It will therefore not give rise to any ethical issues. The most problematic issue will be the use of texts; however, the researcher will choose texts that are in the public domain to avoid breaching any copyright restrictions.

3.13. Conclusion

In this chapter, the different methods and approaches that are employed in this study were discussed, namely, grounded theory, phenomenological research, a literature review, action research, case study, heuristic evaluation and prototyping. It is clear that an eclectic collection of research methods will be used to do this interdisciplinary research. Grounded theory underpins the approach taken in this study and as theory development is characteristic of grounded theory studies, the researcher will be able to synthesise the work done in this study in a framework that contributes to theory development. Issues relating to the credibility of the study have been considered, as well as possible ethical issues. The next chapter will consider the metadata necessary to improve retrieval of words and phrases.

4. Suggested encoding for texts

Don't use words too big for the subject. Don't say infinitely when you mean very; otherwise you'll have no word left when you want to talk about something really infinite.

C.S. Lewis

In this chapter, the researcher suggests how the encoding of texts can be done to enable retrieval on a fine-grained level. The aim of this research is to demonstrate the possibility of enhanced retrieval. It is not to generate a list of all possible items or elements that can be encoded. Therefore, further aspects that can be encoded can be identified in future research. It is important to note that the metadata schema proposed by the researcher is extensible and customisable. The extensibility of the framework makes it possible for other elements to be added and elements to be exchanged, depending on user need. For example, if it is necessary to know the translator of a work so that all work by a certain translator could be searched, then that field can be added, or if it is useful to know the style that a text was written in, then it could be encoded. This will have to be reflected in the interface design, as these fields will have to be added on the interface for the user to filter, similarly any additional fields will have to be added to the database.

However, the elements chosen here demonstrate the concept of enhanced retrieval on a detailed level. The encoding is organised according to the levels as discussed in the literature review, namely, the morphological level, syntactic level, semantic level, functional level and bibliographic level. For each level, the researcher will explain what information is encoded, as well as the format in which the encoding is done in this study.

4.1. Morphological level

It will be useful to annotate on a morphological level to allow a user to search for inflected forms or specific parts-of-speech, as is done in the Google Books Ngram Viewer as well as the corpus search tools.

On the morphological level, the lemma (to search for inflected forms) and part-of-speech for each word are encoded.

As discussed in chapter 2, the lemma is the headword (or dictionary entry) of a word. This is then the form of the word without any inflections or other changes. For example,

the lemma of *forsworn* is *forswear* or of *had* is *have*. This information is valuable and should be encoded. The researcher has followed the annotation guidelines set forth by Hardie (2014), where each word is enclosed in a TEI word (w) tag and the lemma is indicated as an attribute. For example:

```
<w lemma="forswear">forsworn</w>
```

However, to avoid redundancy, the lemma is not included if it is exactly the same as the word.

The part-of-speech category of each word is included in the annotations. The Penn Treebank tagset has been chosen to annotate the words. As mentioned in chapter 2, this is a widely used tagset and appears in many annotated corpora. It is used in some natural language processing tools, such as the tools developed by Stanford. Also following the guidelines of Hardie (2014), the part-of-speech tag for each word is added as an attribute in the TEI word tag. For example, the word *forsworn* is a past participle:

```
<w pos="VBN">forsworn</w>
```

Refer to Table 2 in chapter 2 for the list of Penn Treebank tags.

Two sentences will be used here (Figure 94) to show how the encoding can be done for complete sentences and combining the lemma and the part-of-speech tag. The two sentences are:

- “For the last three months he had forsworn letters, newspapers, and telegrams.”
- “Some people hold that breakfast is the best meal of the day.”

Each sentence is encoded with the TEI sentence tag (s) and is marked with a number (id). Each word is also marked with a number (id).


```

<s id="1">
  <tokens>
    <w id="1" lemma="for" pos="IN">For</w>
    <w id="2" pos="DT">the</w>
    <w id="3" pos="JJ">last</w>
    <w id="4" pos="CD">three</w>
    <w id="5" lemma="month" pos="NNS">months</w>
    <w id="6" pos="PRP">he</w>
    <w id="7" lemma="have" pos="VBD">had</w>
    <w id="8" lemma="forswear" pos="VBN">forsworn</w>
    <w id="9" lemma="letter" pos="NNS">letters</w>
    <w id="10" pos=",">,</w>
    <w id="11" lemma="newspaper" pos="NNS">newspapers</w>
    <w id="12" pos=",">,</w>
    <w id="13" pos="CC">and</w>
    <w id="14" lemma="telegram" pos="NNS">telegrams</w>
    <w id="15" pos=".">.</w>
  </tokens>
</s>

```

```

<s id="2">
  <tokens>
    <w id="23" pos="JJ">Some</w>
    <w id="24" pos="NN">people</w>
    <w id="25" pos="VBP">hold</w>
    <w id="26" pos="IN">that</w>
    <w id="27" pos="NN">breakfast</w>
    <w id="28" lemma="be" pos="VBZ">is</w>
    <w id="29" pos="DT">the</w>
    <w id="30" pos="JJS">best</w>
    <w id="31" pos="NN">meal</w>
    <w id="32" pos="IN">of</w>
    <w id="33" pos="DT">the</w>
    <w id="34" pos="NN">day</w>
    <w id="35" pos=".">.</w>
  </tokens>
</s>

```

Figure 94 Two sentences encoded on a morphological level

This will allow a user to do the following types of searches:

- 1) search for a word where the word is part of a specific word category
- 2) search for all inflections of a word (i.e. to search for all the inflections of *run*)

4.2. Syntactic level

It will be useful to annotate on a syntactic level. For retrieval, it seems more useful to annotate using dependency grammars as opposed to phrase structure grammars. It does not make sense to filter according to a certain phrase, as a phrase can be nested in other phrases. However, dependency grammar can be used so that a user can search for specific dependency relationships, as is shown in the Google Books Ngram Viewer.

For the syntactic level the researcher has selected to encode the dependency grammar of the sentences. Dependency grammars indicate binary relationships between words, which can be encoded, saved in a database structure and used in retrieval.

The researcher has selected the Universal Dependency annotations (version 1) to annotate the dependencies. These annotations are widely known and used, for example by the Stanford Parser. See Table 3 for the list of Universal Dependency relations. Though enhanced dependencies are available, the researcher has opted to use the basic dependencies listed by the Universal Dependency. This will be sufficient to identify head and dependent relationships to enable retrieval on this level.

An example will be used to illustrate the dependencies and how they are encoded in this research (Table 6). In the sentence: “The man shook the sleeping child by the shoulder.” the following relationships hold:

Table 6 Dependency relationships in table format

Dependent (modifies)	Head (governs)	Dependency type
The	man	det
man	shook	nsubj
shook	root	root
the	child	det
sleeping	child	amod
child	shook	doobj
by	shoulder	case
the	shoulder	det
shoulder	child	nmod
.	shook	punct

These relationships are graphically illustrated Figure 95.

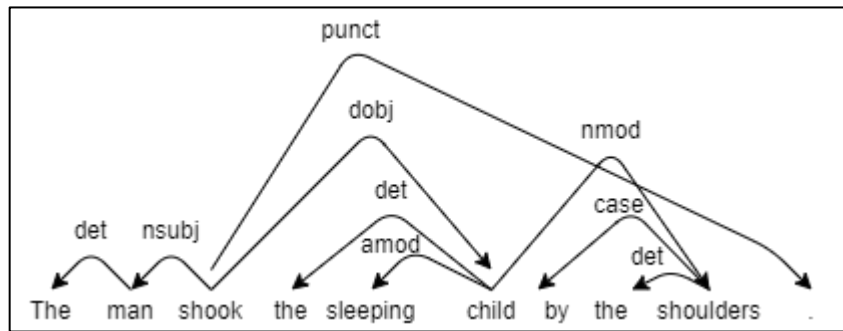


Figure 95 Graphical representation of dependencies using the Stanford CoreNLP API

The dependency relations can be encoded in XML and an example of such an encoding is shown in Figure 96. The same style of encoding used by the Stanford CoreNLP parser is used. The attribute “idx” is used to refer to the id of the word as used in the morphological level.

```

<dependencies type="basic-dependencies">
  <dep type="root">
    <governor idx="0">ROOT</governor>
    <dependent idx="3">shook</dependent>
  </dep>
  <dep type="det">
    <governor idx="2">man</governor>
    <dependent idx="1">The</dependent>
  </dep>
  <dep type="nsubj">
    <governor idx="3">shook</governor>
    <dependent idx="2">man</dependent>
  </dep>
  <dep type="det">
    <governor idx="6">child</governor>
    <dependent idx="4">the</dependent>
  </dep>
  <dep type="amod">
    <governor idx="6">child</governor>
    <dependent idx="5">sleeping</dependent>
  </dep>
  <dep type="dobj">
    <governor idx="3">shook</governor>
    <dependent idx="6">child</dependent>
  </dep>
  <dep type="case">
    <governor idx="9">shoulder</governor>
    <dependent idx="7">by</dependent>
  </dep>
  <dep type="det">
    <governor idx="9">shoulder</governor>
    <dependent idx="8">the</dependent>
  </dep>
  <dep type="nmod">
    <governor idx="6">child</governor>
    <dependent idx="9">shoulder</dependent>
  </dep>
  <dep type="punct">
    <governor idx="3">shook</governor>
    <dependent idx="10">.</dependent>
  </dep>
</dependencies>

```

Figure 96 Encoding of the dependency grammar of a sentence

Syntactic annotations will allow a user to search for a word where it modifies another word, for example, to get all the instances where *beautiful* modifies *day* (e.g. *beautiful day*; *beautiful and sunny day*; *beautiful and bright day*).

4.3. Semantic level

When filtering information, it would be useful to link each word to the most probable meaning (sense) in a dictionary, considering the context of the word.

On the semantic level the meaning of each word is encoded. As explained in chapter 2, WordNet is one of the most well-known and well-developed resources for semantic analysis. It is used for the semantic representation of words in other studies (e.g. Finlayson, 2015) where the semantics needs to be represented. The WordNet label for each word in the sentence “For the last three months he had forsworn letters, newspapers, and telegrams.” is shown in Table 7.

Table 7 WordNet labels and definitions for a sentence

Word	Synset	Definition
last	last.s.01	immediately past
three	three.n.01	the cardinal number that is the sum of one and one and one
months	calendar_month.n.01	one of the twelve divisions of the calendar year
had	have.v.01	have or possess, either in a concrete or an abstract sense
forsworn	abjure.v.01	formally reject or disavow a formerly held belief, usually under pressure
letters	letter.n.01	a written message addressed to a person or organization
newspapers	newspaper.n.01	a daily or weekly publication on folded sheets; contains news and articles and advertisements
telegrams	telegram.n.01	a message transmitted by telegraph

This information can be encoded, as is shown in Figure 97.

```

<w id="1">For</w>
<w id="2">the</w>
<w id="3" wordnet="last.s.01">last</w>
<w id="4" wordnet="three.n.01">three</w>
<w id="5" wordnet="calendar_month.n.01">months</w>
<w id="6">he</w>
<w id="7" wordnet="have.v.01">had</w>
<w id="8" wordnet="abjure.v.01">forsworn</w>
<w id="9" wordnet="letter.n.01">letters</w>
<w id="10">,</w>
<w id="11" wordnet="newspaper.n.01">newspapers</w>
<w id="12">,</w>
<w id="13">and</w>
<w id="14" wordnet="telegram.n.01">telegrams</w>
<w id="15">.</w>

```

Figure 97 Encoding of the semantic information of a sentence

Not all words have a WordNet synset. Where there is no WordNet synset, there is no wordnet attribute encoded.

Semantic metadata will allow a user to search for all instances of a word with a specific meaning. For example, if a user wanted to search for all instances of *Lord* where it is used to refer to a nobleman and not referring to the Judeo-Christian God, they can make use of the WordNet sense.

4.4. Functional level

Some information on the functional level is encoded to enhance retrieval of information.

The following structures that can be found in texts or characteristics of sections of texts will be encoded:

- text
- body
- front matter
- back matter
- heading
- paragraph
- direct speech
- quoted text
- names
- dates
- notes
- regularisation
- language (if the language of a section is different to that of the main text)
- in-text date (if the date of a section is different to that of the main text)
- in-text genre (if the genre of a section is different to that of the main text)
- in-text author (if the author of a section is different to that of the main text)

Most of these structural units or characteristics of texts can be encoded by TEI. However, the last three bulleted items cannot be encoded with standard TEI. As such, this study makes use of a customisation of TEI. In section 2.5.4, the concept of customisation in TEI was discussed.

Each functional item with the corresponding TEI element and an example is shown in Table 8.

The reason for adding these additional elements, which necessitates the creation of a customisation, should be made clear. One of the main contentions of this study is that

the properties of a section of text can differ from the bibliographic information that is used for the main text. For example, a text can mainly be in English, but can contain some Latin phrases or quotes; or the genre of a book can be prose, but a poem can be included. The researcher therefore argues that certain properties of a text should also be encoded on a fine-grained level. To some extent this can be done in TEI, but not to the extent that the researcher intends. As such, the researcher will create a customisation of TEI. The researcher suggests that information for genre, language, date of publication and author on a fine-grained level should be captured. TEI can encode language that is different to the rest of the text with the <foreign> element. However, as the other fine-grained bibliographic data cannot be captured by TEI, the researcher suggests that additional attributes to capture information for genre, date of publication and author should be used.

Some of this information could be encoded through the TEI elements for citations or references. The <ref> element is used when material is quoted, and the text explicitly mentions the source of the quote. An example is given to illustrate.

The news this morning was upsetting, but as Charles Dickens writes <quote>“if there were no bad people, there would be no good lawyers”</quote><ref>(Dickens, 1841)</ref>.

The cited quotation could also be used to indicate a quotation from some other document, together with a bibliographic reference to its source.

The researcher contends, however, that sometimes material in the text has different properties to the bibliographic details of the rest of the text that is not explicitly mentioned in the text. Similarly, sections of texts that are not necessarily quoted from other sources have different properties to the main text, for example, a preface is not quoted material, but written by a different person than the main author. As such, the researcher introduces some attributes that can be used to indicate properties that are different from the main text. This also means that others extending on this work could add their own properties.

Table 8 shows how the attributes for data on a fine-grained level can be used. A distinction between the properties of the entire text and the properties of a section in the text is maintained. A word can therefore be marked as published in 1958 (referring to the publication of the main text) and have an original publication data of 1650, for example.

Table 8 Functional encoding

Item	TEI element	Explanation	Example
Text	<text>	This element is used to identify a single text.	<text> <body>
Body	<body>	The text element minimally contains a body element. The body contains the lower level structural elements, such as paragraph. Next to the body, the front and back matter can be encoded.	<p> Lord Peter Wimsey stretched himself luxuriously between the sheets provided by the Hôtel Meurice. </p> <body> </text>
Front matter	<front>	Items that can be encoded as front matter are title pages, headers, prefaces, or dedications.	<front> <div type="contents"> <head>Table of Contents</head> <list> <item>I. "Of His Malice Aforethought"</item> <item>II. The Green-Eyed Cat</item> <item>III. Mudstains and Bloodstains</item> </list> </div> </front>
Back matter	<back>	Items that can be encoded as back matter are appendices, glossaries, notes or indexes.	<back> <div type="notes"> <head>TRANSCRIBER'S NOTES:</head> <p>To keep the flavour of the dialect, I have not made any corrections with regards to the spelling in the dialogues. However, I have made the following changes as noted. </p> </div> </back>

Heading	<head>	Headings on all levels are encoded with the <head> element.	<head>CHAPTER I</head>
Paragraph	<p>	Paragraphs can be encoded with the <p> element.	<p>Lord Peter Wimsey stretched himself luxuriously between the sheets provided by the Hôtel Meurice.</p>
Direct speech	<said>	The TEI element <said> is used to indicate spoken text. The attribute @direct is used to indicate whether it is regarded as direct speech.	<said direct="true">"Contrast,"</said> philosophised Lord Peter sleepily, <said direct="true">is life. Corsica—Paris—then London.... Good morning, Bunter."</said>
Quoted text	<quote>	This element is used to indicate that phrase is attributed some agency external to the text. (In this example the quoted speech is also direct speech.)	<quote><said direct="true">"O, Who hath done this deed?"</said></quote>
Names	<name>	This element contains a proper noun or noun phrase.	<p>Lord <name type="person">Peter Wimsey</name> stretched himself luxuriously between the sheets provided by the Hôtel Meurice.</p>
Dates	<date>	This element contains a date.	<date when="2013-04">April 2013</date>
Notes	<note>	This element is used to encode notes.	<p>... that no finite extension is capable of containing an infinite number of parts; and consequently that no finite extension is infinitely divisible <note n="1" place="foot" type="authorial">It has been objected to me, that infinite divisibility supposes.</note>.</p>
Regularisation	<choice> <orig> <reg>	These elements are used to indicate the original form of spelling as well as the regularised spelling.	In this example from the book "The Five Red Herrings" by Dorothy L. Sayers, the text "if the man wad juist be peaceable aboot it" was written to indicate a Scottish accent and the regularised spelling would read "if the man would just be peaceable about it".

			<pre> <p>...if the man <choice> <orig>wad</orig> <reg>would</reg> </choice> <choice> <orig>juist</orig> <reg>just</reg> </choice> be peaceable <choice> <orig>about</orig> <reg>about</reg> </choice> it </p> </pre>
Language attribute	foreign @xml:lang="fr"	This element is used to indicate what language the text is. It can be used in the <foreign> element.	<foreign xml:lang="fr">La Rôtisserie de la Reine Pédauque, L'Anneau d'Améthyste,</foreign> South Wind
In-text date attribute	@inTextDate	Text originally published at different date than rest of text. This is a new custom element.	<pre> <quote inTextGenre="drama" inTextAuthor="William Shakespeare" inTextDate="1603">"O, Who hath done this deed?"</quote> </pre>
In-text genre attribute	@inTextGenre	Text a different genre than rest of text. This is a new custom element.	
In-text author attribute	@inTextAuthor	Text from a different author is quoted. This is a new custom element.	

Sections from two different texts have been encoded with this TEI customisation to illustrate how the functional encoding can be done. Figure 98 shows a section from *Clouds of Witness*, a novel by Dorothy L. Sayers. Of particular interest in this example is the quote from Shakespeare at the beginning of the chapter.

```

<text>
  <body>
    <div>
      <head>CHAPTER I</head>
      <head>"OF HIS MALICE AFORETHOUGHT"</head>
      <p><quote inTextGenre="W_fict_drama" inTextAuthor="William Shakespeare"
inTextDate="1603">"O, Who hath done this deed?"</quote> Othello</p>
      <p>
        Lord <name type="person">Peter Wimsey</name> stretched himself luxuriously
        between the sheets provided by the <foreign xml:lang="fr">Hôtel Meurice
        </foreign>. After his exertions in the unravelling of the Battersea
        Mystery, he had followed Sir <name type="person">Julian Freke's</name>
        advice and taken a holiday. He had felt suddenly weary of breakfasting
        every morning before his view over the Green Park; he had realised that
        the picking up of first editions at sales afforded insufficient exercise
        for a man of thirty-three; the very crimes of London were
        over-sophisticated. He had abandoned his flat and his friends and fled to
        the wilds of Corsica. For the last three months he had forsworn letters,
        newspapers, and telegrams. He had tramped about the mountains, admiring
        from a cautious distance the wild beauty of Corsican peasant-women, and
        studying the vendetta in its natural haunt. In such conditions murder
        seemed not only reasonable, but lovable. Bunter, his confidential man and
        assistant sleuth, had nobly sacrificed his civilised habits, had let his
        master go dirty and even unshaven, and had turned his faithful camera from
        the recording of finger-prints to that of craggy scenery. It had been very
        refreshing.
      </p>
    </div>
  </body>
</text>

```

Figure 98 Example of functional encoding (*Clouds of witness*)

Figure 99 shows an extract from *The Life of St. Teresa of Jesus, of the Order of Our Lady of Carmel* (<https://www.gutenberg.org/files/8120/8120-h/8120-h.htm>). This section shows a quote in Spanish. It is also worth noting that in this case the author of the work is St. Teresa, but the introduction is by Benedict Zimmerman.

```

<body>
  <div>
    <head>Introduction to the Present Edition.</head>
    <p inTextAuthor="Benedict Zimmerman">
      When Mr. <name type="person">Lewis</name> undertook the translation of St.
      <name type="person">Teresa's</name> works,
      he had before him <name type="person">Don Vicente de la Fuente's</name>
      edition (Madrid, 1861-1862) ,
      supposed to be a faithful transcript of the original. In <date when="1873">1873
      </date> the Sociedad Foto-Tipografica-Catolica of Madrid published a
      photographic
      reproduction of the Saint's autograph in 412 pages in folio, which establishes
      the true text once for all. <name type="person">Don Vicente</name>
      prepared a transcript of this, in which he wisely adopted the modern way of
      spelling but otherwise preserved the original text,
      or at least pretended to do so, for a minute comparison between autograph and
      transcript reveals the startling fact that nearly a thousand inaccuracies
      have been allowed to creep in. Most of these variants are immaterial, but
      there are some which ought not to have been overlooked.
      Thus, in Chapter XVIII. § 20, St. <name type="person">Teresa</name>'s words
      are: <foreign xml:lang="esp">Un gran letrado de la orden
      del glorioso santo Domingo</foreign>, while <name type="person">Don Vicente
      </name> retains the old reading
      <foreign xml:lang="esp">De la orden del glorioso patriarca santo Domingo
      </foreign>. Mr. <name type="person">Lewis</name> possessed a copy of this
      photographic reproduction, but utilised it only in one instance in his second
      edition.
      <note n="1" place="foot" type="authorial">Chap. xxxiv., note 5.</note>
    </p>
  </div>

```

Figure 99 Example of functional encoding (*The life of St. Teresa*)

When a customisation is created, it is important that this customisation is formalised through a schema. According to Bauman and Flanders (2018) a TEI schema defines a vocabulary and a grammar for the customisation and provides the enforcement of the specified constraints. The schema therefore makes it possible for an encoded text to be validated.

There are various ways in which a schema can be written for a TEI customisation (TEI – Text Encoding Initiative, n.d.). For this project, the researcher used the ROMA interface provided by TEI to generate a standard schema (Mittelbach & Rahtz, 2018). This application allows a user to specify what should be in the schema through a web interface, without writing it in XML. Modules and elements can be selected or removed, and classes can be changed. The generated schema can be exported in various formats.

After creating a standard schema in ROMA, the researcher exported the schema in an XSD format. The researcher subsequently manually edited the schema (document.xsd)

in a text editor, to create bibliographic attributes and specify which elements may contain these attributes.

Figure 100 shows how the attributes to encode in-text bibliographic data are defined.

Figure 101 shows how the attributes can be added to an element, in this case the *quote* element.

```
<xs:attributeGroup name="att.inTextLevelBibliographicData.attributes">
  <xs:attributeGroup ref="tei:att.inTextLevelBibliographicData.attribute.inTextGenre"/>
  <xs:attributeGroup ref="tei:att.inTextLevelBibliographicData.attribute.inTextAuthor"/>
  <xs:attributeGroup ref="tei:att.inTextLevelBibliographicData.attribute.inTextDate"/>
</xs:attributeGroup>
<xs:attributeGroup name="att.inTextLevelBibliographicData.attribute.inTextGenre">
  <xs:attribute name="inTextGenre" type="xs:string">
    <xs:annotation>
      <xs:documentation>defines and in text genre that is different to the main text.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:attributeGroup>
<xs:attributeGroup name="att.inTextLevelBibliographicData.attribute.inTextAuthor">
  <xs:attribute name="inTextAuthor" type="xs:string">
    <xs:annotation>
      <xs:documentation>defines and in text author that is different to the main text.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:attributeGroup>
<xs:attributeGroup name="att.inTextLevelBibliographicData.attribute.inTextDate">
  <xs:attribute name="inTextDate" type="xs:string">
    <xs:annotation>
      <xs:documentation>defines and in text date that is different to the main text.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:attributeGroup>
```

Figure 100 Defining attributes in a schema

```
<xs:element name="quote">
  <xs:annotation>
    <xs:documentation>(quotation) contains a phrase or passage attributed by the narrator or author to some agency external to t
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="tei:macro.specialPara">
        <xs:attributeGroup ref="tei:att.global.attributes"/>
        <xs:attributeGroup ref="tei:att.typed.attributes"/>
        <xs:attributeGroup ref="tei:att.msExcerpt.attributes"/>
        <xs:attributeGroup ref="tei:att.notated.attributes"/>
        <xs:attributeGroup ref="tei:att.inTextLevelBibliographicData.attributes"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Figure 101 Adding attributes to an element

4.5. Bibliographic level

The importance of bibliographic metadata for retrieval of items and to understand the composition of the corpus has already been highlighted in section 2.5.5.

It is clear that there are many bibliographic schemas that could be used, and much work has been done to develop and adapt these schemas to capture information about resources. Extensive work has been done on the development of various schemas. For example, MARC is widely used to transmit data between library systems (Banerjee & Reese, 2018). Some more recent schemas, such as MODS, embrace work done in MARC and enable rich descriptions of resources (Banerjee & Reese, 2018). Dublin Core has a small set of elements, is flexible and widely used (Gilliland, 2016).

The aim of this study is not to attempt an extension or adaptation of a bibliographic schema, but to combine different types (or layers) of metadata to allow a user to retrieve words or phrases with specific properties. Furthermore, the study wishes to show that bibliographic metadata can transcend the level of the resource and can be useful for retrieval when applied to sections within a resource, as discussed in section 2.4.

If the data from the bibliographic level are combined with the data from other levels it will allow certain interesting queries to be asked, for example, how a certain author uses adjectives, or how the use of adjectives of a certain author changes over a period of time.

In order to prove that the layer of bibliographic metadata for an item can be combined with other layers of metadata (such as morphological metadata) it is not necessary to include many bibliographic elements. For the purposes of this study only a selection of bibliographic metadata elements is necessary. However, the proposed framework should then be extensible and flexible so that further bibliographic elements could be added.

The set of bibliographic metadata elements for this study has been chosen as they include some of the most common bibliographic fields and can demonstrate how retrieval on this level can be done. The following bibliographic data of the text are encoded in this study:

- Title
- Author
- Date of publication

- Publisher
- Place of publication
- Language
- Genre
- Subject
- Source of text
- Biography of author

Before discussing the elements, it is necessary to consider the extensibility of this level of metadata. As only basic bibliographic elements were selected, the standard TEI fields in the TEI were sufficient to capture this information.

The bibliographic metadata could be extended in two different ways. TEI is extensible and customisable and could be adapted to include more elements that are not currently catered for. Secondly, the TEI standard has a xenoData field that “provides a container element into which metadata in non-TEI formats may be placed” (TEI – Text Encoding Initiative, 2020). This would allow for complete bibliographic records in schemas such as Dublin Core or MODS to be added. The required information could then be retrieved from those fields.

It is recognised that there are many other metadata elements that could be of interest to humanities scholars that were not included in this set of bibliographic metadata elements. Scholars of texts could be interested in different editions, adaptations, translations and various other aspects. As the metadata framework in this study is extensible it could be customised to the needs of a certain group of scholars. For example, in this set there is only one field for date of publication. If, however, it is important for a group of scholars to have further information such as information about various editions, then that could be added to the framework.

The title refers to the title of a text and the author field captures who the creator of the work is. The information for these fields is captured as plain text. The title field has a source attribute that links to the full text of the item. The author field has a source attribute that links to more information about the author.

The publisher of the text, the date and place of publication refer to the publication information of the source text of which the digital version is a copy. The information for

these fields is captured as plain text. The primary language of the text is captured in the language field.

The genre field captures information about the kind of text. Lee (2001) argues that texts in a corpus should be classified into genres so that researchers know exactly what kind of texts they are examining. Though there are various terms used to describe categories of texts (e.g. genres, registers, text types), Lee (2001) discusses these variations in depth and suggests that there should be no objections to using the term genre to describe most of the categories used in different corpora. He also argues that there is no requirement for genres to be established literary or non-literary genres “only for them to be culturally recognisable as groupings of texts at some level of abstraction” (Lee, 2001: 52). There are different lists of genre categories that could be used. For example, the *MARC Genre Term List* includes “terms that describe general categories, or genres that may be applied to various types of information resources” and is maintained by the Library of Congress (The Library of Congress, 2017). Another genre list is the BNC Index (Lee, 2001). The genre categories in this list were chosen after examining the categories in existing corpora. These categories were “carefully selected to capture as wide a range as possible of the numerous spoken and written texts in the English language”. The BNC Index was selected for this research, as it is a comprehensive list and has been successfully applied in the BNC.

The subject field captures information regarding the topic (aboutness) of the text. The Library of Congress Subject Headings (LCSH) was selected as controlled vocabulary for this research. Much work has been done on developing and maintaining controlled vocabularies. The purpose of this study was not to evaluate different controlled vocabularies. For the two bibliographic fields *subject* and *genre*, certain controlled vocabularies were chosen to illustrate that the prototype could accommodate controlled vocabularies. The choice of a specific controlled vocabulary could be influenced by the needs of a certain target community. The prototype could be customised to accept different controlled vocabulary systems. The integration of the different levels of metadata will not be impacted by a change in controlled vocabulary.

By encoding this bibliographic data, a user will be able to search and filter according to some detail, for example to search for all texts in the database by a certain author. The encoding of these elements could easily be done in the TEI document.

Each TEI document must contain a header. The TEI header contains meta-information about the text that is encoded. The header can be likened to a library card catalogue that does not contain actual text, but information about the text.

The TEI header must contain a file description element (`fileDesc`) that describes the electronic file. An example of a TEI header is shown in Figure 102. The file description element must also contain a title statement (a statement about the electronic text), a publication statement (information about the publication of the electronic text) and also a source description (describing the source of the electronic text).

```
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>Cold Comfort Farm</title>
    </titleStmt>
    <publicationStmt>
      <publisher>Centre for Digitisation</publisher>
    </publicationStmt>
    <sourceDesc>
      <bibl>
        <title>Cold Comfort Farm </title>
        <author>Stella Gibbons</author>.
        <pubPlace>London</pubPlace>
        <publisher>Dent</publisher>
        <date>1906</date>
      </bibl>
    </sourceDesc>
  </fileDesc>
</teiHeader>
```

Figure 102 An example of a TEI header

For this study the bibliographic information is encoded in the TEI header. It is worth noting that the TEI header can accommodate metadata from formal standards. For example, if a library has created rich descriptions of items using the MODS standard it can be included in the TEI header. The TEI header includes a `<xenData>` element, in which metadata in non-TEI formats can be added. Figure 103 shows the MODS record for *Clouds of Witness* by Dorothy L. Sayers as included in a TEI header.

```

<mods:mods>
  <mods:titleInfo>
    <mods:title>Clouds of Witness</mods:title>
  </mods:titleInfo>
  <mods:name type="personal">
    <mods:namePart>Dorothy L. Sayers</mods:namePart>
    <mods:role>
      <mods:roleTerm type="code" authority="marcrelator">aut</mods:roleTerm>
      <mods:roleTerm type="text" authority="marcrelator">Author</mods:roleTerm>
    </mods:role>
  </mods:name>
  <mods:typeOfResource>text</mods:typeOfResource>
  <mods:originInfo>
    <mods:place>
      <mods:placeTerm type="text">London</mods:placeTerm>
    </mods:place>
    <mods:publisher>Victor Gollancz Lt</mods:publisher>
    <mods:dateIssued>1958</mods:dateIssued>
  </mods:originInfo>
  <mods:language>
    <mods:languageTerm authority="iso639-2b">eng</mods:languageTerm>
    <mods:languageTerm type="text">English</mods:languageTerm>
  </mods:language>
  <mods:subject authority="lesh">
    <mods:topic>Detective and mystery fiction</mods:topic>
  </mods:subject>
  <mods:genre authority="marcgt">book</mods:genre>
  <mods:genre authority="marcgt">fiction</mods:genre>
  <mods:genre authority="marcgt">novel</mods:genre>
  <mods:relatedItem type="host">
    <mods:titleInfo>
      <mods:title>University of Pretoria</mods:title>
    </mods:titleInfo>
  </mods:relatedItem>
</mods:mods>

```

Figure 103 Example of a MODS record created as demonstration

It would be useful to be able to capture metadata in a specific standard if the texts that were being encoded were already described by such metadata. For example, if texts from a library were received and these texts had rich descriptions in MODS associated with them, then it would be beneficial to keep these descriptions. This research will not make use of the additional xenoData element for bibliographic descriptions in other formats, but in future work it could be included if texts with existing rich descriptions were being used.

Table 9 shows the bibliographic data to be encoded, the TEI element that is used for the encoding, as well as an example.

Table 9 Bibliographic data in TEI

Bibliographic data	TEI element	Example
Title	<title>	<title>Clouds of Witness</title>
Author	<author>	<author>Dorothy L. Sayers</author>
Date of publication	<date>	<date>1958</date>
Publisher	<publisher>	<publisher> Victor Gollancz Lt </publisher>
Place of publication	<pubPlace>	<pubPlace>London</pubPlace>
Genre	<keywords> <terms>	<keywords scheme="BNCIndex"> <term>W_fict_prose</term> </keywords>
Subject	<keywords> <terms>	<keywords scheme="http://id.loc.gov/authorities/about.html#lcsht"> <term>Detective and mystery fiction</term> </keywords>
Language	<language>	<language ident="en">English</language>
Source of text	@source (The source attribute of the title tag)	<title source="https://gutenberg.ca/ebooks/sayers-clouds/sayers-clouds-00-h-dir/sayers-clouds-00-h.html">Clouds of Witness</title>
Biography of author	@source (The source attribute of the author tag)	<author source="https://en.wikipedia.org/wiki/Dorothy_L._Sayers">Dorothy L. Sayers</author>

4.6. Combined example

In this section the metadata from all levels will be combined.

The ideal situation would be to have all metadata in one file. Such a file can be used by a system to filter according to a user's requirements. However, as the researcher studied the different levels of encoding, it became apparent that it would become unnecessarily complex to include the encoding of all levels in one file.

Adding all metadata to one file will become very difficult for a human to read and modify. It will not be a problem for a computer, but, since at this stage much of the encoding will have to be done manually it has to be understandable to a human to some extent.

This is especially evident with the encoding of the dependency grammar on the syntactic level. The data are stored per sentence but do not use the exact structure of the sentence as it is written. This will become particularly confusing if stored with the functional encoding.

As such, the researcher suggests that two files of each text are created, each containing a layer of encoding. The first document should contain the first three levels of encoding, namely the morphological, syntactic and semantic levels. The second document should contain the last two levels of encoding, namely, the functional and bibliographic levels. Furthermore, the encoding for sentences should be included in the first file, in order to make the second file clearer for other functional encoding. Figure 104 shows an example of the first layer of encoding and Figure 105 shows an example of the second layer.

```

<?xml version="1.0" encoding="UTF-8"?>
<text>
  <sentences>
    ...
      <s id="25">
        <tokens>
          <w id="1" lemma="he" pos="PRP">He</w>
          <w id="2" lemma="blink" pos="VBD"
wordnet="blink.v.01">blinked</w>
          <w id="3" pos="IN">at</w>
          <w id="4" pos="DT">the</w>
          <w id="5" pos="NN" wordnet="sunlight.n.01">sunlight</w>
          <w id="6" pos=".">.</w>
        </tokens>
        <dependencies type="basic-dependencies">
          <dep type="root">
            <governor idx="0">ROOT</governor>
            <dependent idx="2">blinked</dependent>
          </dep>
          <dep type="nsubj">
            <governor idx="2">blinked</governor>
            <dependent idx="1">He</dependent>
          </dep>
          <dep type="case">
            <governor idx="5">sunlight</governor>
            <dependent idx="3">at</dependent>
          </dep>
          <dep type="det">
            <governor idx="5">sunlight</governor>
            <dependent idx="4">the</dependent>
          </dep>
          <dep type="nmod">
            <governor idx="2">blinked</governor>
            <dependent idx="5">sunlight</dependent>
          </dep>
          <dep type="punct">
            <governor idx="2">blinked</governor>
            <dependent idx="6">.</dependent>
          </dep>
        </dependencies>
      </s>
    ...
  </sentences>
</text>

```

Figure 104 First layer of encoding

```

<TEI xmlns="http://www.tei-c.org/ns/1.0"
xmlns:mods="http://www.loc.gov/mods/v3">
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>Clouds of Witness</title>
      <author>Dorothy L. Sayers</author>
      <respStmt>
        <resp>Encoded by</resp>
        <name>Liezl Ball</name>
      </respStmt>
    </titleStmt>
    <publicationStmt>
      <publisher>University of Pretoria</publisher>
      <pubPlace>Pretoria</pubPlace>
      <date>2019</date>
    </publicationStmt>
    <sourceDesc>
      <bibl>
        <title
source="https://gutenberg....clouds.html">Clouds of Witness</title>
        <author
source="https://en.wikipedia...Sayers">Dorothy L. Sayers</author>
        <date>1958</date>
        <publisher>Victor Gollancz Lt</publisher>
        <pubPlace>London</pubPlace>
      </bibl>
    </sourceDesc>
  </fileDesc>
  <profileDesc>
    <langUsage>
      <language ident="en">English</language>
    </langUsage>
    <textClass>
      <keywords
scheme="http://id.loc.gov/authorities/about.html#lcsch">
        <term>Detective and mystery fiction</term>
      </keywords>
      <keywords scheme="BNC_index">
        <term>W_fict_prose</term>
      </keywords>
    </textClass>
  </profileDesc>
</teiHeader>
<text>
<body>
  <div>
    <head>CHAPTER I</head>
    ...
    <p><said direct="true">"Thanks,"</said> said Lord <name
type="person">Peter</name>. He blinked at the sunlight.
    </p>
    ...
  </body>
</text>
</TEI>

```

Figure 105 Second layer of encoding

From these two files, information for filtering could be extracted by a search system. The next step is to design a system that can process these two files and store the necessary information in a database. The database will then store the relevant information for each word. For each word, the following information will be available:

General

- The document to which it belongs
- The sentence to which it belongs
- The location in the sentence
- The word itself (as it appears in the text)

Morphological data

- The lemma
- The part-of-speech tag

Syntactic data

- Governor (to determine dependency relationships)
- Dependency type

Semantic data

- WordNet sense

Functional level

- Body
- Front matter
- Back matter
- Heading
- Paragraph
- Direct speech
- Quoted text
- Name
- Date
- Note
- Regularised word
- In-text author
- In-text date
- In-text genre
- In-text language

Bibliographic data

- Title
- Author
- Date of publication
- Publisher
- Place of publication

- Language
- Genre
- Subject
- Source of text
- Biography of author

Table 10 is used to give an overview of this suggestion. This table shows the information captured for each word, and where the information would be found in the encoded documents. Examples are also given.

The first example word, namely *chapter*, is taken from the sentence *Chapter 1* from the novel *Clouds of Witness* by Dorothy L. Sayers, encoded as a heading.

```
<head>CHAPTER I</head>
```

The second example word, namely *deed*, is taken from the sentence *O, Who hath done this deed?* from the novel *Clouds of Witness* by Dorothy L. Sayers, encoded as a quote.

```
<p><quote inTextGenre="W_fict_drama" inTextAuthor="William Shakespear" inTextDate="1603"><said>"O, Who hath done this deed?"</said></quote> Othello</p>.
```


Table 10 Data stored for each word

Data	Location in encoded documents	Example 1 (chapter)	Example 2 (deed)
Word ID	book1_layer1.xml/text/sentences/s/tokens/w @id	1	13
Sentence ID	book1_layer1.xml/text/sentences/s @id	1	3
Document ID	generated	1	1
Word	book1_layer1.xml/text/sentences/s/tokens/w	CHAPTER	deed
Lemma	book1_layer1.xml/text/sentences/s/tokens/w @lemma	chapter	deed
POS	book1_layer1.xml/text/sentences/s/tokens/w @pos	NN	NN
Regularised original	book1_layer2.xml/TEI/text/.../choice/orig		
Governer	book1_layer1.xml/text/sentences/s/dependencies/dep/governor where .../dep/dependent @idx=word ID	I	done
Dependency type	book1_layer1.xml/text/sentences/s/dependencies/dep @type where .../dep/@idx=word ID	compound	dobj
Sense	book1_layer1.xml/text/sentences/s/tokens/w @wordnet	chapter.n.01	act.n.02
Front matter	book1_layer2.xml/TEI/text/.../front	FALSE	FALSE
Back matter	book1_layer2.xml/TEI/text/.../back	FALSE	FALSE
Body	book1_layer2.xml/TEI/text/.../body	TRUE	TRUE
Paragraph	book1_layer2.xml/TEI/text/.../p	FALSE	TRUE
Heading	book1_layer2.xml/TEI/text/.../head	TRUE	FALSE
Direct speech	book1_layer2.xml/TEI/text/.../said	FALSE	TRUE
Name	book1_layer2.xml/TEI/text/.../name	FALSE	FALSE
Date	book1_layer2.xml/TEI/text/.../date	FALSE	FALSE
Note	book1_layer2.xml/TEI/text/.../note	FALSE	FALSE
Title	book1_layer2.xml/TEI/teiHeader/fileDesc/sourceDesc/bibl/title	Clouds of witness	Clouds of witness
Author *	book1_layer2.xml/TEI/teiHeader/fileDesc/sourceDesc/bibl/author	Doroty L. Sayers	Doroty L. Sayers
Date published	book1_layer2.xml/TEI/teiHeader/fileDesc/sourceDesc/bibl/date	1985	1985
Publisher	book1_layer2.xml/TEI/teiHeader/fileDesc/sourceDesc/bibl/publisher	Victor Gollancz Lt	Victor Gollancz Lt

Place of publication	book1_layer2.xml/TEI/teiHeader/fileDesc/sourceDesc/bibl/pubPlace	London	Britain
Genre *	book1_layer2.xml/TEI/teiHeader/profileDesc/textClass/keywords/terms WHERE keywords @ scheme=BNC_index	W_fict_prose	W_fict_prose
Subject	book1_layer2.xml/TEI/teiHeader/profileDesc/textClass/keywords/terms WHERE keywords @ scheme=http://id.loc.gov/authorities/about.html#lcs	Detective and mystery fiction	Detective and mystery fiction
Language	book1_layer2.xml/TEI/teiHeader/profileDesc/langUsage/language	eng	English
Source	book1_layer2.xml/TEI/teiHeader/fileDesc/sourceDesc/bibl/title @source	https://gutenberg.ca/ ... clouds-00-h.html	https://gutenberg.ca/ ... clouds-00-h.html
Author biography	book1_layer2.xml/TEI/teiHeader/fileDesc/sourceDesc/bibl/author @source	https://en.wikipedia ... Sayers	https://en.wikipedia ... Sayers
In-text author	book1_layer2.xml/TEI/text/...@inTextAuthor		William Shakespeare
In-text date	book1_layer2.xml/TEI/text/...@inTextDate		1603
In-text genre	book1_layer2.xml/TEI/text/...@inTextGenre		W_fict_drama
In-text language	book1_layer2.xml/TEI/text/.../foreign @ lang		

From this table it should be clear that if such information for each word were to be stored, a user could easily filter to find exactly the instances that comply with certain criteria. For example, a filter could be applied to only show words where direct speech is TRUE and then within that set to search for specific words. However, it should also be evident that a large amount of redundant information is stored. As such, the technical implementation should address this.

4.7. Conclusion

In chapter 2, metadata that can be used to describe texts, features of texts and properties of words in texts were discussed. It was clear that there is a vast amount of metadata that can be applied to texts and words. From metadata about the text itself (e.g. information about the author and publisher) to metadata about words in the text (e.g. part-of-speech categories) were considered. The metadata were divided into five categories, namely morphological, syntactic, semantic, functional and bibliographic. In chapter 2, various ways in which the metadata could be made explicit through encoding were also considered. Furthermore, ways in which tools use this metadata to enable users to retrieve specific words or phrases were considered. It became apparent that current tools do not effectively support retrieval of words or phrases on a fine-grained level. It was found that if the detailed metadata are made explicit (encoded) it can be useful for retrieval and that work in this area is necessary. In concluding chapter 2, it was found that metadata that can enhance retrieval should be identified, applied and tested.

In this chapter, such metadata were identified and discussed. Metadata for each of the five categories identified in chapter 2 were suggested. This chapter also considers the way in which the metadata can be encoded and which standards and conventions are suitable. Furthermore, a suggestion was made as to how all the metadata can practically be applied in different files so that all metadata can be included.

By encoding texts with the suggested metadata, powerful filtering and search options become possible. In order for the search to work, the metadata from the two files that are created in the encoding process have to be combined and stored in a database. This will be discussed in chapter 6. The next chapter will first look at various texts that were selected and encoded for this study.

5. Encoding of sample texts to improve retrieval

The twinkling of an eye. That is the most wonderful expression. I've thought from time to time it was the best thing in life, that little incandescence you see in people when the charm of a thing strikes them, or the humor of it.

Gilead by Marilynne Robinson

5.1. Introduction

This study aims to determine if the retrieval of words and phrases can be improved through enhanced metadata. In the previous chapter the researcher suggested encoding metadata on five different levels in order to make detailed information of a text explicit. In order to determine if this suggested encoding indeed improves retrieval, a prototype was developed to test the retrieval based on texts encoded in the suggested manner. The encoded texts used in this prototype will be discussed in this chapter.

5.2. Texts selected for encoding

In order to test that all the elements in the encoding scheme suggested in this study can be retrieved, text samples that have examples of all the elements in the encoding scheme had to be chosen. For example, in order to test that text that is direct speech can be retrieved, a text sample with direct speech had to be encoded.

Five different texts were selected for the purposes of this study. Texts that contained features that needed to be encoded and tested for retrieval were selected. Sections of these text with interesting features were sampled for encoding.

As was explained in chapter 3, the purpose of a prototype is not to be a fully-fledged system, but to test a concept. It is therefore only necessary to encode a sufficient number of texts to test all the features of the system. Encoding a very large number of texts will not add to the purpose of this study. If all the elements suggested in the scheme have been encoded and can be retrieved, then an evaluation can be made. Encoding more texts will not benefit this study further. The same argument applies to encoding a whole text or only a section of a text. Small samples from several texts can provide the necessary data for the evaluation of this prototype.

The texts were obtained from the website of the Project Gutenberg (<https://www.gutenberg.org/> and <http://gutenberg.ca/>). The texts through this project are made available with a *caveat* that readers must check the copyright terms of their own

countries. In this study only a small percentage of each text was selected for encoding and results are displayed as quotes. The project is also a research project. As such, even if there was an uncertainty about copyright restrictions, the use of the samples for research purposes should fall under the terms of fair dealing/use (Band & Gerafi, 2015).

The five texts that were selected for this study are listed in Table 11. The table indicates the date the text was published, the title, author and the last column indicates the number of words from each text that was encoded. Some interesting features of each text are subsequently discussed.

Table 11 Texts selected for encoding

Nr.	Date	Title	Author	Sample size (words)
1.	1813	Pride and Prejudice	Jane Austen	468
2.	1880	Ben Hur: A Tale of the Christ	Lew Wallace	566
3.	1904	The Life of St. Teresa of Jesus, of the Order of Our Lady of Carmel	Teresa of Avila	901
4.	1919	My man Jeeves	P.G. Wodehouse	495
5.	1958	Clouds of Witness	Dorothy L. Sayers	1119
Total words				3549

The sample selected from the novel *Pride and Prejudice* has many examples of direct speech.

The sample from the novel *Ben Hur: A Tale of the Christ* was particularly useful to demonstrate quoted text and the use of additional bibliographic elements, that is the author of a quote that is different to the author of the main text. In order to include many different quotes, the sample from the novel *Ben Hur: A Tale of the Christ* does not contain one continuous section of text from the novel. Various samples from this novel were selected to create the final sample. This sample includes various quotes that were used in the novel, particularly at the beginning of a section. There are quotes from Shakespeare, Wordsworth, Schiller and more.

The third text, *The Life of St. Teresa of Jesus, of the Order of Our Lady of Carmel*, has many different interesting aspects. It includes a title page, table of contents and an index, which meant that the front matter and back matter could be encoded. It includes an introduction that is not written by the author of the text, which meant the additional bibliographic data could be encoded. The sample includes Latin and Spanish quotes, which meant that the use of a foreign language in a text could be encoded. The sample also includes some footnotes.

The fourth example, *My man Jeeves*, is a text that is written in a very informal style and made the semantic encoding interesting.

The novel, *Clouds of Witness*, was selected as it had many of the features pertinent to this study. There are quotes from external sources, there are examples of direct speech, some French words, and headings of chapters.

In Table 12, each level of encoding, with the elements in each level, is listed and whether it was encoded in each text.

Table 12 Possible encodings of each text

Level	Elements	[1] <i>Pride and ...</i>	[2] <i>Ben Hur: A ...</i>	[3] <i>The Life of ...</i>	[4] <i>My man ...</i>	[5] <i>Clouds of ...</i>
Morphological level	Lemma	Yes	Yes	Yes	Yes	Yes
	Part-of-speech	Yes	Yes	Yes	Yes	Yes
Syntactic level	Syntactic dependencies	Yes	Yes	Yes	Yes	Yes
Semantic level	WordNet sense	Yes	Yes	Yes	Yes	Yes
Functional level	Text	Yes	Yes	Yes	Yes	Yes
	Body	Yes	Yes	Yes	Yes	Yes
	Front matter	No	No	Yes	Yes	No
	Back matter	No	No	Yes	No	No
	Heading	Yes	Yes	Yes	Yes	Yes
	Paragraph	Yes	Yes	Yes	Yes	Yes
	Poems	No	No	No	No	Yes
	Direct speech	Yes	No	Yes	Yes	Yes
	Quoted text	No	Yes	Yes	No	Yes
	Names	Yes	Yes	Yes	Yes	Yes
	Dates	No	No	Yes	No	No
	Notes	No	No	Yes	No	No
	Regularisation	No	No	No	No	Yes
	Language attribute	No	No	Yes	No	Yes
	Date attribute	No	Yes	Yes	No	Yes
	Genre attribute	No	Yes	Yes	No	Yes
Author attribute	No	Yes	Yes	No	Yes	
Bibliographic level	Title	Yes	Yes	Yes	Yes	Yes
	Author	Yes	Yes	Yes	Yes	Yes
	Date of publication	Yes	Yes	Yes	Yes	Yes
	Publisher	Yes	Yes	Yes	Yes	Yes
	Place of publication	Yes	Yes	Yes	Yes	Yes
	Genre	Yes	Yes	Yes	Yes	Yes
	Subject	Yes	Yes	Yes	Yes	Yes
	Language	Yes	Yes	Yes	Yes	Yes
	Source of text	Yes	Yes	Yes	Yes	Yes
Biography of author	Yes	Yes	Yes	Yes	Yes	

5.3. Encoding process

In this section the researcher will explain how the encoding was done for each level.

5.3.1. Morphological level

For the morphological level, the different words (tokens) in the texts have to be identified and a lemma and part-of-speech category have to be assigned for each word. Sentences are also identified on this level.

As was discussed in chapter 2, much progress has been made on using software to automate the identification of words and their associated lemma and part-of-speech category. In chapter 7 some of the software available for automated encoding will be evaluated; however, the quality of these tools is such that automated encoding could be used to some extent in this study. This chapter will therefore not discuss the quality of the encoding, but will explain how the tools were used in this study. Software was used for the initial encoding on this level; afterwards it was checked manually.

The Stanford CoreNLP library was used in this study for this initial encoding. The performance is relatively good and provides the basis for the rest of the encoding.

The encoding produced by the library was then manually checked by the researcher and errors were corrected. The library managed to separate words very well and almost no corrections had to be done. The identification of sentences was more problematic. Standard sentences in a continuous stretch of text were easily identified, but headings, quotes, and the use of abbreviations were sometimes encoded incorrectly and had to be corrected manually. In most cases the correct lemma was identified and assigned. The part-of-speech categories assigned to words were also mostly correct, but still had to be checked and corrected in some places.

This process can be described as semi-automated encoding.

5.3.2. Syntactic level

On the syntactic level the dependencies have to be identified, in other words the governor and dependent and the type of dependency.

A similar approach as for the morphological level was followed.

The Stanford CoreNLP library was used in this study for the encoding of this level. The performance of the library will be discussed in depth in chapter 7, but the application of the library in this study will be discussed here.

The performance of the library was sufficient to start the encoding process. The morphological encoding and the syntactic encoding are done at the same time by the library and the results are stored in the same document. If there were errors on the morphological level, for example an incorrect part-of-speech category was assigned, then the syntactic encoding could be affected; this was then checked and corrected.

The encoding for this level can be regarded as a semi-automated.

5.3.3. Semantic level

On the semantic level the correct sense from the WordNet database has to be assigned to each word.

Though there is progress being made with the development of tools to determine the meaning of words, as will be discussed in chapter 7, the performance of the tool available was not such that it was useful to this study.

The encoding on this level was completely manual. For each noun, verb, adjective or adverb, the available senses in the WordNet database were checked and the correct sense assigned.

The correction of the morphological and syntactic encoding and the assignment of the correct sense happened at the same time. The data for the first three levels of encoding are stored in the same file.

5.3.4. Functional level

The structure of the text is encoded on this level. The encoding on this level is done manually. The data for the functional and bibliographic encoding are stored in the second file. A text editor was used to create the TEI encoded document. The different textual features, for example headings and paragraphs, were identified and encoded. In the case of quotes, external sources sometimes had to be consulted, for example, to check the source of a quote. The TEI encoded document was then validated against the schema to ensure that there were no errors in the document.

5.3.5. Bibliographic level

The bibliographic data were encoded manually. The data were stored in the header of the TEI encoded document. Most of the bibliographic data could be obtained from the information on Project Gutenberg. However, some data had to be checked on other information sources.

5.4. Examples

Due to space restrictions the full set of encoded data is not included in the thesis. However, an example for each element is given in Table 13.

Table 13 Examples of encoding in texts

Level	Elements	Example	Source text
Morphological level	Lemma	<w id="29" lemma="extend" ...>extending</w>	[2]
	Part-of-speech	<w id="17" pos="NN" ...>fortune</w>	[1]
Syntactic level	Syntactic dependencies	<dep type="amod"> <governor idx="15">beauty</governor> <dependent idx="14">wild</dependent> </dep>	[5]
Semantic level	WordNet sense	<w id="54" wordnet="mind.n.01" ...>head</w>	[4]
Functional level	Text	<text>	[1]
	Body	<body> ... </body> </text>	
	Front matter	<front> <div type="contents">...</div> </front>	[4]
	Back matter	<back> <div type="index">...</div> </back>	[3]
	Heading	<head>Chapter 1</head>	[1]
	Paragraph	<p>It happened after...</p>	[4]
	Direct speech	<said direct="true">"What is his name?"</said>	[1]
	Names	<name type="person">Teresa</name>	[3]
	Dates	<date when="1520">1520</date>	[3]
	Notes	<note n="2" place="foot" type="authorial">Chap. xviii. ... </note>	[3]
	Regularisation	<choice><orig>mediæval</orig><reg>medieval</reg></choice>	[5]
	Language attribute	<foreign xml:lang="la">communicare ...</foreign>	[3]
	Quoted text	<quote inTextGenre="W_fict_poetry" inTextAuthor="William Wordsworth"	[2]
	Date attribute	inTextDate="1807">	
	Genre attribute	"And, through the heat of conflict, keeps the law,	

	Author attribute	In calmness made, and sees what he foresaw." </quote>	
Bibliographic level	Title	<title source="https://www.gutenberg.org/files/1342/1342-h/1342-h.htm">Pride and Prejudice</title>	[1]
	Source of text		
	Author	<author source="https://en.wikipedia.org/wiki/Jane_Austen">Jane Austen</author>	
	Biography of author		
	Date of publication	<date>1813</date>	
	Publisher	<publisher>T. Egerton</publisher>	
	Place of publication	<pubPlace>Whitehall</pubPlace>	
	Genre	<keywords scheme="BNC_index"> <term>W_fict_prose</term> </keywords>	
	Subject	<keywords scheme="http://id.loc.gov/authorities/about.html#lcsch"> <term>Social classes--Fiction</term> </keywords>	
	Language	<langUsage> <language ident="en">English</language> </langUsage>	

5.5. Conclusion

In order to test the retrieval of words and phrases according to certain properties, using the prototype developed for this study, encoded texts are necessary. In this chapter the texts that were selected for encoding were noted and the reasons for their selection were given. The process of encoding for each level was explained and an example of each element that was encoded was given. The encoding makes information about the words and texts explicit and can be used for retrieval. The retrieval of words according to this information (or properties) will be discussed in the next chapter.

6. Prototype to test retrieval of encoded texts

...and he cried: 'That was the word I wanted; you have said the word.'

The Innocence of Father Brown by G.K. Chesterton

6.1. Introduction

In this chapter, the prototype developed to test the retrieval of encoded texts will be discussed. The first section (6.2) will consider the implementation of the prototype. The purpose of this discussion is to give the reader a basic understanding of how the tool works. The purpose is not to give a detailed technical discussion, including all the algorithms, classes and functions that were used to develop this prototype, as it is deemed outside the scope of this study. By excluding some of the detail, the explanation should be succinct and clear. In the following section (6.3), the way in which the prototype works will be discussed, using examples as illustrations.

6.2. Implementation

A combination of languages and technologies were used to develop the tool. These are Node.js, MySQL, Javascript, HTML and CSS. Node.js is a Javascript runtime environment that allows scripts to be run outside the browser environment. It is free, open-source and is compatible across platforms (<https://nodejs.dev/>). This was used to create the back end (including the structure, processing, and algorithms) of the system. The technology used for the database is MySQL, a widely used open-source database (<https://www.mysql.com/>). The interface was written in Javascript, HTML and CSS. Javascript is a scripting language used in web pages, HTML is a markup language used to specify the display of pages on the web and CSS is used to describe the style of web pages (<https://www.w3schools.com/>).

The decision was made to process the metadata encoded in XML and save the data from the XML into a relational database. The reason for this was to make the querying of the data more efficient. Querying in XML can be cumbersome and slow. Furthermore, the encoding had to be done in two files to accommodate all the data, as was explained in chapter 4. This in effect made querying the XML files directly almost impossible. A relational structure was created to incorporate all the information from the two encoded files into one database. The relation database technology chosen was MySQL, as it has a powerful query functionality and performs well. Node.js was selected as language for implementation. One of the main advantages of using Node.js

is that the language used to specify the structure of the program and the language used for the interface is the same.

The tool works as follows. Two XML encoded files are submitted to the tool for each document that is added to the database. The first file, henceforth referred to as *file A*, contains the first three levels of encoding, namely, the morphologic, syntactic and semantic levels. The second file, henceforth referred to as *file B*, contains the next two levels of encoding, namely, the functional and bibliographic levels. The structure and content of these files are discussed in chapter 4.

The two files are sent to the tool and have to be parsed. Firstly, the bibliographic elements from file B are used to create a unique document in the database. In other words, the title, author, publisher and other bibliographic information, is taken from the header of the file B and used to create the unique document. The document will also have a unique identifier. For example, the text document *Pride and Prejudice* from Jane Austen should be added to the tool. The two files (A and B) that were used to encode this text are sent to the tool. The tool reads the bibliographic metadata from file B (e.g. title = *Pride and Prejudice*, author = Jane Austen) and uses this to create a unique document in the database. A unique ID (e.g. 3) is assigned to this document.

Next, file A is parsed, and each sentence in the file is linked to the document to which it belongs. Each sentence in the document has a unique number. Then, each word is seen as a token and each token is assigned to the sentence to which it belongs. Each token in a sentence has a unique number. For example, assume the sentence *It is a truth universally acknowledged...* from *Pride and Prejudice* is being parsed. Then, token 4, *truth*, belongs to sentence 2, *It is a truth universally acknowledged...*, which belongs to document 3, *Pride and Prejudice*. This is illustrated in Figure 106.

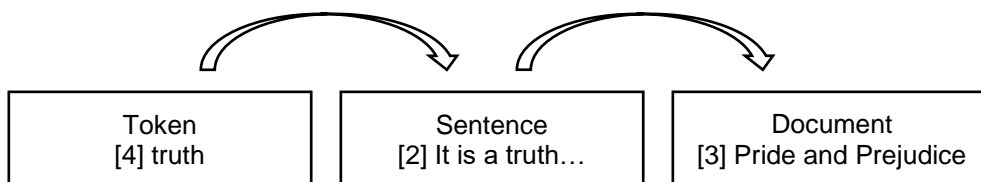


Figure 106 Linking of tokens, sentences and documents

Apart from tokens, there is a table (list) of words in the database. Each token is linked to a word. As the file is parsed, each token is compared to the words already in the database. If a word that is the same as the token being tested already exists in the database, it is not added, otherwise a new word is added. This means that duplicate words are not stored in the database.

To illustrate, a simple example will be used. Assume the list of words already in the database is *word1*, *word2*, *word3*. The next token being processed is *word4*. The code checks to see if *word4* already exists in the database. It does not, and it is added. Assume the next token being processed is *word2*. The code checks to see if *word2* already exists in the database. It does and is therefore not added, but a link from the token to *word2* is created.

In the next step the properties of each token are stored. This is the part-of-speech category, lemma, syntactic dependencies, and sense. At this point, the code also checks to see if there are capital letters used in the word, and if so, a version is stored where all letters are in lower case. This enables case-sensitive searching. All these properties are stored in the database and linked to the token.

The code then moves back to file B. The bibliographic information in the header has already been saved in the database to create the unique document. Now, the functional metadata have to be stored in the database.

Here, the tool relies on the information that it has stored in the database from file A. This enables the tool to know what word it is expecting next in file B. For example, if the first word that was found in file A, is *Chapter*, then the first word in the text in file B, should be *Chapter*. This matching is crucial, as the tool has to identify which token, that is already saved in the database, is being processed here, so that additional functional properties can be linked to that token.

The encoding in Figure 107 will be used to illustrate the process that is followed.

```
<body>
  <head>Chapter 1</head>
  <p>It is a truth universally acknowledged,
```

Figure 107 Sample of encoding to illustrate parsing

The tool proceeds through the XML tree and at each point determines if the value that it has found is an element node (e.g. <head>) or a text node (e.g. Chapter). The first element that is found is made the parent. The parent is then linked to the document. In this example the first element is <body>. In other words, <body> is linked to the document that we are working with and is made the parent. The tool proceeds through the encoded file. The next value that is found is an element node, <head>. This value is now made a child of the current parent (<body>), and a hierarchical structure is

formed. The current element is made the new parent. Therefore, the new parent is <head>.

The tool proceeds. The next value that is found is a text node. The whole string is found, in other words *Chapter 1*, not just the next token, which is *Chapter*.

Here the tool works with the concept of a cursor that moves through the text string. The tool checks to see if the value that is expected (token that was found in file A) is at the start of the text string that is being examined. For example, the first token that was identified in file A was *Chapter*. This means the first text (not XML elements) that is expected in file B is *Chapter*. If this is the case, the token that was saved in the database (linked to a sentence, linked to a document) is also linked to the functional hierarchy as a child.

The tool will proceed through the text string to find each next token and link it to the hierarchy.

This hierarchical structure enables the tool to allow detailed filtering. If a user specifies that they are looking for instances of *Chapter* that appear in a heading, the tool will find the word chapter in the database, then, for each instance, it will traverse through the hierarchy to see if one of the ancestors of this instance is a heading.

This hierarchical structure is also important for the additional bibliographic properties. As was explained earlier, the bibliographic properties from file B are used to create a unique document. As the tool parses the rest of file B, the functional elements are linked to the document. Thus, the root (or top) of the hierarchical structure is the document, with the document-level bibliographic properties. If no new bibliographic properties are added, all children will inherit the bibliographic properties of the root. However, if an element that has additional bibliographic properties is added to the tree, these new properties will override the properties of the root, from that point in the tree. This means that the tool can find an instance (token) and traverse the tree from that instance and examine each parent in the hierarchy and see if there are any bibliographic properties associated with any of the parents. If document node (root) is reached and no other bibliographic properties have been found, the instance has the same bibliographic properties as the document. However, if one of the parents has associated bibliographic properties then the instance has that in-text bibliographic data.

An example will be used to illustrate this concept. Consider the text in Figure 108.


```

<body>
  <head>BOOK SECOND</head>
  <p>
    <quote inTextGenre="W_fict_poetry" inTextAuthor="Lord Byron" inTextDate="1812">
      "There is a fire..."
    </quote>
  </p>
  <p>
    <quote inTextGenre="W_fict_drama" inTextAuthor="William Shakespeare" inTextDate="1609">
      "Doubt though the stars are fire;"
    </quote>
  </p>
  <p>
    The fire gave warmth to the whole room. She was reading a book.
  </p>
</body>

```

Figure 108 Comprehensive example of encoding to illustrate parsing

There are two instances of the word *book*, and three instances of the word *fire*. The first instance of *book* is in the heading and the second is in the last paragraph. The first instance of *fire* is in a quote by Lord Byron, the next is in a quote by Shakespeare and the next instance of *fire* is merely in a paragraph.

The hierarchical structure for the instances of *book* and *fire* are shown in Figure 109.

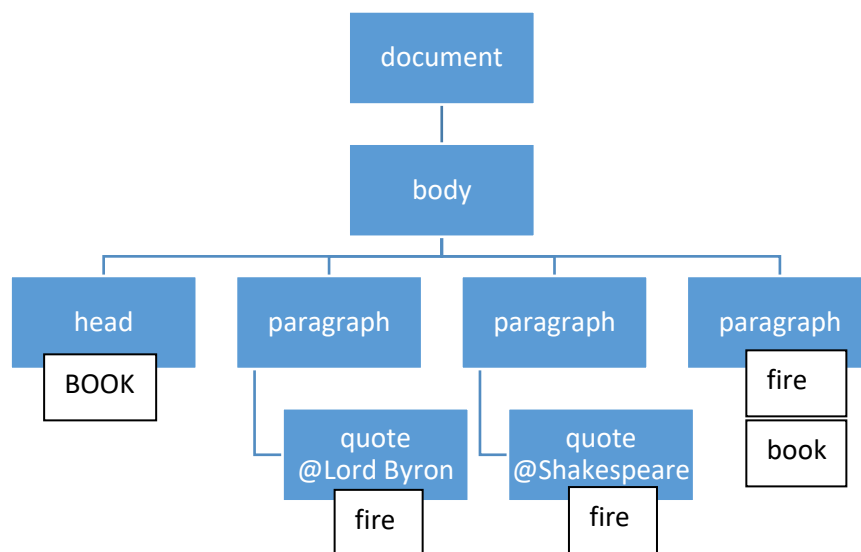


Figure 109 Hierarchical structure

Assume a user is looking for instances of the word *book* (case ignored) where it appears in a heading. The tool will find the instances and for each instance traverse through the tree, starting from the instance, to see if any of the parents is a heading. In this example, the first instance of *book* has a parent that is a heading, while the second instance does not. As such, only the first instance is valid for this query.

As another example, assume a user searches for the word *fire*, that is used in a quote and where the author is Shakespeare. The tool will find the tokens (*fire*), and for each instance it will traverse the hierarchical structure to find the properties of that instance.

For the first instance, one of the parents of this token is indeed a quote, but the additional bibliographic data show that the author is not Shakespeare. When looking at the next instance, one of the parents of this token is also a quote and the additional bibliographic data indicate that the author is Shakespeare. In the last instance, none of the parents of this token is a quote. The only token that will be retrieved in this example is the second instance.

The core classes in the tool are shown in a UML (Unified Modelling Language) diagram in Figure 110. There are many other minor classes in the tool. Only a selection of the most important classes is shown here to clarify the fundamentals of this tool.

This diagram shows classes and relationships between classes in the tool. An arrow indicates relationship and the direction of the relationship. A 1 and * on an arrow indicates a one to many relationship and a 0..1 and * indicates a zero or one to many relationship. Variables are included in the classes, and variable types are indicated. The maximum length for strings is indicated in brackets. Examples of data are given in white blocks.

This diagram shows how the token is linked to a sentence and to a document and that a document has various bibliographic properties. It shows how the token is linked to a word, a capitalised version, and other properties.

It is also evident that duplication is avoided, by creating tables for different items, such as author, publisher, word and sense. As was explained earlier, through the example of how words are stored and tokens are linked to words, an item is saved only once, and another instance of the same item is linked to the saved item. Examples in this category are language and genre.

Most notably absent from this diagram are the functional properties that were discussed (e.g. headings, paragraphs). The reason for excluding them is that in the implementation they are not real classes in the databases. These concepts are created dynamically, inherent properties from a Part class and are connected by generated links that are of type PartLink. The technical detail of this linking and its representation in the database is beyond the scope of this discussion.

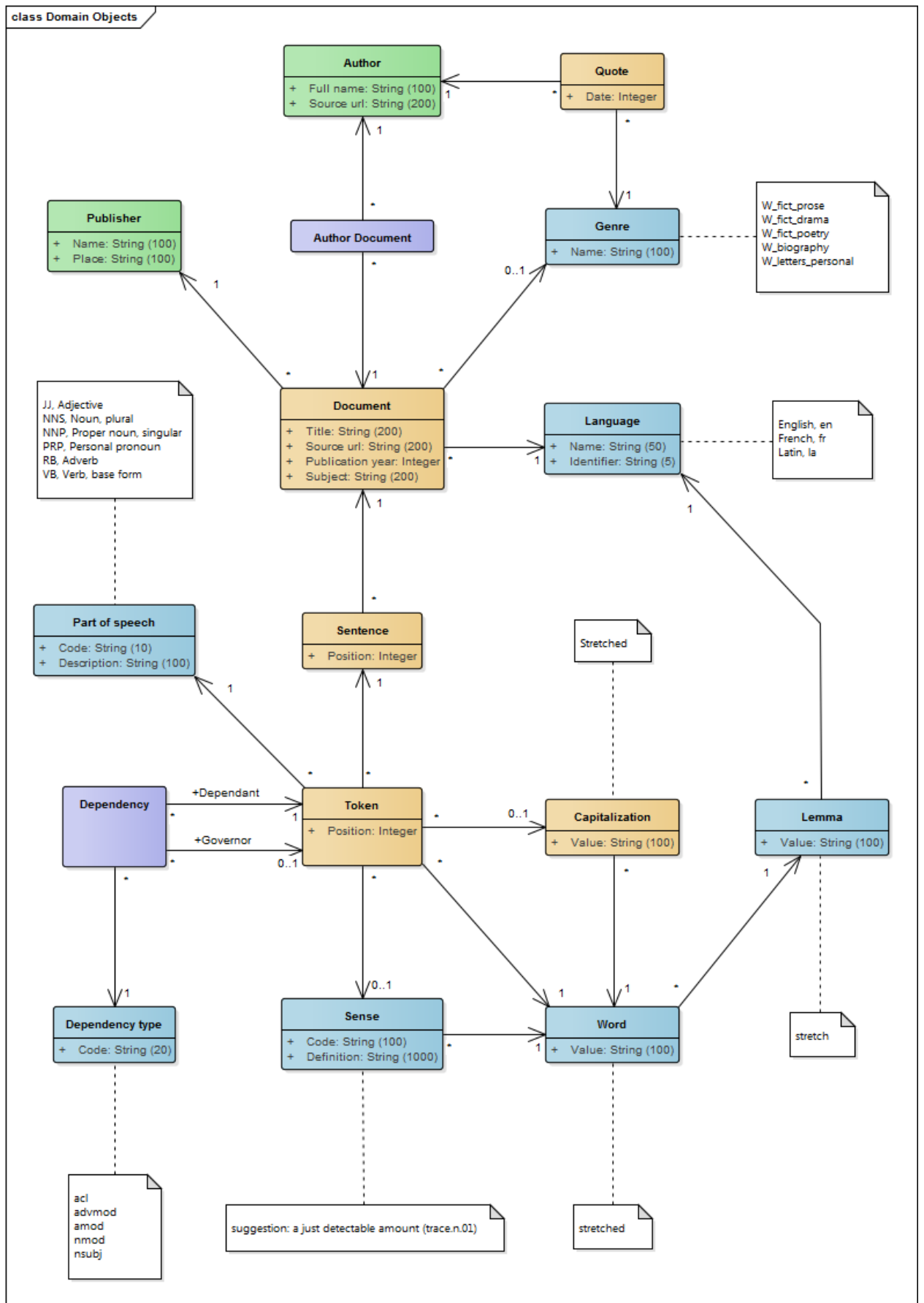


Figure 110 Core classes

6.3. Searching in the tool

6.3.1. Overview of the tool

The search tool that was developed for this study allows a user to search and filter according to different properties of a word. The tool is called *inkling*.

The tool is a prototype, developed to prove a concept and not to be a commercial system. As such, it has a simple interface, sufficient to test functionality. There are no superfluous graphics or other design elements. The tool contains all elements necessary to test the premises of this study.

To demonstrate the functionality of the tool various examples will be used. Some examples can be considered sensible in that they could be examples of information that someone could realistically search for. Other examples could be considered nonsensical, as someone will not necessarily search for such information, but these examples are necessary to demonstrate the functionality of the tool.

6.3.2. Home page

The home page for *inkling* is shown in Figure 111.

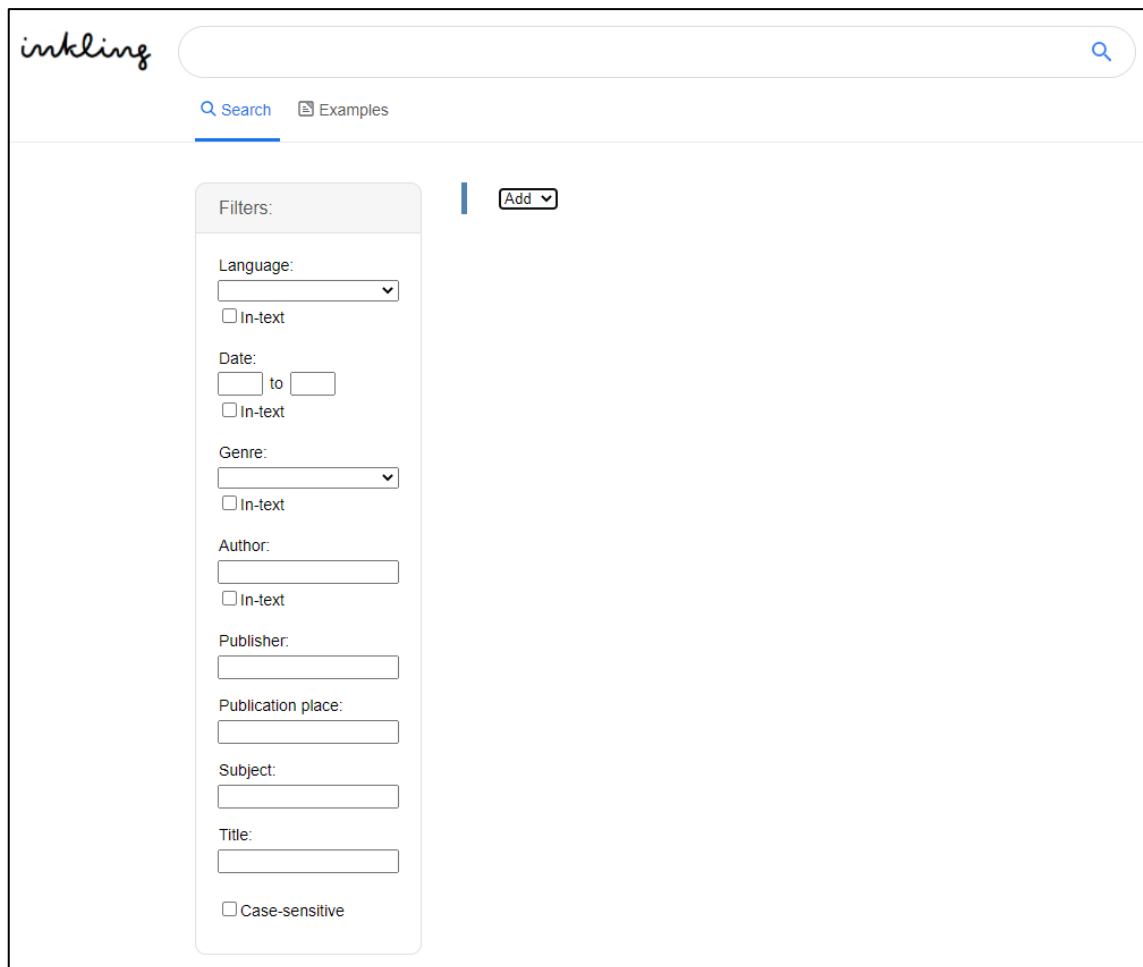


Figure 111 Home page

There is a search field at the top of the page. A word or search query could be entered in this field. Below the search field is the main page that includes the filters, the graphical user interface for searching and the area where the results are displayed.

The search field can offer the user the option to enter a simple word to search for, or to enter a more advanced search using a query language. This field has not been enabled in this version. Currently, it displays the query string that can be used for the search the user is building with the graphical user interface. The purpose is to demonstrate that searching by using a query language could be implemented in this tool. This will be discussed in more detail when discussing searching using a query language.

6.3.3. The graphical user interface

The graphical user interface allows the user to build queries using dropdown menus, checkboxes and input fields. To start a search, the user has to open the dropdown menu and select an item with which to start the search (Figure 112).

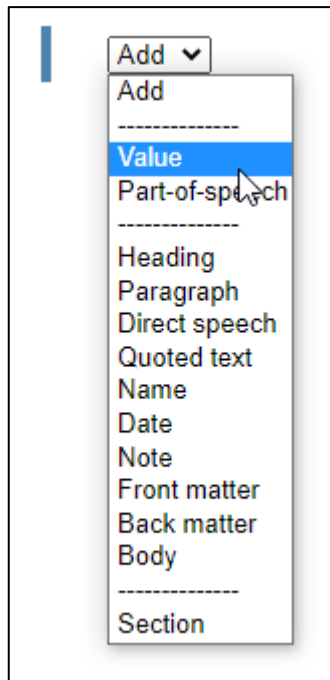


Figure 112 Menu to add items to search

The dropdown menu is separated into three segments, divided by a dashed line. The menu is dynamic and menu items can be added or removed depending on the context. (This will be explained subsequently with examples.) The first segment in the menu allows the user to add a value or a part-of-speech category to search for.

6.3.4. Simple search

The most basic option is for a user to search for a single, complete word. In order to do this, the user selects the *Value* option and enters a value in the field. Figure 113 shows that the user selected to add a value. An input field appears in which the user enters the value to search for, in this example *out*. The results are listed in the results pane.

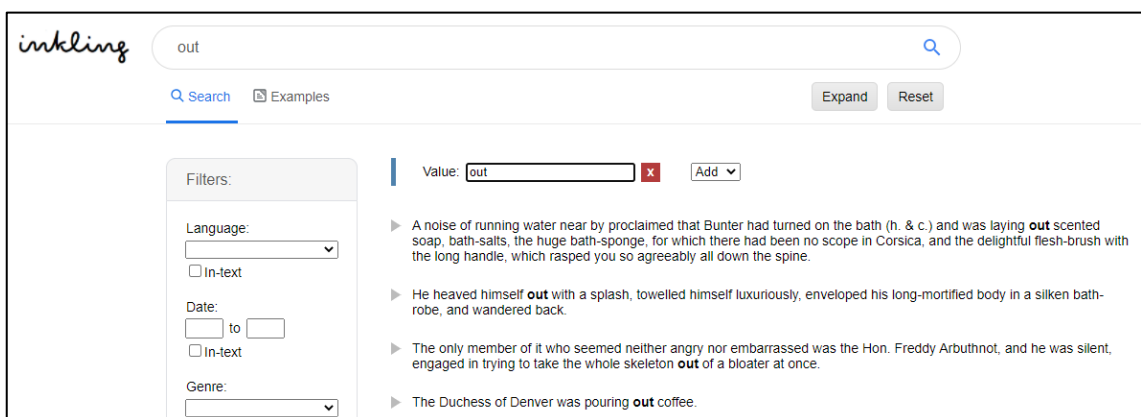


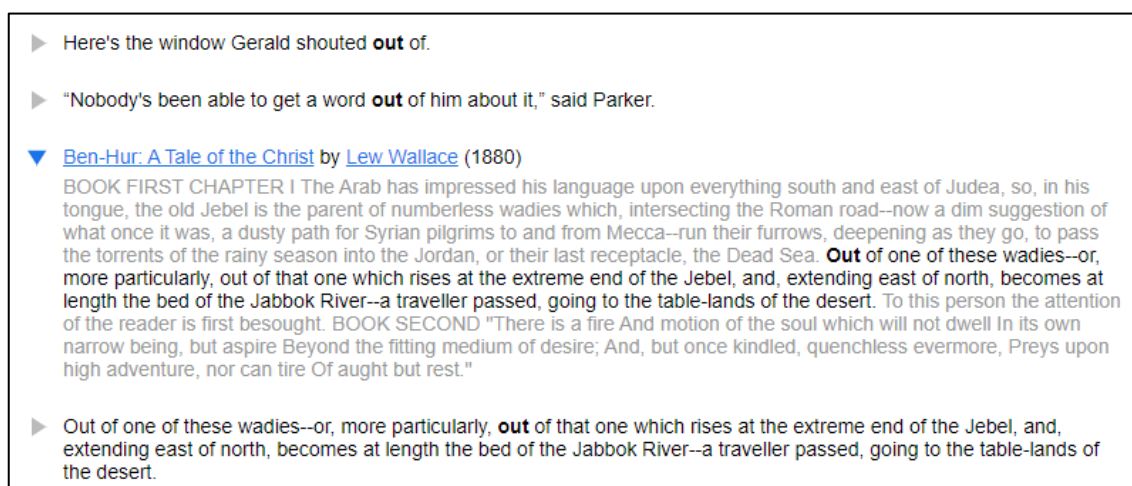
Figure 113 Simple search in inkling

It is also apparent that the search field at the top of the page is being populated. This is a simple example, where a single word is being searched for. As the search gets more complex, more commands are added to the search field.

The screenshots from here onwards will not show the logo or the search field at the top. Sections of the screen will be cropped out to ensure the greatest clarity.

6.3.5. Results

The results show the instances that match the search query. Some context is given for each instance. The value that is being searched for is highlighted in bold in each result. Minimal context is shown, but a user can expand an item in the results list to see more context. Each result is preceded by a triangle. The triangle can be used to expand or collapse a result. In Figure 114, the third result has been expanded to show more context. The value that is being searched for is in bold, the minimal context is in black and the broader context is in grey. Some bibliographic data about the text (document) that the item is from are also given, namely, the title, author, and publication date.



The screenshot shows a search results interface. It contains three items, each preceded by a triangle icon. The first two items are collapsed, indicated by right-pointing triangles. The third item is expanded, indicated by a downward-pointing triangle. The expanded item shows a title and author in blue, followed by a block of text in grey. Within this text, the word 'out' is bolded. Below the grey text, there is another block of text in black, also containing the word 'out' in bold. The entire screenshot is enclosed in a black rectangular border.

- ▶ Here's the window Gerald shouted **out** of.
- ▶ "Nobody's been able to get a word **out** of him about it," said Parker.
- ▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
BOOK FIRST CHAPTER I The Arab has impressed his language upon everything south and east of Judea, so, in his tongue, the old Jebel is the parent of numberless wadies which, intersecting the Roman road--now a dim suggestion of what once it was, a dusty path for Syrian pilgrims to and from Mecca--run their furrows, deepening as they go, to pass the torrents of the rainy season into the Jordan, or their last receptacle, the Dead Sea. **Out of one of these wadies--or, more particularly, out of that one which rises at the extreme end of the Jebel, and, extending east of north, becomes at length the bed of the Jabbok River--a traveller passed, going to the table-lands of the desert.** To this person the attention of the reader is first besought. BOOK SECOND "There is a fire And motion of the soul which will not dwell In its own narrow being, but aspire Beyond the fitting medium of desire; And, but once kindled, quenchless evermore, Preys upon high adventure, nor can tire Of aught but rest."
- ▶ Out of one of these wadies--or, more particularly, **out** of that one which rises at the extreme end of the Jebel, and, extending east of north, becomes at length the bed of the Jabbok River--a traveller passed, going to the table-lands of the desert.

Figure 114 Context for an item

A user can follow the link in the title to open the complete text from where this item came (Figure 115), or click on the link in the author's name to open the biography of the author in an external source (Figure 116).

The Project Gutenberg eBook of Ben-Hur: A Tale of the Christ, by Lew Wallace

This eBook is for the use of anyone anywhere at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this eBook or online at www.gutenberg.net

Title: Ben-Hur: A Tale of the Christ

Author: Lew Wallace

Posting Date: January 19, 2010 [EBook #2145]
 Release Date: April, 2000
 [Last updated: May 18, 2014]

Language: English

Character set encoding: ISO-8859-1

*** START OF THIS PROJECT GUTENBERG EBOOK BEN-HUR: A TALE OF THE CHRIST ***

Produced by an anonymous Project Gutenberg volunteer. HTML version by Al Haines.

Ben-Hur: A Tale of the Christ
 by Lew Wallace

Figure 115 Source of text

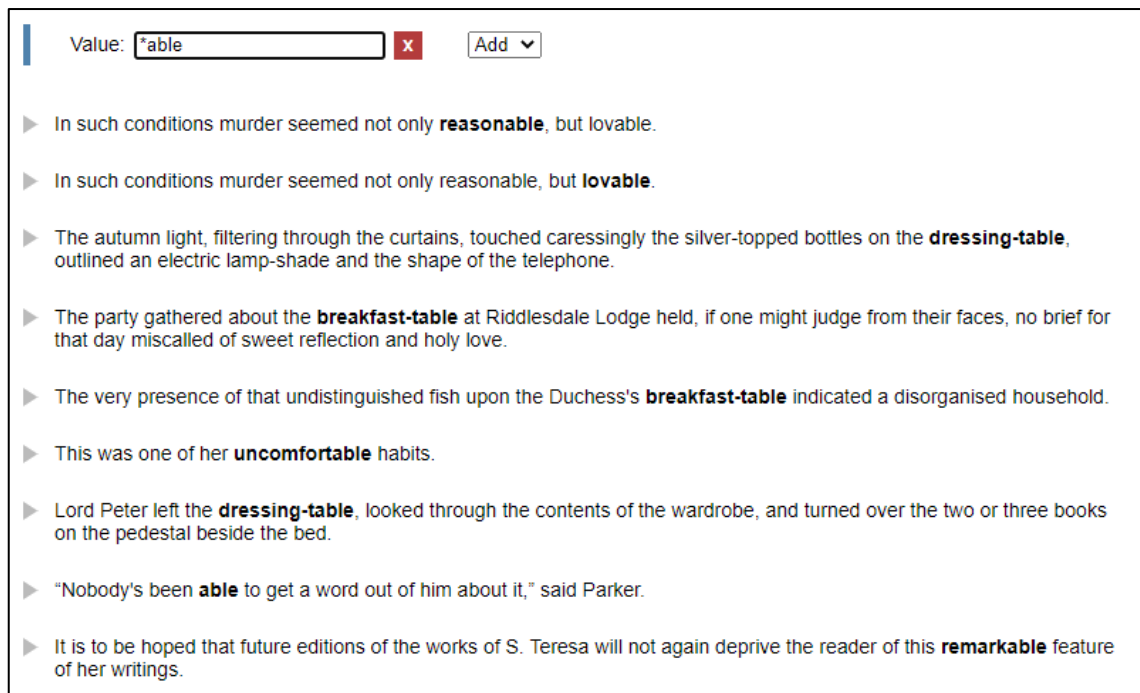
The screenshot shows the Wikipedia article for Lew Wallace. On the left is the Wikipedia logo and a sidebar with navigation links: Main page, Contents, Featured content, Current events, Random article, Donate to Wikipedia, Wikipedia store, Interaction, Help, About Wikipedia, Community portal, Recent changes, and Contact page. The main content area has tabs for 'Article' and 'Talk', and buttons for 'Read' and 'Edit'. The title is 'Lew Wallace' with a subtitle 'From Wikipedia, the free encyclopedia'. The text begins with a note: 'For the Oregon state senator, see Lew Wallace (politician)'. The main text starts with 'Lewis Wallace (April 10, 1827 – February 15, 1905) was an American lawyer, Union general in the American Civil War, governor of the New Mexico Territory, politician, diplomat, and author from Indiana. Among his novels and biographies, Wallace is best known for his historical adventure story, *Ben-Hur: A Tale of the Christ* (1880), a bestselling novel that has been called "the most influential Christian book of the nineteenth century."^[1] Wallace's military career included service in the Mexican–American War and the American Civil War.

Figure 116 Biography of author

6.3.6. Truncation

It is possible to search for complete words or use truncation to search for word stems. The truncation symbol used in this system is the asterisk (*). Left and right truncation is allowed. For example, a user can search for *bath**, and retrieve items such as *bath*,

bath-robe, *bath-salts*, *bath-sponge*; or a user can search for **able*, and return items such as *reasonable*, *lovable*, *dressing-table* (see Figure 117).



The screenshot shows a search interface. At the top, there is a search box containing the text '*able'. To the right of the search box is a red 'x' icon and an 'Add' button with a dropdown arrow. Below the search box is a list of search results, each preceded by a right-pointing triangle icon. The results are as follows:

- ▶ In such conditions murder seemed not only **reasonable**, but lovable.
- ▶ In such conditions murder seemed not only reasonable, but **lovable**.
- ▶ The autumn light, filtering through the curtains, touched caressingly the silver-topped bottles on the **dressing-table**, outlined an electric lamp-shade and the shape of the telephone.
- ▶ The party gathered about the **breakfast-table** at Riddlesdale Lodge held, if one might judge from their faces, no brief for that day miscalled of sweet reflection and holy love.
- ▶ The very presence of that undistinguished fish upon the Duchess's **breakfast-table** indicated a disorganised household.
- ▶ This was one of her **uncomfortable** habits.
- ▶ Lord Peter left the **dressing-table**, looked through the contents of the wardrobe, and turned over the two or three books on the pedestal beside the bed.
- ▶ "Nobody's been **able** to get a word out of him about it," said Parker.
- ▶ It is to be hoped that future editions of the works of S. Teresa will not again deprive the reader of this **remarkable** feature of her writings.

Figure 117 Truncation

The prototype does not allow wildcard characters within a word.

6.3.7. Searching for inflected forms

A user can also specify that all inflected forms of a term should be retrieved. This is effectively searching in the lemma field. In other words, a user can search for a term, such as *run*, and specify that all inflected forms must be retrieved, and the results will include the inflections for *run* (*running*, *ran*, *runs*). The example in Figure 118 shows a simple search for the word *have* where the inflected form is not considered. The menu is dynamically changed to include an option for inflections. The menu is adapted dynamically according to context, to avoid unnecessary errors. If a value is not specified, it does not make sense to search for inflections.

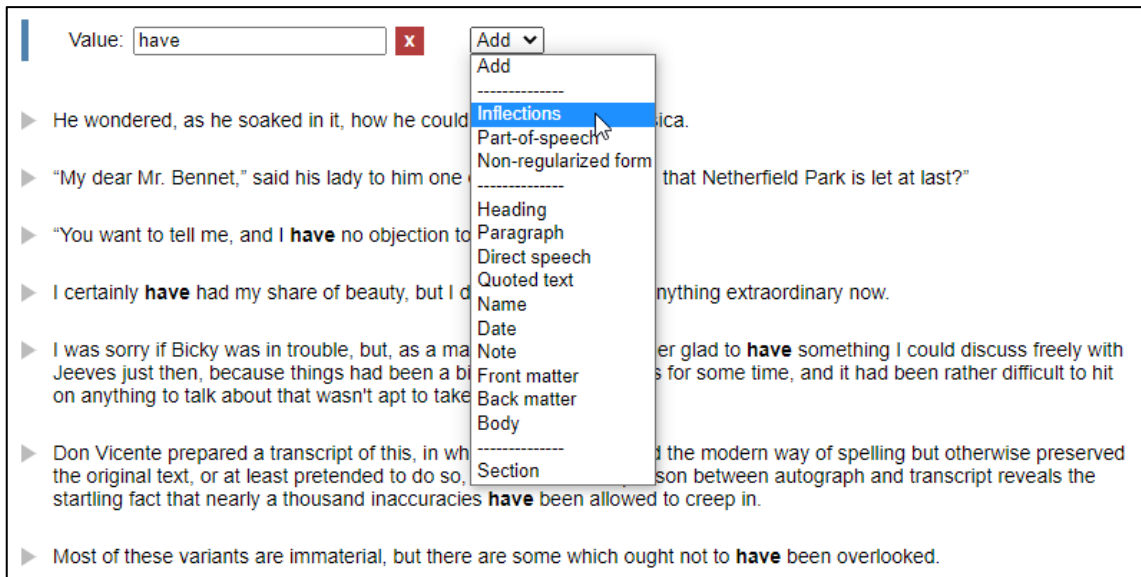


Figure 118 Option to search for inflected forms appears in the menu

As the user selects *Inflections* from the menu, all inflected forms of the word are retrieved. The results shown in the screenshot in Figure 119 now include inflections of the term *have*, namely *hath* and *had*. The following are also retrieved but could not be shown on one screenshot: *has*, *-s*, *-'ve*, *having*.

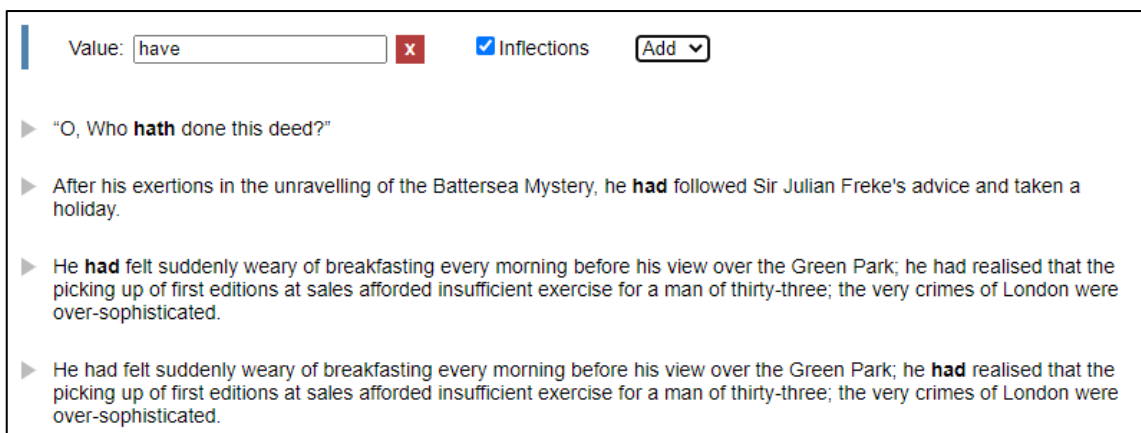


Figure 119 Inflections retrieved

The current implementation does not allow a user to search for a word in its inflected form (e.g. *ran*) and then retrieve all other inflections. This is, however, technically possible to implement. This would mean that the search term first has to be passed to an NLP tool that will determine the lemma and then returns this information to the search tool. This could be considered in a future implementation of the tool.

6.3.8. Part-of-speech category

A user can search for words of a specific part-of-speech category. Consider the search for all instances that end with *-able* (shown in Figure 120). A user could specify that only singular nouns should be retrieved.

The user first selects to add the *Part-of-speech* option (Figure 120), then selects the part-of-speech to use (Figure 121), and the results are returned (Figure 122).

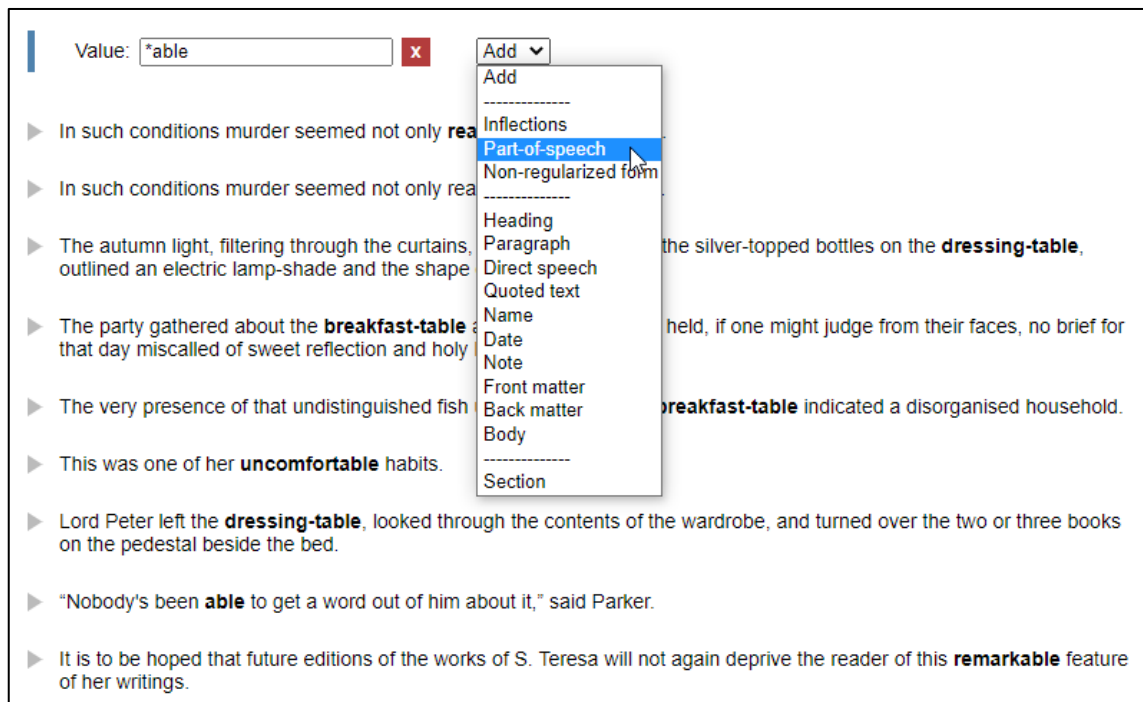


Figure 120 Select part-of-speech from menu

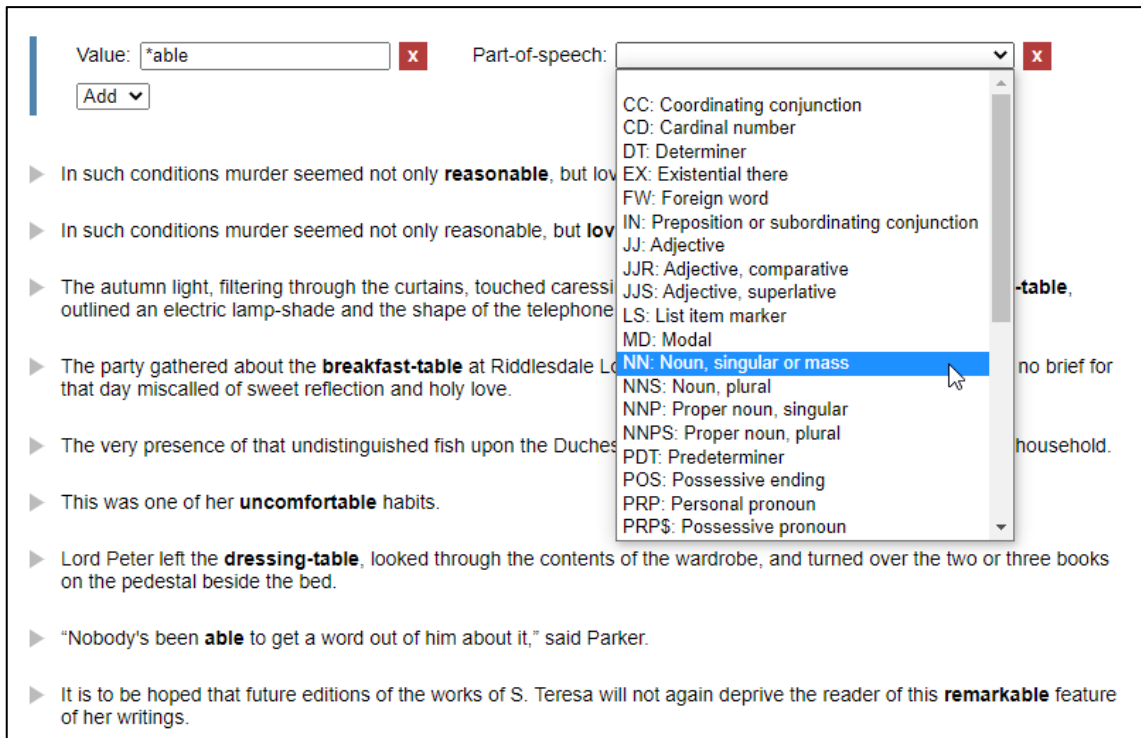


Figure 121 Select noun (singular or mass) from list

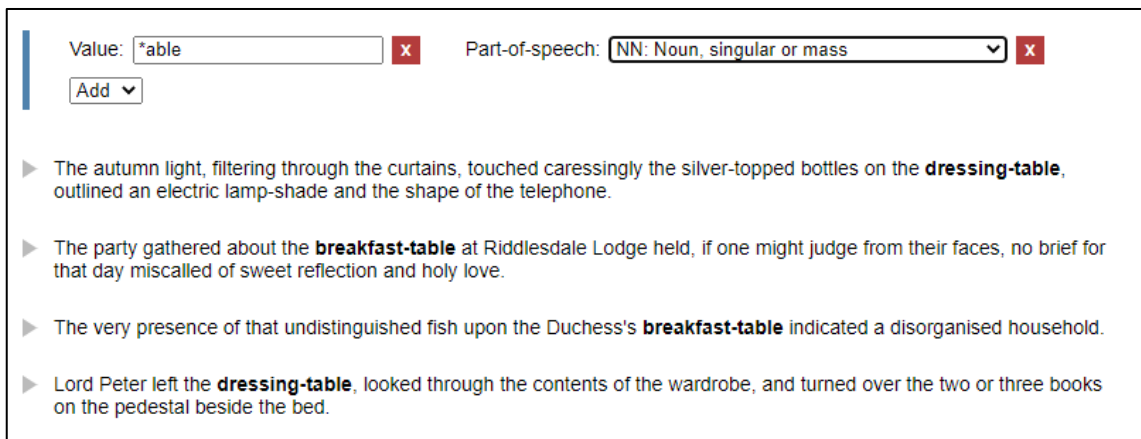


Figure 122 Only nouns are returned

Though part-of-speech tags can be selected from the dropdown list, a good working knowledge of part-of-speech analysis and the tags that are used will be beneficial when searching using the metadata of the morphological level.

6.3.9. Multiple words

It is possible to search for multiple words or phrases. A user will specify a value to search for, then select from the menu to add a new section to add the next word. In Figure 123, a user searches for the value *good* and then adds a new section. The user will then specify the next value to search for. In Figure 124, the user specifies that the next value to search for is *morning*. This will return results for the phrase *good morning*.

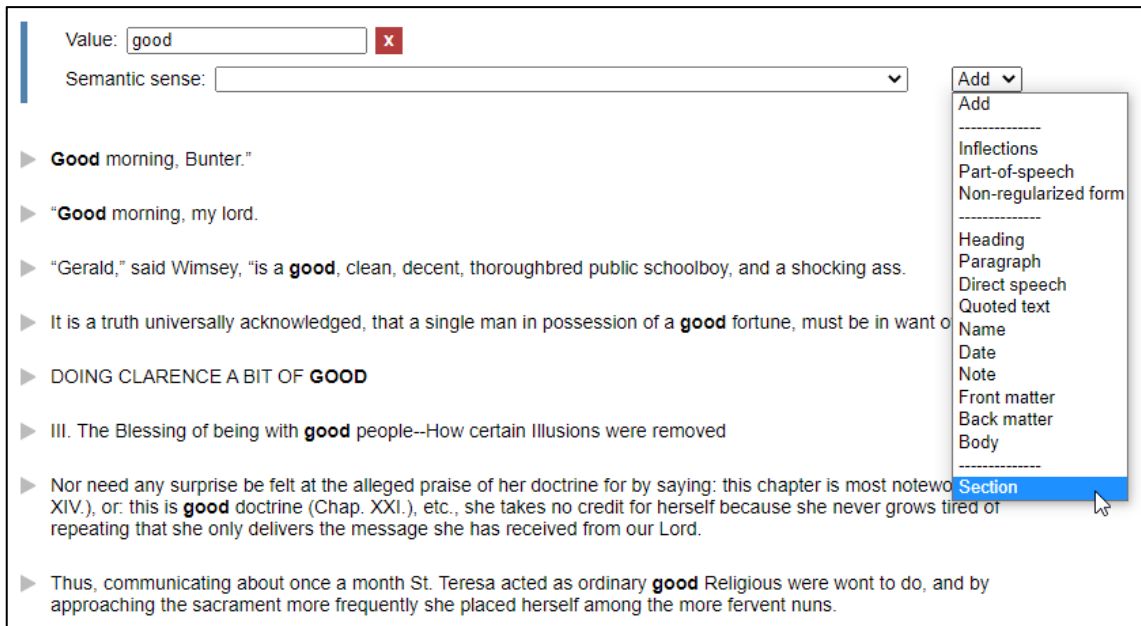


Figure 123 Adding a section

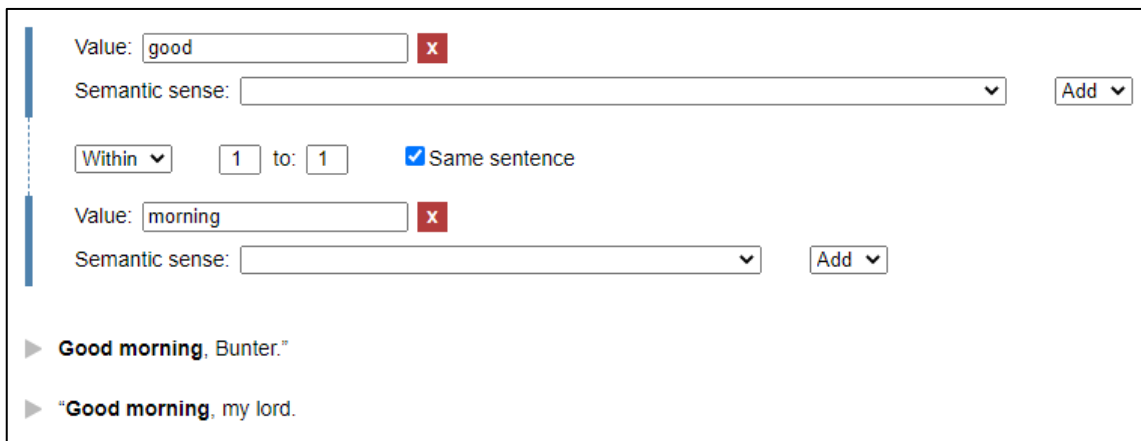


Figure 124 Adding a word to search for

Apart from searching for phrases, a user can search for words near other words and specify if these words should be in the same sentence or not. A user can specify how many words are allowed between words that are being searched for by changing the numbers next to the *Within* dropdown. In Figure 125, the search query specifies that the first value is *single* and must be within 8 words from the value *fortune*, and it must be within the same sentence.

Value:

to: Same sentence

Value:

▶ It is a truth universally acknowledged, that a **single** man in possession of a good **fortune**, must be in want of a wife.

▶ A **single** man of large **fortune**; four or five thousand a year.

Figure 125 Searching for words near other words

It is also possible to search for words that are near each other that are not in the same sentence (Figure 126). The default is to search in the forward direction (words right of the word being searched for), but by adding a minus to the first number, the user can change the direction to search left of the first value (Figure 127).

Value:

to: Same sentence

Value:

▶ It was a glorious **bath**. He wondered, as he **soaked** in it, how he could have existed in Corsica.

Figure 126 Searching across sentences

Value:

Semantic sense:

to: Same sentence

Value: Semantic sense:

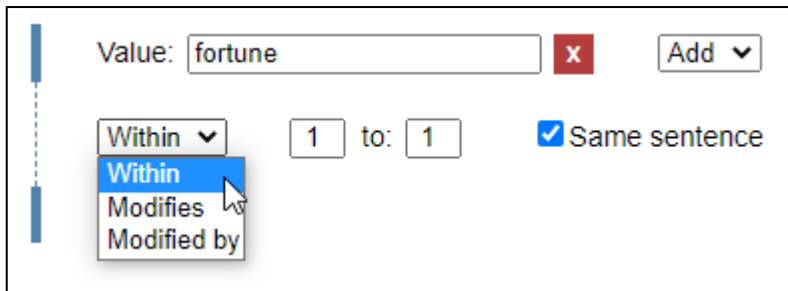
▶ You and the girls may go, or you may send them by themselves, which perhaps will be still better, for as you are as handsome as any of them, Mr. Bingley may **like** you the **best** of the party."

Figure 127 Searching in the opposite direction

6.3.10. Dependencies

A user can search for relationship between words, as defined by dependency grammar. This means a user can search for words that are modified by other words or modify other words. The type of dependency relationship can also be specified.

In Figure 128, the value that is being searched for is *fortune*. A new section is added. The *Within* dropdown can be changed to either *Modifies* or *Modified by*.



The screenshot shows a search interface. At the top, there is a text input field containing the word "fortune", followed by a red "x" icon and an "Add" button with a dropdown arrow. Below this, there is a "Within" dropdown menu that is open, showing three options: "Within" (highlighted in blue), "Modifies", and "Modified by". To the right of the dropdown, there are two input fields, both containing the number "1", with the text "to:" between them. Further right, there is a checked checkbox labeled "Same sentence".

Figure 128 Selecting to search for dependencies

The menu items for the syntactic dependencies are included in the *Within* dropdown, because the options in this dropdown specify how one value (that is being search for) should relate to another value (that is being searched for). The simplest option is specifying the proximity of values (by using the *Within* option). The other options are to search for words that have a relation with another word (by using *Modifies* and *Modified*).

In Figure 129 the search is for all instances where *fortune* is modified by *large*, and where the dependency type is adjectival modifier.



The screenshot shows a search interface with two search criteria. The first criterion has a "Value:" field with "fortune", a red "x" icon, and an "Add" button. The second criterion has a "Value:" field with "large", a red "x" icon, and an "Add" button. Between the two criteria, there is a "Modified by" dropdown menu and a "Dependency type:" dropdown menu set to "amod: Adjectival modifier". Below the search criteria, there are two search results, each preceded by a right-pointing triangle icon. The first result is a quote: "Why, my dear, you must know, Mrs. Long says that Netherfield is taken by a young man of **large fortune** from the north of England; that he came down on Monday in a chaise and four to see the place, and was so much delighted with it, that he agreed with Mr. Morris immediately; that he is to take possession before Michaelmas, and some of his servants are to be in the house by the end of next week." The second result is: "A single man of **large fortune**; four or five thousand a year."

Figure 129 Searching for dependencies

Some more examples are given to illustrate this type of searching. In Figure 130, instances where *very* modifies an adjective are retrieved and in Figure 131 instances where *his* modifies a noun are retrieved.

Value:

Semantic sense:

Dependency type:

Part-of-speech:

▶ It had been **very refreshing**.

▶ But it is **very likely** that he may fall in love with one of them, and therefore you must visit him as soon as he comes."

Figure 130 Example where very modifies an adjective

Value:

Dependency type:

Part-of-speech:

▶ "OF **HIS MALICE** AFORETHOUGHT"

▶ He had felt suddenly weary of breakfasting every morning before **his view** over the Green Park; he had realised that the picking up of first editions at sales afforded insufficient exercise for a man of thirty-three; the very crimes of London were over-sophisticated.

▶ He had abandoned **his flat** and his friends and fled to the wilds of Corsica.

▶ Bunter, **his confidential man** and assistant sleuth, had nobly sacrificed his civilised habits, had let his master go dirty and even unshaven, and had turned his faithful camera from the recording of finger-prints to that of craggy scenery.

▶ Bunter, his confidential man and assistant sleuth, had nobly sacrificed his civilised habits, had let **his master** go dirty and even unshaven, and had turned his faithful camera from the recording of finger-prints to that of craggy scenery.

▶ Bunter, his confidential man and assistant sleuth, had nobly sacrificed his civilised habits, had let his master go dirty and even unshaven, and had turned **his faithful camera** from the recording of finger-prints to that of craggy scenery.

▶ He heaved himself out with a splash, towelled himself luxuriously, enveloped **his long-mortified body** in a silken bath-robe, and wandered back.

▶ To **his immense surprise** he perceived Mr. Bunter calmly replacing all the fittings in his dressing-case.

Figure 131 Example where his modifies a noun

Another example of using dependency types to retrieve specific instances is given. Consider the example where a user would like to retrieve all instances where a specific word is a direct object in a sentence and can modify any verb. For example, if the direct object is *book*, the user would want to retrieve examples such as *reads/throws/studies/sees the book*. The example in Figure 132 illustrates how this type of searching can be performed using the dependency types in this prototype. The direct object the researcher would like to retrieve is *reader*, in combination with any verb. Therefore, the value is specified as *reader*. To indicate that it should be the direct object, the dependency type is set to direct object. To indicate that any verb can be

modified by the word, the part-of-speech category is set to verb. The examples *carry the reader* and *deprive the reader* are returned.

The screenshot shows a search interface with the following elements:

- Value:** A text input field containing "reader" with a red "x" icon to its right.
- Semantic sense:** A dropdown menu.
- Add:** A button with a downward arrow.
- Modifies:** A dropdown menu.
- Dependency type:** A dropdown menu containing "dobj: Direct object".
- Part-of-speech:** A dropdown menu containing "VB: Verb, base form" with a red "x" icon to its right.
- Add:** A button with a downward arrow.

Below the filters, there are two example sentences:

- ▶ It is necessary now to **carry** the **reader** forward twenty-one years, to the beginning of the administration of Valerius Gratus, the fourth imperial governor of Judea --a period which will be remembered as rent by political agitations in Jerusalem, if, indeed, it be not the precise time of the opening of the final quarrel between the Jew and the Roman.
- ▶ It is to be hoped that future editions of the works of S. Teresa will not again **deprive** the **reader** of this remarkable feature of her writings.

Figure 132 Searching for instances where reader is the direct object

When searching according to dependency grammar (words modifying other words with specific relationships) a good working knowledge of syntax is necessary. The user will need to understand the theory of syntactic dependencies as well as the annotations that were used for the dependencies. Even though the labels are written out (e.g. direct object for *dobj*), the user will need to know what the labels mean. This can be illustrated by the tag *case*, which is used in these guidelines (UD) for all case-marking elements including prepositions. Other guidelines may use other labels or terms.

Most of the other types of searches in this prototype will be useful to a broader audience, but particularly on the morphological and syntactic level, linguistic knowledge is necessary to gain the full benefit of searching using the metadata on these levels.

6.3.11. Search according to meaning (semantics)

A word can have multiple meanings. If a user searches for a word, for which there are multiple meanings in the database, a *Semantic sense* dropdown menu appears. The user can now select the meaning that (s)he is interested in. In Figure 133, the value that is being searched for is *man*. There are two different meanings for *man* recorded in the database. The first meaning is *an adult person who is male* and the second meaning is *a manservant who acts as a personal attendant to his employer*. In this example, the second meaning has been selected (Figure 134). Now the tool returns all the results that have the same meaning, but not necessarily the same word. For example, the word *valet* shares the meaning *a manservant who acts as a personal attendant to his employer*. If a user would like to see only the original word with a specific meaning, the *Match value* checkbox must be selected (see Figure 135).

Value: x

Semantic sense: Add ▾

man.n.01: an adult person who is male (as opposed to a woman)

valet.n.01: a manservant who acts as a personal attendant to his employer

- ▶ He had felt suddenly weary, or drowsy, or, morning before his hour, he had realised that the picking up of first editions at sales afforded insufficient exercise for a **man** of thirty-three; the very crimes of London were over-sophisticated.
- ▶ Bunter, his confidential **man** and assistant sleuth, had nobly sacrificed his civilised habits, had let his master go dirty and even unshaven, and had turned his faithful camera from the recording of finger-prints to that of craggy scenery.
- ▶ It is a truth universally acknowledged, that a single **man** in possession of a good fortune, must be in want of a wife.
- ▶ However little known the feelings or views of such a **man** may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters.
- ▶ "Why, my dear, you must know, Mrs. Long says that Netherfield is taken by a young **man** of large fortune from the north of England; that he came down on Monday in a chaise and four to see the place, and was so much delighted with it, that he agreed with Mr. Morris immediately; that he is to take possession before Michaelmas, and some of his servants are to be in the house by the end of next week."
- ▶ A single **man** of large fortune; four or five thousand a year.
- ▶ Sometimes of a morning, as I've sat in bed sucking down the early cup of tea and watched my **man** Jeeves flitting about the room and putting out the raiment for the day, I've wondered what the deuce I should do if the fellow ever took it into his head to leave me.

Figure 133 Searching according to semantic sense

Value: x

Semantic sense: Match value

Add ▾

- ▶ In a soporific interval he heard the **valet** de chambre bringing in coffee and rolls.
- ▶ Sometimes of a morning, as I've sat in bed sucking down the early cup of tea and watched my **man** Jeeves flitting about the room and putting out the raiment for the day, I've wondered what the deuce I should do if the fellow ever took it into his head to leave me.
- ▶ Young Reggie Foljambe to my certain knowledge offered him double what I was giving him, and Alistair Bingham-Reeves, who's got a **valet** who had been known to press his trousers sideways, used to look at him, when he came to see me, with a kind of glittering hungry eye which disturbed me deucedly.

Figure 134 A sense has been selected

Value: x

Semantic sense: Match value

Add ▾

- ▶ Sometimes of a morning, as I've sat in bed sucking down the early cup of tea and watched my **man** Jeeves flitting about the room and putting out the raiment for the day, I've wondered what the deuce I should do if the fellow ever took it into his head to leave me.

Figure 135 Search for a sense and match the value

6.3.12. Search according to functional properties

A user can search according to the functional properties that were encoded, namely, the heading, paragraph, direct speech, quoted text, names, dates, notes, front matter, back matter, body and regularised form. The in-text bibliographic properties are also encoded on this level but searching according to these properties will be discussed with the document-level bibliographic properties.

The functional properties can be selected from the dropdown menu. Most of these properties are grouped together after the second dashed line (see Figure 136). To search for a value in its original form (not regularised) is treated differently and will also be discussed.

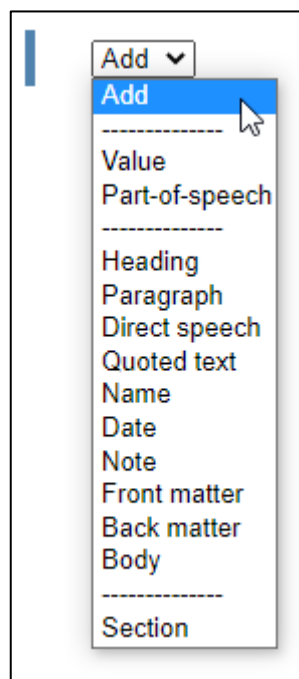


Figure 136 Functional properties in the menu

The use of these properties will be illustrated with an example for each property.

The search term in Figure 137 is **ought*, searching for all instances that end in *-ought*. (This is an example of a search that is used to demonstrate functionality but is not necessarily a realistic example.) The dropdown menu is used to select the *Heading* option, to search for instances that appear in headings (Figure 138). The only result that ends in *-ought* which also is in a heading in this prototype is *Of his malice aforethought*, which is a heading in the book *Clouds of Witness* (Figure 139).

Value:

- ▶ "OF HIS MALICE **AFORETHOUGHT**"
- ▶ Is there anything else in this room I **ought** to look at?"
- ▶ To this person the attention of the reader is first **besought**.
- ▶ When a woman has five grown-up daughters, she **ought** to give over thinking of her own beauty."
- ▶ I got back to the flat latish one night, and when Jeeves **brought** me the final drink he said: "Mr. Bickersteth called to see you this evening, sir, while you were out."
- ▶ Most of these variants are immaterial, but there are some which **ought** not to have been overlooked.
- ▶ What she herself **thought** of her books is best told by Yepes in a letter to Father Luis de Leon, the first editor of her works: "She was pleased when her writings were being praised and her Order and the convents were held in esteem.

Figure 137 Searching for the value *ought on inkling

Value:

- ▶ "OF HIS MALICE **AFORETHOUGHT**"
- ▶ Is there anything else in this room I **ought** to lo
- ▶ To this person the attention of the reader is first
- ▶ When a woman has five grown-up daughters, s
- ▶ I got back to the flat latish one night, and when you this evening, sir, while you were out."
- ▶ Most of these variants are immaterial, but there
- ▶ What she herself **thought** of her books is best told by Yepes in a letter to Father Luis de Leon, the first editor of her works: "She was pleased when her writings were being praised and her Order and the convents were held in esteem.

Dropdown menu options:

- Add
-
- Inflections
- Part-of-speech
- Non-regularized form
-
- Heading**
- Paragraph
- Direct speech
- Quoted text
- Name
- Date
- Note
- Front matter
- Back matter
- Body
-
- Section

Figure 138 Selecting to search only in headings

Value: Heading

▼ [Clouds of Witness](#) by [Dorothy L. Sayers](#) (1958)

CHAPTER "OF HIS MALICE **AFORETHOUGHT**" "O, Who hath done this deed?" Othello Lord Peter Wimsey stretched himself luxuriously between the sheets provided by the Hôtel Meurice.

Figure 139 Filtered according to heading

Figure 140 takes the same search for instances ending in *-ought*, but displays the instances that appear in paragraphs, not headings.

Value: Paragraph

- ▶ Is there anything else in this room I **ought** to look at?"
- ▶ To this person the attention of the reader is first **besought**.
- ▶ When a woman has five grown-up daughters, she **ought** to give over thinking of her own beauty."
- ▶ I got back to the flat latish one night, and when Jeeves **brought** me the final drink he said: "Mr. Bickersteth called to see you this evening, sir, while you were out."
- ▶ Most of these variants are immaterial, but there are some which **ought** not to have been overlooked.
- ▶ What she herself **thought** of her books is best told by Yepes in a letter to Father Luis de Leon, the first editor of her works: "She was pleased when her writings were being praised and her Order and the convents were held in esteem.

Figure 140 Filtering according to paragraph

In Figure 141 the search term is *her*. The *Direct speech* option is selected from the dropdown menu and instances occurring in direct speech are retrieved (Figure 142).

Value:

- ▶ This was one of **her** uncomfortable habits.
- ▶ She was a long-necked, long-backed woman, v air and her children.
- ▶ She was a long-necked, long-backed woman, v air and **her** children.
- ▶ She was never embarrassed, and **her** anger, th d to be visible, made itself felt the more.
- ▶ I could hardly keep the Duchess out of **her** bed
- ▶ In the interval Judea had been subjected to cha many ways, but in nothing so much as her political status.
- ▶ In the interval Judea had been subjected to changes affecting her in many ways, but in nothing so much as **her** political status.
- ▶ When a woman has five grown-up daughters, she ought to give over thinking of **her** own beauty."

Dropdown menu options: Add, Inflections, Part-of-speech, Non-regularized form, Heading, Paragraph, **Direct speech**, Quoted text, Name, Date, Note, Front matter, Back matter, Body, Section

Figure 141 Searching for the value her on inkling

Value: Direct

- ▶ I could hardly keep the Duchess out of **her** bedroom."
- ▶ When a woman has five grown-up daughters, she ought to give over thinking of **her** own beauty."

Figure 142 Filtered according to direct speech

In Figure 143 the value being searched for is *not*; in Figure 144 only instances of *not* that appear in a quote are retrieved. In this prototype, there are two quotes from *Ben Hur* that have the value *not* in them, the one is a quote from *Childe Harold* and the other from *Antony and Cleopatra*.

Value: ✕ Add ▾

- ▶ In such conditions murder seemed **not** only reasonable, but lovable.
- ▶ "But if he has the letter, why **not** produce it?"
- ▶ "There is a fire And motion of the soul which will **not** dwell In its own narrow being, but aspire Beyond the fitting medium of desire; And, but once kindled, quenchless evermore, Preys upon high adventure, nor can tire Of aught but rest."
- ▶ It is necessary now to carry the reader forward twenty-one years, to the beginning of the administration of Valerius Gratus, the fourth imperial governor of Judea --a period which will be remembered as rent by political agitations in Jerusalem, if, indeed, it be **not** the precise time of the opening of the final quarrel between the Jew and the Roman.
- ▶ His death's upon him, but **not** dead."
- ▶ Mr. Bennet replied that he had **not**.
- ▶ "Do you **not** want to know who has taken it?" cried his wife impatiently.
- ▶ I certainly have had my share of beauty, but I do **not** pretend to be anything extraordinary now.

Figure 143 Searching for the value *not* on inkling

Value: ✕ Quote Add ▾

- ▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
 Out of one of these wadies--or, more particularly, out of that one which rises at the extreme end of the Jebel, and, extending east of north, becomes at length the bed of the Jabbok River--a traveller passed, going to the table-lands of the desert. To this person the attention of the reader is first besought. BOOK SECOND "There is a fire And motion of the soul which will **not** dwell In its own narrow being, but aspire Beyond the fitting medium of desire; And, but once kindled, quenchless evermore, Preys upon high adventure, nor can tire Of aught but rest." Childe Harold. CHAPTER I It is necessary now to carry the reader forward twenty-one years, to the beginning of the administration of Valerius Gratus, the fourth imperial governor of Judea --a period which will be remembered as rent by political agitations in Jerusalem, if, indeed, it be not the precise time of the opening of the final quarrel between the Jew and the Roman.
- ▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
 How now? is he dead? Diomedes. His death's upon him, but **not** dead." Antony and Cleopatra (act iv., sc. xiii.). BOOK FOURTH "Alva.

Figure 144 Filtered according to quote

In Figure 145 all instances ending in *-er* are being searched for and in Figure 146 only instances that are part of a name are retrieved. For example, Peter and Parker are names that end in *-er* and are retrieved.

Value: ✕ Add ▾

▶ **CHAPTER**

▶ Lord **Peter** Wimsey stretched himself luxuriously between the sheets provided by the Hôtel Meurice.

▶ **After** his exertions in the unravelling of the Battersea Mystery, he had followed Sir Julian Freke's advice and taken a holiday.

▶ He had felt suddenly weary of breakfasting every morning before his view **over** the Green Park; he had realised that the picking up of first editions at sales afforded insufficient exercise for a man of thirty-three; the very crimes of London were over-sophisticated.

▶ In such conditions **murder** seemed not only reasonable, but lovable.

▶ **Bunter**, his confidential man and assistant sleuth, had nobly sacrificed his civilised habits, had let his master go dirty and even unshaven, and had turned his faithful camera from the recording of finger-prints to that of craggy scenery.

▶ Bunter, his confidential man and assistant sleuth, had nobly sacrificed his civilised habits, had let his **master** go dirty and even unshaven, and had turned his faithful camera from the recording of finger-prints to that of craggy scenery.

▶ Now, **however**, the call of the blood was upon Lord Peter.

▶ Now, however, the call of the blood was upon Lord **Peter**.

▶ A noise of running **water** near by proclaimed that Bunter had turned on the bath (h. & c.) and was laying out scented soap, bath-salts, the huge bath-sponge, for which there had been no scope in Corsica, and the delightful flesh-brush with

Figure 145 Searching for instances ending in *-er* on inking

Value: ✕ Name Add ▾

▶ Lord **Peter** Wimsey stretched himself luxuriously between the sheets provided by the Hôtel Meurice.

▶ Now, however, the call of the blood was upon Lord **Peter**.

▶ "Contrast," philosophised Lord **Peter** sleepily, "is life.

▶ Good morning, **Bunter**."

▶ "Thanks," said Lord **Peter**.

▶ To his immense surprise he perceived Mr. **Bunter** calmly replacing all the fittings in his dressing-case.

▶ "I didn't suppose anything," said **Parker** mildly.

▶ Lord **Peter** left the dressing-table, looked through the contents of the wardrobe, and turned over the two or three books on the pedestal beside the bed.

▶ "Nobody's been able to get a word out of him about it," said **Parker**.

Figure 146 Filtered according to name

In Figure 147 the search term that was entered is *18*, searching for all instances that contain 18. In Figure 148 only the instances that contain 18 that are also dates are displayed. This means that the last instance in Figure 147 that refers to a page number is not returned.

Value:

- ▶ When Mr. Lewis undertook the translation of St. Teresa's works, he had before him Don Vicente de la Fuente's edition (Madrid, **1861**-1862), supposed to be a faithful transcript of the original.
- ▶ When Mr. Lewis undertook the translation of St. Teresa's works, he had before him Don Vicente de la Fuente's edition (Madrid, 1861-**1862**), supposed to be a faithful transcript of the original.
- ▶ In **1873** the Sociedad Foto-Tipografica-Catolica of Madrid published a photographic reproduction of the Saint's autograph in 412 pages in folio, which establishes the true text once for all.
- ▶ 3. Fuente, Obras (**1881**), vol. vi. p. 133.
- ▶ Communion, effects of the Saint's, xvi. 3-10, xviii. **10-18**, xxx. 16, xxxviii. 24, Rel. iv. 5, Rel. ix. 13; the Saint's longing for, xxxix. 31; graces of, Rel. ix. 20.

Figure 147 Searching for instances with -18- on inkling

Value: Date

- ▶ When Mr. Lewis undertook the translation of St. Teresa's works, he had before him Don Vicente de la Fuente's edition (Madrid, **1861**-1862), supposed to be a faithful transcript of the original.
- ▶ When Mr. Lewis undertook the translation of St. Teresa's works, he had before him Don Vicente de la Fuente's edition (Madrid, 1861-**1862**), supposed to be a faithful transcript of the original.
- ▶ In **1873** the Sociedad Foto-Tipografica-Catolica of Madrid published a photographic reproduction of the Saint's autograph in 412 pages in folio, which establishes the true text once for all.
- ▶ 3. Fuente, Obras (**1881**), vol. vi. p. 133.

Figure 148 Filtered according to date

To search in notes (e.g. footnotes) the user also selects the corresponding option from the dropdown menu. In Figure 149 all instances of *part* are retrieved and in Figure 150 only instances that appear in footnotes are retrieved. This particular example is from the twelfth footnote in *The Life of St. Teresa of Jesus, of the Order of Our Lady of Carmel*.

Value: x

Semantic sense: Add ▾

▶ There used to be all sorts of attempts on the **part** of low blighters to sneak him away from me.

▶ 12. Constitutions of 1462. **Part** i., cap. x.

Figure 149 Searching for the value part in inking

Value: x

Semantic sense: Note

▾

▼ [The Life of St. Teresa of Jesus, of the Order of Our Lady of Carmel](#) by [Teresa of Avila](#) (1904)

3. Fuente, Obras (1881), vol. vi. p. 133. The Constitutions of the Order specified twelve days on which all those that were not priests should communicate, adding: Verumtamen fratres professi prout Deus eis devotionem contulerit diebus dominicis et festis duplicibus (i.e., on feasts of our Lady, the Apostles, etc.), communicare poterunt si qui velint. Thus, communicating about once a month St. Teresa acted as ordinary good Religious were wont to do, and by approaching the sacrament more frequently she placed herself among the more fervent nuns. 12. Constitutions of 1462. **Part** i., cap. x. Index. Cheerfulness, importance of, xii. 1. Cherubim, xxix. 16.

Figure 150 Filtered according to notes

The value *saint* will be used to demonstrate searching in the front and back matter. Front matter includes sections such as a table of contents or title page and back matter includes sections such as an index or glossary. All the instances of *saint* are shown in Figure 151. In Figure 152 only instances in front matter are retrieved and in Figure 153 only instances in back matter are retrieved. In Figure 154 only instances from the body are retrieved.

Value:

- ▶ Annals of the **Saint's** Life
- ▶ I. Childhood and early Impressions--The Blessing of pious Parents--Desire of Martyrdom--Death of the **Saint's** Mother
- ▶ II. Early Impressions--Dangerous Books and Companions--The **Saint** is placed in a Monastery
- ▶ In 1873 the Sociedad Foto-Tipografica-Catolica of Madrid published a photographic reproduction of the **Saint's** autograph in 412 pages in folio, which establishes the true text once for all.
- ▶ The Book of Foundations and the Way of Perfection contain similar arguments in the **Saint's** handwriting.
- ▶ Church, the, ceremonies of, xxxi. 4; the **Saint's** reverence for, xxxiii. 6.
- ▶ Clare, St., encourages the **Saint**, xxxiii. 15.
- ▶ Comforts, worldly, the **Saint's** fear of, xxxiv. 4.
- ▶ Communion, effects of the **Saint's**, xvi. 3-10, xviii. 10-18, xxx. 16, xxxviii. 24, Rel. iv. 5, Rel. ix. 13; the **Saint's** longing for, xxxix. 31; graces of, Rel. ix. 20.

Figure 151 Searching for the value saint on inking

Value: Front matter

- ▶ Annals of the **Saint's** Life
- ▶ I. Childhood and early Impressions--The Blessing of pious Parents--Desire of Martyrdom--Death of the **Saint's** Mother
- ▶ II. Early Impressions--Dangerous Books and Companions--The **Saint** is placed in a Monastery

Figure 152 Filtered according to front matter

Value: Back matter

- ▶ Church, the, ceremonies of, xxxi. 4; the **Saint's** reverence for, xxxiii. 6.
- ▶ Clare, St., encourages the **Saint**, xxxiii. 15.
- ▶ Comforts, worldly, the **Saint's** fear of, xxxiv. 4.
- ▶ Communion, effects of the **Saint's**, xvi. 3-10, xviii. 10-18, xxx. 16, xxxviii. 24, Rel. iv. 5, Rel. ix. 13; the **Saint's** longing for, xxxix. 31; graces of, Rel. ix. 20.
- ▶ Communion, effects of the **Saint's**, xvi. 3-10, xviii. 10-18, xxx. 16, xxxviii. 24, Rel. iv. 5, Rel. ix. 13; the **Saint's** longing for, xxxix. 31; graces of, Rel. ix. 20.
- ▶ Complaint, loving, of the **Saint**, xxxvii. 13.
- ▶ Confession, frequent, of the **Saint**, v. 17; matter of, Rel. v. 11.

Figure 153 Filtered according to back matter

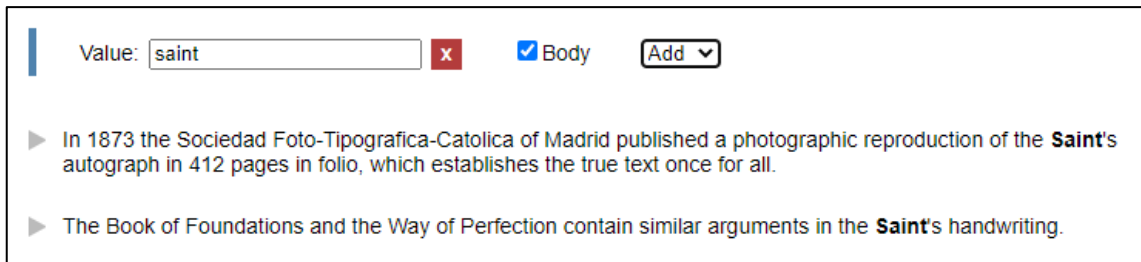


Figure 154 Filtered according to body

The option to search for a value in its original form (non-regularised) is added to the menu as soon as a value is entered. It is added in the first group in the menu, as it conceptually fits with searching for variants of words. The other functional properties denote searching in a section, for example, searching in a paragraph.

The default on inkling is to search for the regularised form of a word. For example, the value 'em found in the sample from *Clouds of Witness*, means *them*. A user has to specify that they want to search for the non-regularised form to search for the value in its original form (Figure 155). The results display the regularised form of an item (Figure 156).

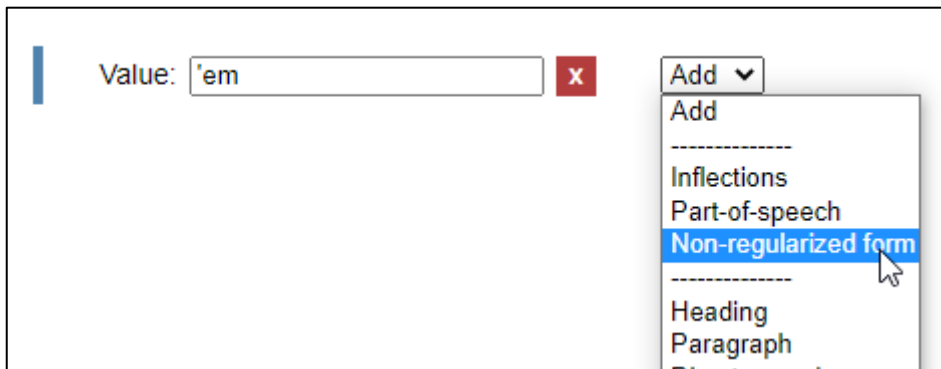


Figure 155 Searching for the value 'em on inkling

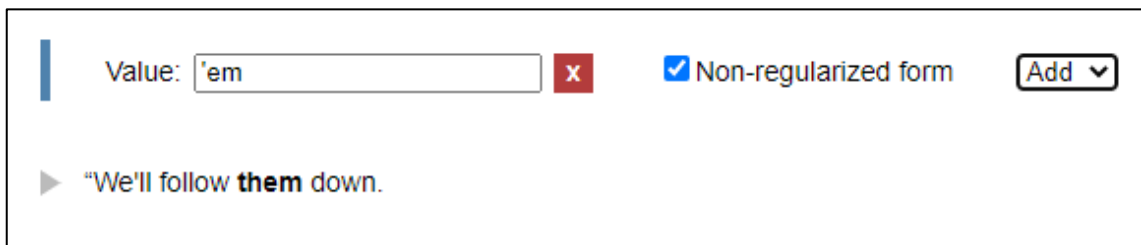


Figure 156 Searching values as non-regularised

It is possible to combine functional properties and to select more than one functional property to search in. In Figure 157 the tool searches for all instances that end in -er, that is a name, that is in a paragraph and that is direct speech.

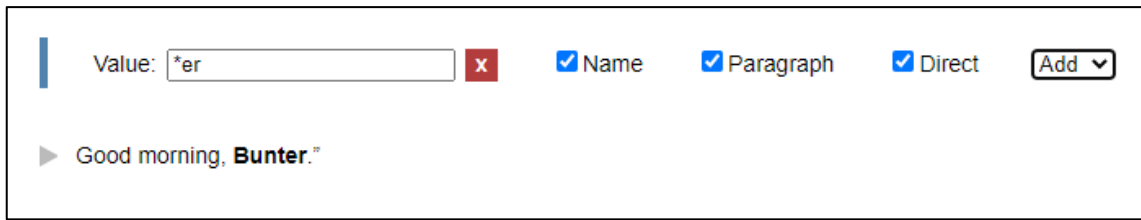


Figure 157 Combining functional properties on inking

6.3.13. Searching according to bibliographic properties

Options to filter according to bibliographic properties are available in a pane on the left of the screen. The following properties are available: language, date, genre, author, publisher, publication place, subject, title. The last option on this pane, is the checkbox to specify whether the search should be case sensitive or not.

Searching for the value **men* will be used to demonstrate filtering by language. Figure 158 shows all the instances that end with *-men* that have been retrieved. The last instance shown on the screenshot is *Verumtamen*, which is Latin.

A user can select to filter the search to a specific language, for example English (Figure 159). This filtering works on the document-level. This means that if the document is regarded as English, then all the words in the document are regarded as English. However, a user can specify that this filtering should be applied to the text-level (Figure 160). This means that the bibliographic properties on a detailed level will be taken into consideration. In this example *Verumtamen* is marked as Latin and excluded from the results. If the filter is changed to retrieve Latin instances, only *Verumtamen* is retrieved (Figure 161).

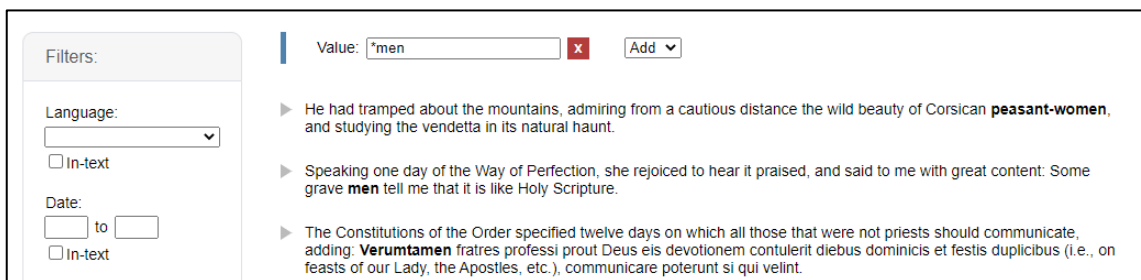


Figure 158 Searching for all instances that end with **men*

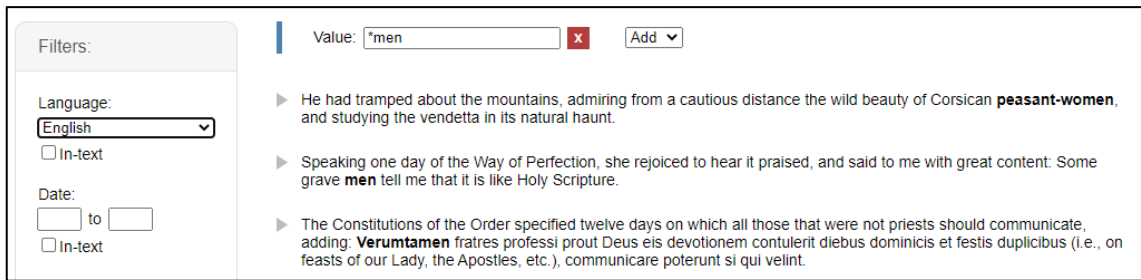


Figure 159 Filtering according to English as language on a document level

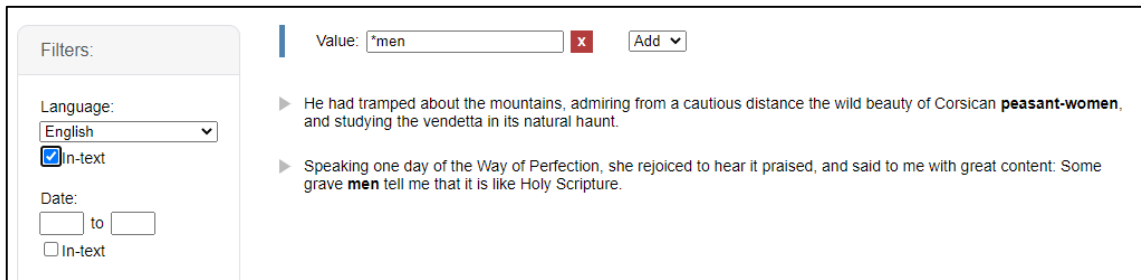


Figure 160 Filtering according to English on a text-level

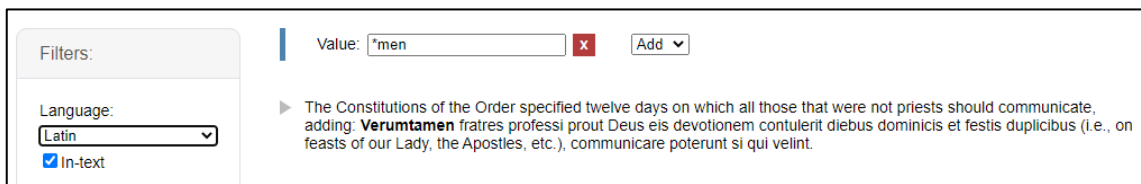


Figure 161 Filtering according to Latin on a text-level

To demonstrate filtering by date, three instances of *just* are retrieved (see Figure 162). The first is a quote from Shirley in *Ben Hur*. The publication date for *Ben Hur* is 1880, but the quote from Shirley is from 1659. The second instance is from *Pride and Prejudice*, published in 1830. The third instance is from *My Man Jeeves*, published in 1919.

Value: ✕

Semantic sense: ▼ Add ▼

- ▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
Then I must wait for justice Until it come; and they are happiest far Whose consciences may calmly wait their right." Schiller, Don Carlos (act iv., sc. xv.) BOOK FIFTH "Only the actions of the **just** Smell sweet and blossom in the dust." SHIRLEY. "And, through the heat of conflict, keeps the law, In calmness made, and sees what he foresaw." WORDSWORTH.
- ▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters. "My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?" Mr. Bennet replied that he had not. "But it is," returned she; "for Mrs. Long has **just** been here, and she told me all about it." Mr. Bennet made no answer. "Do you not want to know who has taken it?" cried his wife impatiently. "You want to tell me, and I have no objection to hearing it."
- ▼ [My Man Jeeves](#) by [P.G. Wodehouse](#) (1919)
"What, pipped?" "He gave that impression, sir." I sipped the whisky. I was sorry if Bicky was in trouble, but, as a matter of fact, I was rather glad to have something I could discuss freely with Jeeves **just** then, because things had been a bit strained between us for some time, and it had been rather difficult to hit on anything to talk about that wasn't apt to take a personal turn. You see, I had decided—rightly or wrongly—to grow a moustache and this had cut Jeeves to the quick. He couldn't stick the thing at any price, and I had been living ever since in an atmosphere of bally disapproval till I was getting jolly well fed up with it. What I mean is, while there's no doubt that in certain matters of dress Jeeves's judgment is absolutely sound and should be followed, it seemed to me that it was getting a bit too thick if he was going to edit my face as well as my costume.

Figure 162 Searching for the value just on inkling

By applying filters and searching for instances that were published between 1800 and 1900, only the first two instances are retrieved (Figure 163). This is filtering on the document level and considers the publication date of the document. If one specifies that the in-text bibliographic properties must be taken into consideration, then only the second instance is retrieved (Figure 164). Figure 165 shows that the first instance is retrieved if the publication date is changed to search for instances published before 1700 according to the detailed metadata.

Filters:

Language:

In-text

Date: to

In-text

Genre:

In-text

Author:

Value: ✕

Semantic sense: ▼ Add ▼

- ▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
Then I must wait for justice Until it come; and they are happiest far Whose consciences may calmly wait their right." Schiller, Don Carlos (act iv., sc. xv.) BOOK FIFTH "Only the actions of the **just** Smell sweet and blossom in the dust." SHIRLEY. "And, through the heat of conflict, keeps the law, In calmness made, and sees what he foresaw." WORDSWORTH.
- ▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters. "My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?" Mr. Bennet replied that he had not. "But it is," returned she; "for Mrs. Long has **just** been here, and she told me all about it." Mr. Bennet made no answer. "Do you not want to know who has taken it?" cried his wife impatiently. "You want to tell me, and I have no objection to hearing it."

Figure 163 Filtering according to date on the document-level

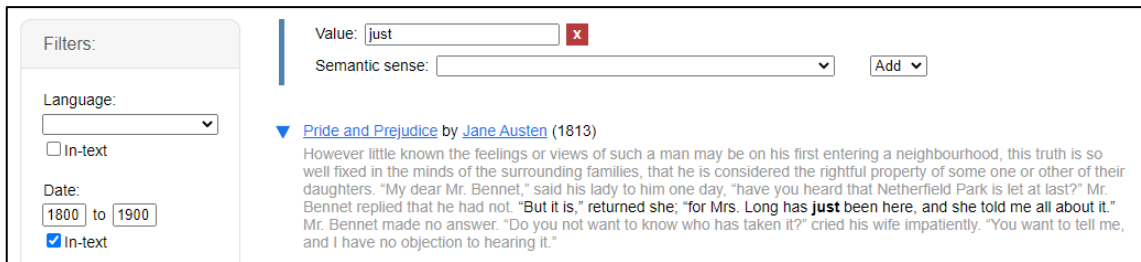


Figure 164 Filtering according to date on text-level

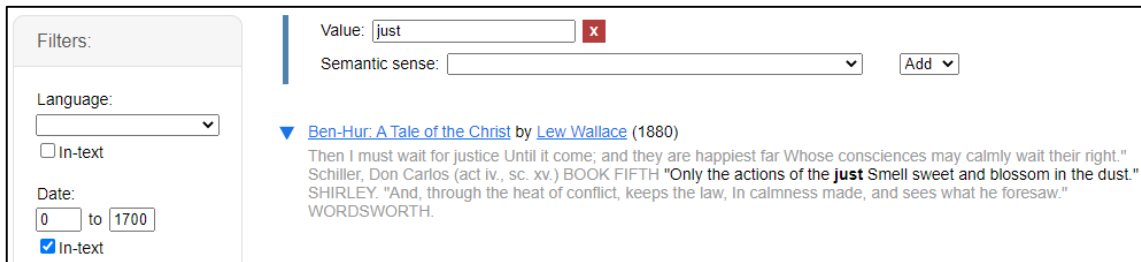


Figure 165 Filtering according to date on text-level (2)

To demonstrate filtering by genre, five instances that start with *come* are retrieved and displayed (see Figure 166). The context for these instances has been expanded. Except for the second instance, all the instances appear in the regular text of the document. The second instance is a quote from Schiller in *Ben Hur* and is evidently from a drama. The genre for the document *Ben Hur* is prose, but the genre for this quote is drama. In Figure 167 the filter has been changed to search for drama on the text-level and retrieves only the quote from Schiller. The filter is changed in Figure 168, and the filter for genre is set to prose, also on a text-level, and the quote from Shirley is not retrieved.

Filters:

Language: In-text

Date: to In-text

Genre: In-text

Author: In-text

Publisher:

Publication place:

Subject:

Title:

Case-sensitive

Value:

- ▼ [Clouds of Witness](#) by [Dorothy L. Sayers](#) (1958)
"That's an idea," he said. "There were occasions—mild ones, but Helen would make the most of them." He whistled thoughtfully. "Still, when it **comes** to the gallows—" "Do you suppose, Wimsey, that your brother really contemplates the gallows?" asked Parker. "I think Murbles put it to him pretty straight," said Lord Peter. "Quite so."
- ▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
Should the monarch prove unjust— And, at this time— Queen. Then I must wait for justice Until it **come**; and they are happiest far Whose consciences may calmly wait their right." Schiller, Don Carlos (act iv., sc. xv.) BOOK FIFTH "Only the actions of the just Smell sweet and blossom in the dust."
- ▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
"Is that his design in settling here?" "Design! Nonsense, how can you talk so! But it is very likely that he may fall in love with one of them, and therefore you must visit him as soon as he **comes**." "I see no occasion for that. You and the girls may go, or you may send them by themselves, which perhaps will be still better, for as you are as handsome as any of them, Mr. Bingley may like you the best of the party." "My dear, you flatter me."
- ▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
I certainly have had my share of beauty, but I do not pretend to be anything extraordinary now. When a woman has five grown-up daughters, she ought to give over thinking of her own beauty." "In such cases, a woman has not often much beauty to think of." "But, my dear, you must indeed go and see Mr. Bingley when he **comes** into the neighbourhood." "It is more than I engage for, I assure you."
- ▼ [The Life of St. Teresa of Jesus, of the Order of Our Lady of Carmel](#) by [Teresa of Avila](#) (1904)
Mr. Lewis possessed a copy of this photographic reproduction, but utilised it only in one instance in his second edition. 1. Chap. xxxiv., note 5. The publication of the autograph has settled a point of some importance. The Bollandists (n. 1520), discussing the question whether the headings of the chapters (appended to this Introduction) are by St. Teresa or a later addition, **come** to the conclusion (against the authors of the Reforma de los Descalços) that they are clearly an interpolation (clarissime patet) on account of the praise of the doctrine contained in these arguments. Notwithstanding their high authority the Bollandists are in this respect perfectly wrong, the arguments are entirely in St. Teresa's own hand and are exclusively her own work. The Book of Foundations and the Way of Perfection contain similar arguments in the Saint's handwriting. Nor need any surprise be felt at the alleged praise of her doctrine for by saying: this chapter is most noteworthy (Chap. XIV.), or: this is good doctrine (Chap. XXI.), etc., she takes no credit for herself because she never grows tired of repeating that she only delivers the message she has received from our Lord.

Figure 166 Searching for all instances that begin with come

Filters:

Language: In-text

Date: to In-text

Genre: In-text

Value:

- ▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
Should the monarch prove unjust— And, at this time— Queen. Then I must wait for justice Until it **come**; and they are happiest far Whose consciences may calmly wait their right." Schiller, Don Carlos (act iv., sc. xv.) BOOK FIFTH "Only the actions of the just Smell sweet and blossom in the dust."

Figure 167 Filtered according to genre on text-level

Filters:

Language: In-text

Date: to In-text

Genre: In-text

Author: In-text

Value:

- ▼ [Clouds of Witness](#) by [Dorothy L. Sayers](#) (1958)
"That's an idea," he said. "There were occasions—mild ones, but Helen would make the most of them." He whistled thoughtfully. "Still, when it **comes** to the gallows—" "Do you suppose, Wimsey, that your brother really contemplates the gallows?" asked Parker. "I think Murbles put it to him pretty straight," said Lord Peter. "Quite so."
- ▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
"Is that his design in settling here?" "Design! Nonsense, how can you talk so! But it is very likely that he may fall in love with one of them, and therefore you must visit him as soon as he **comes**." "I see no occasion for that. You and the girls may go, or you may send them by themselves, which perhaps will be still better, for as you are as handsome as any of them, Mr. Bingley may like you the best of the party." "My dear, you flatter me."
- ▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
I certainly have had my share of beauty, but I do not pretend to be anything extraordinary now. When a woman has five grown-up daughters, she ought to give over thinking of her own beauty." "In such cases, a woman has not often much beauty to think of." "But, my dear, you must indeed go and see Mr. Bingley when he **comes** into the neighbourhood." "It is more than I engage for, I assure you."

Figure 168 Filtered according to genre on text-level (2)

In Figure 169 all instances that start with see and were written by Wordsworth are retrieved. The filter is set to text-level. In Figure 170 all instances that start with *de* written by Shakespeare are retrieved, also on a text-level. The filters that use 'text input fields' to receive the users input, work with regular expressions and will match search text with the values in the database. This means that one does not have to type out the whole name of the author as is evident in Figure 170.

Filters:

Value:

Language:
 In-text

Date: to
 In-text

Genre:
 In-text

Author:
 In-text

▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
 BOOK FIFTH "Only the actions of the just Smell sweet and blossom in the dust." SHIRLEY. "And, through the heat of conflict, keeps the law, In calmness made, and **sees** what he foresaw." WORDSWORTH. CHAPTER I The morning after the bacchanalia in the saloon of the palace, the divan was covered with young patricians.

Figure 169 Searching for instances that start with see written by Wordsworth

Filters:

Value:

Language:
 In-text

Date: to
 In-text

Genre:
 In-text

Author:
 In-text

▼ [Clouds of Witness](#) by [Dorothy L. Sayers](#) (1958)
 CHAPTER "OF HIS MALICE AFORETHOUGHT" "O, Who hath done this **deed**?" Othello Lord Peter Wimsey stretched himself luxuriously between the sheets provided by the Hôtel Meurice. After his exertions in the unravelling of the Battersea Mystery, he had followed Sir Julian Freke's advice and taken a holiday.

▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
 Our size of sorrow, Proportion'd to our cause, must be as great As that which makes it. -- Enter, below, DIOMEDES. How now? **is he dead?** Diomedes. His death's upon him, but not dead." Antony and Cleopatra (act iv., sc. xiii.).

▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
 How now? is he dead? Diomedes. His **death's** upon him, but not dead." Antony and Cleopatra (act iv., sc. xiii.). BOOK FOURTH "Alva.

▼ [Ben-Hur: A Tale of the Christ](#) by [Lew Wallace](#) (1880)
 How now? is he dead? Diomedes. His **death's** upon him, but not **dead**." Antony and Cleopatra (act iv., sc. xiii.). BOOK FOURTH "Alva.

Figure 170 Searching for instances that start with de written by Shakespeare

The last filtering options will be demonstrated in one example. In Figure 171 all instances of *dear* are retrieved, but filtered by publisher (Egerton), publication place (Whitehall), subject (Social classes) and title of the document (Pride and Prejudice).

Filters:

Language: In-text

Date: to In-text

Genre: In-text

Author: In-text

Publisher:

Publication place:

Subject:

Title:

Case-sensitive

Value: Semantic sense:

▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
Chapter 1 It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife. However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters. **"My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?"** Mr. Bennet replied that he had not. "But it is," returned she; "for Mrs. Long has just been here, and she told me all about it." Mr. Bennet made no answer.

▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
"Do you not want to know who has taken it?" cried his wife impatiently. "You want to tell me, and I have no objection to hearing it." This was invitation enough. "Why, **my dear**, you must know, Mrs. Long says that Netherfield is taken by a young man of large fortune from the north of England; that he came down on Monday in a chaise and four to see the place, and was so much delighted with it, that he agreed with Mr. Morris immediately; that he is to take possession before Michaelmas, and some of his servants are to be in the house by the end of next week." "What is his name?" "Bingley." "Is he married or single?"

▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
"Bingley." "Is he married or single?" "Oh! **Single, my dear**, to be sure! A single man of large fortune; four or five thousand a year. What a fine thing for our girls!" "How so?"

▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
What a fine thing for our girls! "How so? How can it affect them?" **"My dear Mr. Bennet,"** replied his wife, "how can you be so tiresome! You must know that I am thinking of his marrying one of them." "Is that his design in settling here?" "Design!"

▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
But it is very likely that he may fall in love with one of them, and therefore you must visit him as soon as he comes." "I see no occasion for that. You and the girls may go, or you may send them by themselves, which perhaps will be still better, for as you are as handsome as any of them, Mr. Bingley may like you the best of the party." **"My dear, you flatter me.** I certainly have had my share of beauty, but I do not pretend to be anything extraordinary now. When a woman has five grown-up daughters, she ought to give over thinking of her own beauty." "In such cases, a woman has not often much beauty to think of."

▼ [Pride and Prejudice](#) by [Jane Austen](#) (1813)
I certainly have had my share of beauty, but I do not pretend to be anything extraordinary now. When a woman has five grown-up daughters, she ought to give over thinking of her own beauty." "In such cases, a woman has not often much beauty to think of." **"But, my dear, you must indeed go and see Mr. Bingley when he comes into the neighbourhood."** "It is more than I engage for, I assure you."

Figure 171 Filtered according to publisher, publication place, subject and title

6.3.14. Combining search options

It is possible to combine the search options and so allow for complex searches. In Figure 172 all instances that end in *ed*, that are verbs in the past tense, that are in direct speech and written by Wodehouse are retrieved.

Filters:

Language: In-text

Date: to In-text

Genre: In-text

Author: In-text

Value: Part-of-speech:

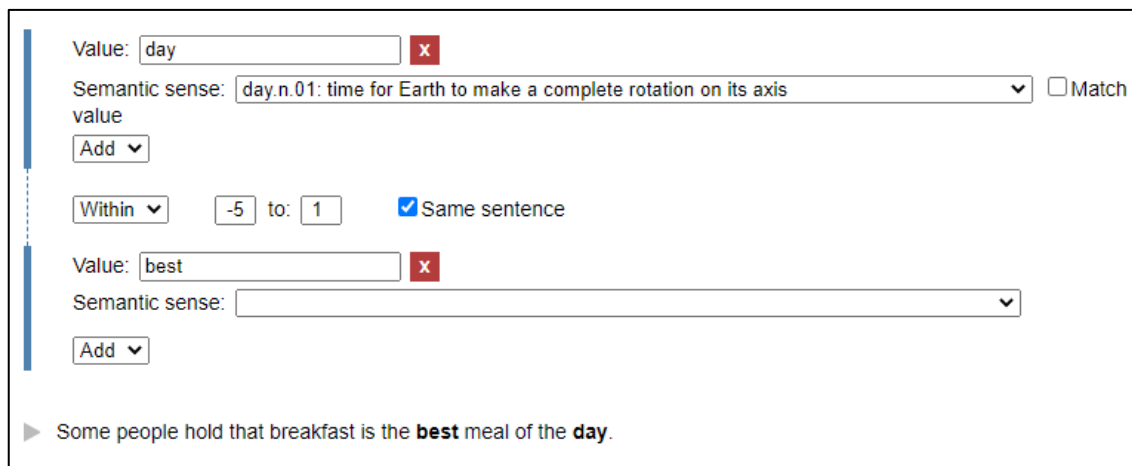
Direct

▼ [My Man Jeeves](#) by [P.G. Wodehouse](#) (1919)
And, what's more, he can always be counted on to extend himself on behalf of any pal of mine who happens to be to all appearances knee-deep in the bouillon. Take the rather rummy case, for instance, of dear old Bicky and his uncle, the hard-boiled egg. It happened after I had been in America for a few months. **I got back to the flat latish one night, and when Jeeves brought me the final drink he said: "Mr. Bickersteth called to see you this evening, sir, while you were out."** "Oh?" I said. "Twice, sir. He appeared a trifle agitated."

▼ [My Man Jeeves](#) by [P.G. Wodehouse](#) (1919)
I got back to the flat latish one night, and when Jeeves brought me the final drink he said: "Mr. Bickersteth called to see you this evening, sir, while you were out." "Oh?" I said. "Twice, sir. **He appeared a trifle agitated.**" "What, pipped?" "He gave that impression, sir." I sipped the whisky.

Figure 172 Combining search options, example 1

In Figure 173, the selection is for all instances of *day* where the meaning is *time for Earth to make a complete rotation on its axis* (not, for example, *some point or period in time* or other meaning), is within five tokens of instances that start with *best*, still within the same sentence. Furthermore, the text-level language must be English, and the genre of the document must be prose.



Value: ✕

Semantic sense: Match value

▼

to: Same sentence

Value: ✕

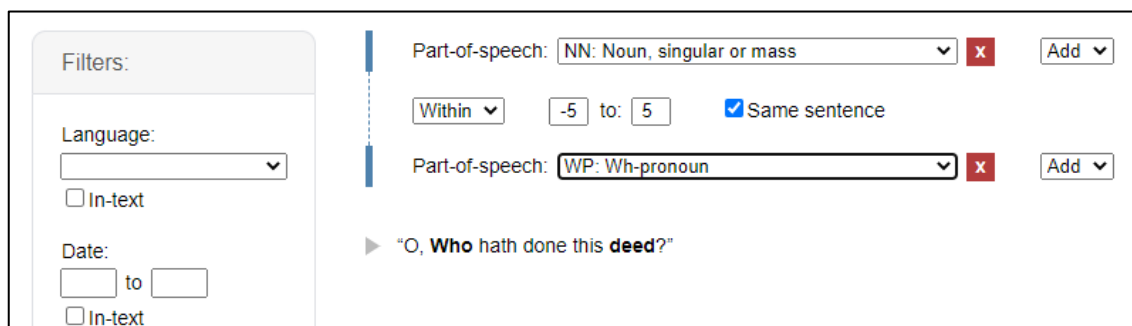
Semantic sense:

▼

► Some people hold that breakfast is the **best** meal of the **day**.

Figure 173 Combining search options, example 2

Another example is given in Figure 174. Here the user is searching for all nouns that are within five tokens (in either direction) of a wh-pronoun.



Filters:

Language:

In-text

Date: to

In-text

Part-of-speech: ✕ ▼

to: Same sentence

Part-of-speech: ✕ ▼

► "O, **Who** hath done this **deed**?"

Figure 174 Combining search options, example 3

6.3.15. Searching using a query language

The search field at the top of the page could be used to search by entering text, instead of using the graphical user interface. This field could incorporate features from a query language. Searching by using a query language has not been implemented in the current version of the tool, but the manner in which it could be used is demonstrated.

The query that is formed by using the graphical user interface is displayed in text form in the search field. This text form uses features from a query language. In Figure 175 a query has been created where the user searches for instances of the word *flatter* within

15 words of the word *beauty*, but not limited to the same sentence. This query is displayed in text format in the search field as *flatter <1..15!> beauty*.

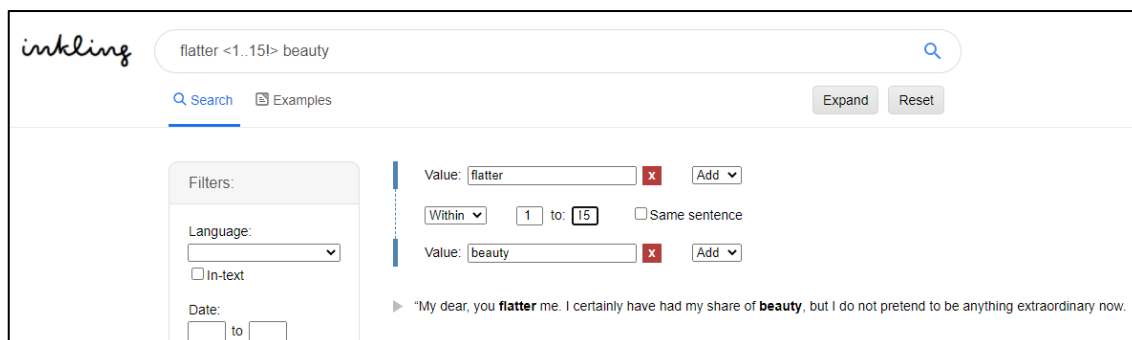


Figure 175 Searching by using a query language

The different search options that can be expressed in a query language will be discussed in this section.

To search for a single word, a user must simply type the word. Truncation in the search field is used in the same way as in the graphical user interface. To search for inflected forms, *_INF* is appended to the end of a verb, for example, *reply_INF*, which will retrieve *replied*, *replying*. To search for a part-of-speech, the part-of-speech tag is appended to the word, for example, *fall_VB*, which will retrieve instances of *fall* where it is used as a verb (base form) and not as other part-of-speech categories. Phrases can simply be entered, for example *my dear*. For words that are near other words, proximity is entered in angular brackets between words. The default is to search towards the right of a word; a minus is used to indicate that the direction is leftwards. An exclamation mark is used to indicate that searching should span sentence boundaries (i.e. not in the same sentence). The example in Figure 175 demonstrates searching for words near other words. Searching for syntactic dependencies is indicated by using *=>*. The direction changes to indicate the governor or dependant. *Good <= morning* searches for instances where *good* modifies *morning*, or *competent => dashed* searches for instances where *competent* is modified by *dashed*. The type of dependency can be inserted between the angular bracket and the equals sign, for example, *<amod=*. The semantic sense (from WordNet) can be appended to the search term, for example, *man.valet.n.01*. To filter according to functional properties, codes for the functional properties can be appended to the word that is being searched for, for example *lady_PAR* searches for instances of *lady* that appears in a paragraph. Filtering according to bibliographic properties is not currently supported in the query language.

Entering search statements as text using this query language will enable an advanced user to search quickly and efficiently. The advanced user will not be delayed by opening dropdown menus, searching for the correct value, creating new sections and other such features from a graphical user interface. A search can quickly be entered and executed. However, this will mean that the user will have to learn the syntax of the query language employed by this tool.

In an effort to help a user to learn the query language, the query that was formulated through the graphical user interface is displayed in the search field. This will enable a user to learn by looking at examples.

6.4. Conclusion

In this chapter, the prototype that was developed to evaluate whether fine-grained metadata can lead to improved retrieval was presented. The technical implementation was discussed, as well as various ways of using the tool to search for words or phrases. The next step is to evaluate the tool using the criteria from chapter 2, namely, interface design, metadata, search options, filtering, search results, complexity of use and help files.

As was explained earlier, the tool that was developed is a prototype, not a complete product. Therefore, the visual design of the system was not of primary importance and the tool does not contain graphics or other visual elements. Nonetheless, the interface is clean and clear. The search bar is at the top of the screen and clearly visible. Once the search bar is activated, a user should be able to start a search immediately. If a user is not confident about using the query language, but would like to build a complex query, the graphical user interface is available to construct a query. The graphical interface uses descriptive labels that are easy to understand.

Some further features were included to make it easier to use the tool. There is a reset button that will clear the search and restore the tool to its default view (Figure 176). There is an expand button to view the extended context of all the examples (Figure 176), and similarly a collapse button to see the results with minimal context (Figure 177).

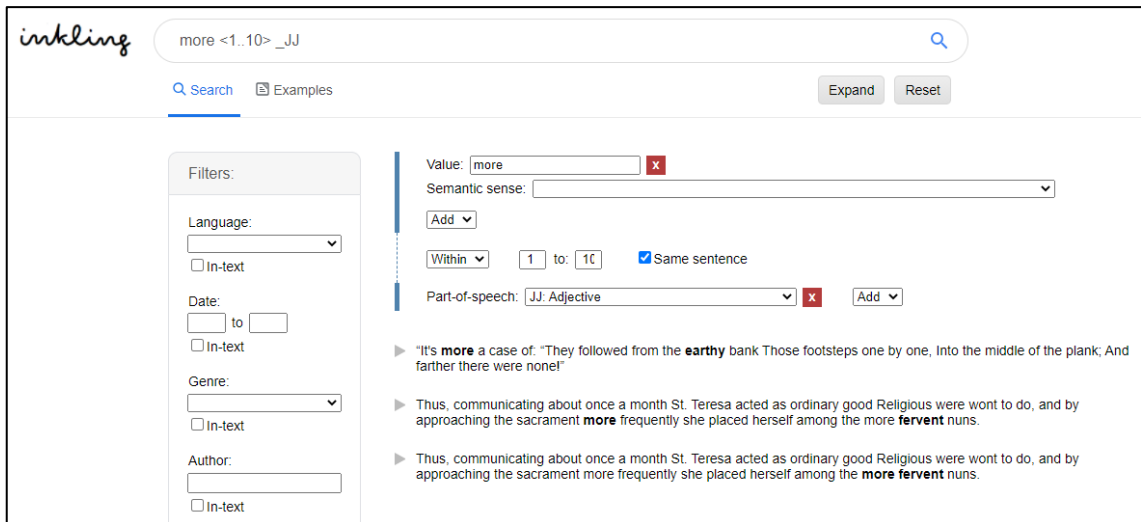


Figure 176 Extra features in inking

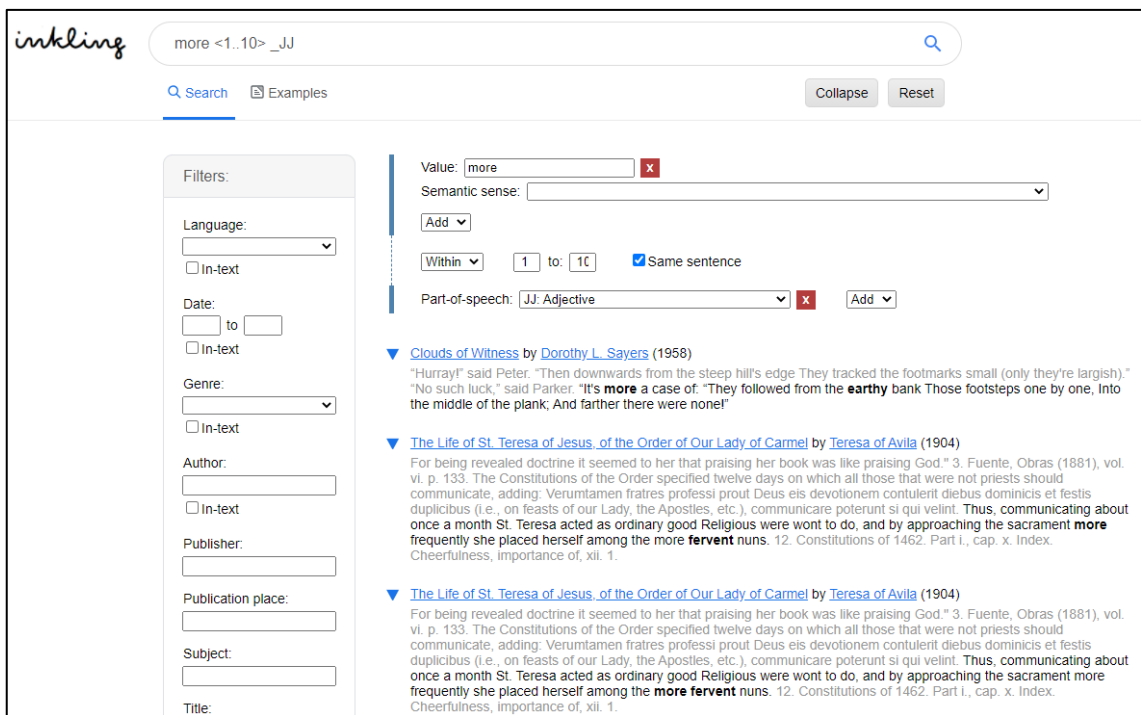


Figure 177 The Collapse button in inking

The metadata in the corpus for this tool have been discussed in chapter 4. The texts are encoded with morphological, syntactic, semantic, functional and bibliographic metadata. In addition to document-level bibliographic metadata, in-text bibliographic metadata are available.

In this chapter, the different search options in the tool were discussed. In summary, it is possible to search for words and phrases. It is also possible to search for words near other words and to specify if the search should cross sentence boundaries. It is possible to use wildcard characters at the beginning or ending of words to search for

variations. It is possible to search for inflected forms. The tool allows a user to specify the part-of-speech category to use in a search. It is possible to search for words that modify other words (dependency relationships) and to search for words with a specific meaning. It was highlighted that to gain the full benefit of searching using the morphological and syntactic metadata a user would need a certain degree of linguistic knowledge.

It is possible to search only in certain areas of a text, for example, only in headings or back matter.

Filtering to a detailed level is allowed. A user can filter according to coarse document-level bibliographic metadata, as well as fine-grained in-text bibliographic metadata. This means that a user is not limited to filtering on the level of the document, but can filter on a more detailed level.

The results are displayed in list form. The keywords are highlighted, and minimal context is given. More context is available, and a user can link to the original text. There is currently no option to view the relative frequency of instances over time displayed in a graph. As this is a prototype, there is not enough data in the tool to be able to create a graph. However, this is a feature that could be implemented if more encoded data are added to the tool.

All queries can be executed by using the graphical user interface, which means that a user does not have to learn complex syntax in order to be able to use the tool. Nor is the user required to understand the metadata in the corpus to be able to search in the tool. However, the advanced search features are available in the graphical user interface.

There are currently no help files in the prototype, as it is still a proof of concept. However, to demonstrate how users could be taught how to use the system easily, various examples have been worked out. These examples are prepopulated queries. A user clicks on an example (Figure 178) and is taken to the search page with the query already populated in the tool (Figure 179). This means a user can see how different queries can be executed.

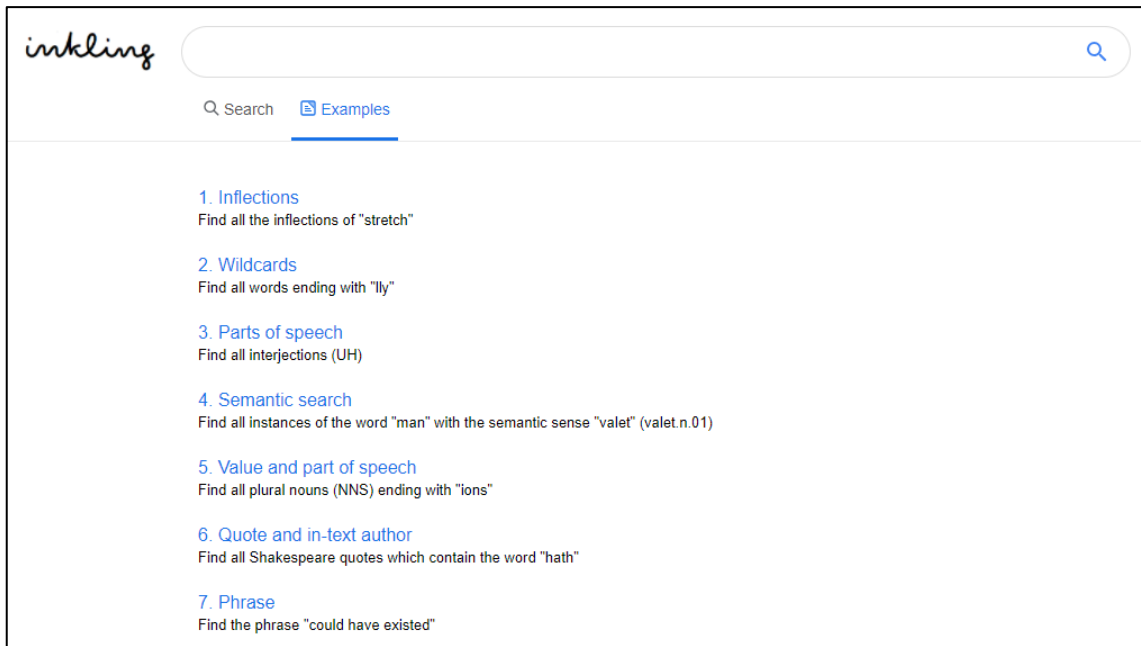


Figure 178 Examples in inkling

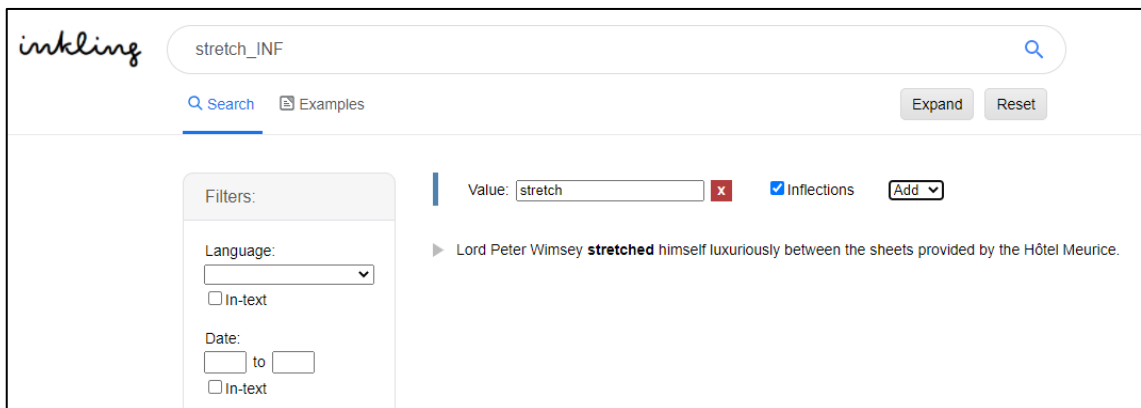


Figure 179 A prepopulated query

This prototype demonstrates that by encoding texts with detailed metadata, improved retrieval is possible. At the very basic level, bibliographic metadata are necessary to understand the composition of the corpus. This tool allows a user to filter using bibliographic data to create exactly the subset of data that will be needed for a specific query. Furthermore, a user does not have to be limited to the metadata that are available for a document. Different sections in a text can be encoded and, where necessary, bibliographic metadata can be applied to the sections in a text. This tool demonstrates that these detailed metadata enable users to search in more specific subsets of data and work on a lower level than the document. Additional metadata about words can be used to refine a search to a greater extent. This tool demonstrates how semantic, syntactic and morphological data can be incorporated to create a very

powerful search tool. It is clear that fine-grained metadata enable users to ask more complex and very specific questions and possibly get to more trustworthy answers.

In order for the tool to be useful, it needs to have documents that are encoded with the metadata described in chapter 4. The encoding is time intensive. The possibility of using software to automate encoding is investigated in the next chapter.

7. Automated processing of text

Humans are toolmakers by nature, but most of us can't build or modify software — arguably our most powerful tool. We're on a mission to make it possible for everyone to shape the tools that shape their lives. Then we'll all be able to tackle the world's problems better, together.

Notion Labs

7.1. Introduction

In the previous chapters, it has been demonstrated that enhanced information retrieval is possible, when the underlying texts are encoded with metadata that make structures and characteristics of the texts explicit, so that a computer can process the data.

Unfortunately, manual encoding of large collections is time consuming and expensive (Roller et al., 2016: 69; Wissler et al., 2014: 1). As a result, the possibility of automated annotation or encoding should be investigated. Collobert et al. (2011: 2494) ask the question, “[will] a computer program ever be able to convert a piece of English text into a programmer friendly data structure that describes the meaning of the natural language text?” Good progress has been made on the automation of some grammatical, lexical and syntactic information; however, automating the annotation of higher levels of linguistic processing is a complex exercise (Hovy & Lavid, 2010) and focus in research seems to be shifting from simpler annotations to more complex types of automated encoding (Zeyrek et al., 2019).

The purpose of this chapter is to investigate the extent to which the encoding of a piece of text could be automated, according to the encoding suggested in this study, using current technologies. The discussion will be structured according to the different levels of encoding of the texts in this study, namely, morphological, syntactic, semantic, functional and bibliographic. The extent to which the encodings of each of these levels can be done automatically will be discussed.

In chapter 2, a literature review of automated encoding was done. In this chapter, the researcher will use available tools to see to what extent current tools can be used to automate the encoding of texts. In this chapter, several examples will be encoded using software and the results will be reviewed. Some areas are well-developed and have various tools publicly available, in other areas little work has been done and there are no (or few) tools publicly available for automated encoding. Therefore, only some of the levels will be tested for automated encoding.

7.2. Encoding texts with existing tools

For the morphological level, the researcher will consider the extent according to which the tools can tokenise sentences and words, identify the lemma, and identify part-of-speech categories. In terms of syntactic parsing, the researcher will consider whether the tool can identify the head (governor) of each word and the type of dependency. On the semantic level, the appropriate WordNet sense should be assigned. There is research being done to be able to identify elements on the functional level, as was discussed in chapter 2, but there is no integrated tool yet that will address the concerns of this study. Therefore, no automated encoding of the elements on the functional level will be done. Neither will the encoding of bibliographic elements be automated in this section, as no appropriate tools are available.

The purpose of this chapter is not to do a comprehensive, statistical evaluation of all available tools or software to do automated encoding. Real comparison and evaluation for some tools, such as syntactic dependencies or semantic encoding, are complex and are beyond the scope of this study. The purpose of this chapter is to provide illustrative examples and discuss the results.

7.2.1. Morphological encoding

The two tools that will be considered for this level are Stanford CoreNLP and spaCy. These two tools were chosen as they cover much of the desired encoding in one library and they are fairly widely used. In the study by Finlayson (2015), Stanford annotators were used without any explanation as to why those annotators and not others were used. Furthermore, they are fairly accessible to someone who is not an expert programmer. Some programming is required, but not extensive.

Stanford CoreNLP (Manning et al., 2014) provides a set of tools that processes human language. It integrates many of the natural language tools developed by Stanford, namely, the part-of-speech (POS) tagger, the named entity recogniser (NER), the parser, the coreference resolution system, sentiment analysis, bootstrapped pattern learning, and the open information extraction tools.

The package works on Windows, Linux and MacOS. It is written in Java and requires Java 1.8+. Stanford CoreNLP can be used from the command-line, a Java programmatic API, an object-oriented simple API, or third-party APIs, or via a web service. Stanford CoreNLP is available to download from the CoreNLP site (<https://stanfordnlp.github.io/CoreNLP/index.html>). The package is a zip file which contains the files required to run CoreNLP.

For the purposes of this study version 3.9.2 of Stanford CoreNLP was downloaded. The latest changes to the package were made on 5 October 2018. The computer this study was conducted on ran Windows 10 and had Java 1.8 installed.

The researcher opted to use the command line interface to interact with Stanford CoreNLP. The following annotators were used in this study: tokenisation, sentence splitting, part-of-speech tagging, lemmatisation and parsing.

spaCy (spaCy, n.d.) is an open-source software library for advanced natural language processing. Some of the features include non-destructive tokenisation, named entity recognition, part-of-speech tagging, labelled dependency parsing and syntax-driven sentence segmentation.

spaCy runs on Unix/Linux, macOS/OS X and Windows. It is written in Python and Cython. There are various options to install spaCy, one being PIP (a Python package manager). If using Windows, it is necessary to also install the Visual C++ Build Tools or Visual Studio Express that were used to compile the Python interpreter of the particular installation. After installation, it is necessary to download a language model. There are three pretrained statistical models for English available.

For the purposes of this study spaCy (version 2.1.8) was installed using PyCharm, an integrated development environment for the Python programming language. Python 3.7 was used. The “en_core_web_sm” language model was selected for this study. It is an English model, trained on OntoNotes. It assigns context-specific token vectors, part-of-speech tags, named entities and does dependency parsing.

It should be noted that the libraries were tested with no changes to any parameters or training. Training the libraries with different models could produce different results.

Encoded examples

Stanford CoreNLP and spaCy were used to encode various examples. These examples were selected to see to what extent the two tools can encode on the morphological level. These examples are as follows:

- Simple sentence
- Simple sentence with inflected forms
- A sentence with the same word as different part-of-speech category
- Sentence with quotation marks, apostrophes (contraction) and hyphens
- Sentence with spelling to indicate dialect

- Paragraph
- Full page

Each of these examples will be discussed. (The Penn Treebank tagset that is used for this encoding is shown in Table 2 in chapter 2.) The encoding for each example is given in a table. Errors are highlighted in grey and comments are given in a column.

Simple sentence

A person may be proud without being vain.

This sentence was taken from *Pride and Prejudice* by Jane Austen.

Table 14 Morphological annotation 1 – Stanford

Stanford			
Word	Lemma	Part-of-speech	Comment
A	a	DT	The tokenisation, lemma identification and part-of-speech tagging was done correctly by Stanford.
person	person	NN	
may	may	MD	
be	be	VB	
proud	proud	JJ	
without	without	IN	
being	be	VBG	
vain	vain	JJ	
.	.	.	

Table 15 Morphological annotation 1 – spaCy

spaCy			
Word	Lemma	Part-of-speech	Comment
A	a	DT	The tokenisation, lemma identification and part-of-speech tagging was done correctly by spaCy.
person	person	NN	
may	may	MD	
be	be	VB	
proud	proud	JJ	
without	without	IN	
being	be	VBG	
vain	vain	JJ	
.	.	.	

Simple sentence with inflected forms

And he threw down half a sovereign, and caught up his coat.

This sentence from *The Innocence of Father Brown* by G.K. Chesterton was selected to evaluate the handling of inflected forms.

Table 16 Morphological annotation 2 – Stanford

Stanford			
Word	Lemma	Part-of-speech	Comment
And	and	CC	
he	he	PRP	
threw	throw	VBD	Inflections are handled correctly, and the correct lemma is identified.
down	down	RP	
half	half	PDT	
a	a	DT	
sovereign	sovereign	JJ	Incorrect part-of-speech, should be a noun (NN).
,	,	,	
and	and	CC	
caught	catch	VBD	Inflections are handled correctly, and the correct lemma is identified.
up	up	RP	
his	he	PRP\$	The correct lemma is identified.
coat	coat	NN	
.	.	.	

Table 17 Morphological annotation 2 – spaCy

spaCy			
Word	Lemma	Part-of-speech	Comment
And	and	CC	
he	-PRON-	PRP	In spaCy the lemma for a pronoun is indicated as -PRON-.
threw	throw	VBD	Inflections are handled correctly, and the correct lemma is identified.
down	down	RP	
half	half	PDT	
a	a	DT	
sovereign	sovereign	NN	
,	,	,	
and	and	CC	
caught	catch	VBD	Inflections are handled correctly, and the correct lemma is identified.
up	up	RP	
his	-PRON-	PRP\$	In spaCy the lemma for a pronoun is indicated as -PRON-.

coat	coat	NN	
.	.	.	

A sentence with the same word as different part-of-speech category

Well, she was well, but she found herself falling down a very deep well.

This sentence was adapted from *Alice in Wonderland* by Lewis Carroll, to include three senses (meanings) of the word “well”.

Table 18 Morphological annotation 3 – Stanford

Stanford			
Word	Lemma	Part-of-speech	Comment
Well	well	RB	Incorrect part-of-speech, it should be an interjection (UH), not an adverb.
,	,	,	
she	she	PRP	
was	be	VBD	
well	well	RB	The correct part-of-speech category is identified.
,	,	,	
but	but	CC	
she	she	PRP	
found	find	VBD	
herself	herself	PRP	
falling	fall	VBG	
down	down	RP	
a	a	DT	
very	very	RB	
deep	deep	RB	Incorrect part-of-speech, it should be an adjective (JJ), not an adverb.
well	well	RB	Incorrect part-of-speech, it should be a noun (NN), not an adverb.
.	.	.	

Table 19 Morphological annotation 3 – spaCy

spaCy			
Word	Lemma	Part-of-speech	Comment
Well	well	UH	The correct part-of-speech category is identified.
,	,	,	
she	-PRON-	PRP	In spaCy the lemma for a pronoun is indicated as -PRON-.
was	be	VBD	

well	well	RB	The correct part-of-speech category is identified.
,	,	,	
but	but	CC	
she	-PRON-	PRP	In spaCy the lemma for a pronoun is indicated as -PRON-.
found	find	VBD	
herself	-PRON-	PRP	In spaCy the lemma for a pronoun is indicated as -PRON-.
Falling	fall	VBG	
down	down	RP	
a	a	DT	
very	very	RB	
deep	deep	JJ	The correct part-of-speech category is identified.
well	well	RB	Incorrect part-of-speech, it should be a noun (NN), not an adverb.
.	.	.	

Sentence with quotation marks, apostrophes (contraction) and hyphens

“Really, you’re as good as a three-act farce.”

This sentence from *The Innocence of Father Brown* by G.K. Chesterton was selected to evaluate the handling of quotation marks, apostrophes and hyphens.

Table 20 Morphological annotation 4 – Stanford

Stanford			
Word	Lemma	Part-of-speech	Comment
“	“	“	Punctuation handled correctly throughout the whole sentence.
Really	really	RB	
,	,	,	
you	you	PRP	
're	be	VBP	Contraction handled correctly.
as	as	IN	Incorrect part-of-speech, it should be an adverb (RB), not a preposition.
good	good	JJ	
as	as	IN	
a	a	DT	
three-act	three-act	JJ	Hyphenated word is kept as one.
farce	farce	NN	
.	.	.	
”	”	”	

Table 21 Morphological annotation 4 – spaCy

spaCy			
Word	Lemma	Part-of-speech	Comment
"	"	NNP	Strangely, the opening quotation mark is tagged as a proper noun. This could be as a result of curly quotation marks. When these were replaced with neutral, straight quotations marks, they were correctly tagged as quotation marks.
Really	really	RB	
,	,	,	
you	-PRON-	PRP	
're	be	VBP	Contraction handled correctly.
as	as	RB	
good	good	JJ	
as	as	IN	
a	a	DT	
three	three	CD	The adjective "three-act" was broken into three separate tokens and combination not analysed.
-	-	HYPH	
act	act	NN	
farce	farce	NN	
.	.	.	
"	"	"	

Sentence with spelling to indicate dialect

I'd 'ave caught the fool but for havin' to pick 'em up.

This sentence from *The Innocence of Father Brown* by G.K. Chesterton was selected to evaluate the handling of spelling used to indicate dialect.

Table 22 Morphological annotation 5 – Stanford

Stanford			
Word	Lemma	Part-of-speech	Comment
I	I	PRP	
'd	would	MD	
`	`	``	The apostrophe in this example is not a quotation but used to indicate the pronunciation of a word.
ave	ave	FW	The lemma should be "have". Incorrect part-of-speech, it should be a verb, base form (VB).
caught	catch	VBD	Incorrect part-of-speech, it should be a past participle (VBN).
the	the	DT	
fool	fool	NN	

but	but	CC	
for	for	IN	
havin	havin	NN	The lemma should be “have”. Incorrect part-of-speech, it should be a present participle (VBG).
'	'	"	The apostrophe in this example is not a quotation but used to indicate the pronunciation of a word.
to	to	TO	
pick	pick	VB	
`	`	``	The apostrophe in this example is not a quotation but used to indicate the pronunciation of a word.
em	em	FW	The lemma should be “them”. Incorrect part-of-speech, it should be a personal pronoun (PRP).
up	up	RP	
.	.	.	

Table 23 Morphological annotation 5 – spaCy

spaCy			
Word	Lemma	Part-of-speech	Comment
I	-PRON-	PRP	In spaCy the lemma for a pronoun is indicated as -PRON-.
'd	would	MD	
'	'	ADD	The apostrophe in this example is used to indicate the pronunciation of a word. According to the spaCy documentation the tag ADD is used to indicate email.
ave	ave	NN	The lemma should be “have”. Incorrect part-of-speech, it should be a verb, base form (VB).
caught	catch	VBN	
the	the	DT	
fool	fool	NN	
but	but	CC	
for	for	IN	
havin'	have	CD	Incorrect part-of-speech, it should be a present participle (VBG).
to	to	TO	
pick	pick	VB	
'	'	DT	The apostrophe in this example is not a determiner but used to indicate the pronunciation of a word.
em	-PRON-	PRP	The researcher is unsure why the lemma is identified as such
up	up	RP	
.	.	.	

Paragraph

Anyone passing the house on the Thursday before Whit-Sunday at about half-past four p.m. would have seen the front door open, and Father Brown, of the small church of St. Mungo, come out smoking a large pipe in company with a very tall French friend of his called Flambeau, who was smoking a very small cigarette. These persons may or may not be of interest to the reader, but the truth is that they were not the only interesting things that were displayed when the front door of the white-and-green house was opened. There are further peculiarities about this house, which must be described to start with, not only that the reader may understand this tragic tale, but also that he may realise what it was that the opening of the door revealed.

This paragraph from *The Innocence of Father Brown* by G.K. Chesterton was selected to evaluate the handling of a paragraph.

Stanford

The three sentences were identified correctly. Due to space restrictions, the complete part-of-speech tags will not be given here. However, the errors will be noted in the table below.

Table 24 Morphological annotation 6 – Stanford

Stanford			
Word	Lemma	Part-of-speech	Comment
Anyone	anyone	NN	Incorrect part-of-speech, it should be a pronoun (PRP).
half-past	half-past	JJ	It is interesting to note that the OED does not recognise half-past as a word, and it is often written without a hyphen.
come	come	VBN	Incorrect part-of-speech, it should be a verb, base form (VB).
smoking	smoking	NN	Incorrect part-of-speech, it should be a present participle (VBG).
white-and-green	white-and-green	JJ	Stanford keeps hyphenated words as single words.
further	further	RB	Incorrect part-of-speech, it should be an adjective (JJ).

spaCy

The three sentences were correctly identified. Due to space restrictions, the complete part-of-speech tags will not be given here. However, the errors will be noted in the table.

Table 25 Morphological annotation 6 – spaCy

spaCy			
Word	Lemma	Part-of-speech	Comment
Anyone	Anyone	NN	Incorrect part-of-speech, it should be a pronoun (PRP).
Whit	Whit	NNP	This is one word.
-	-	HYPH	
Sunday	Sunday	NNP	
half	half	JJ	It is interesting to note that the OED does not recognise half-past as a word, and it is often written without a hyphen.
-	-	HYPH	
past	past	JJ	
his	-PRON-	PRP\$	In spaCy the lemma for a pronoun is indicated as -PRON-.
to	to	IN	Incorrect part-of-speech, it should be “to” (TO).
they	-PRON-	PRP	In spaCy the lemma for a pronoun is indicated as -PRON-.
he	-PRON-	PRP	In spaCy the lemma for a pronoun is indicated as -PRON-.

A full page

The following page from *Clouds of Witness* by Dorothy L. Sayers was selected to test the ability of the tools to handle a section of text and primarily how it would do the sentence segmentation. This example posed some challenges as it contains headings and quotations at the beginning.

CHAPTER I

“OF HIS MALICE AFORETHOUGHT”

“O, Who hath done this deed?”

Othello

Lord Peter Wimsey stretched himself luxuriously between the sheets provided by the Hôtel Meurice. After his exertions in the unravelling of the Battersea Mystery, he had followed Sir Julian Freke's advice and taken a holiday. He had felt suddenly weary of breakfasting every morning before his view over the Green Park; he had realised that the picking up of first editions at sales afforded insufficient exercise for a man of thirty-three; the very crimes of London were over-sophisticated. He had abandoned his flat and his friends and fled to the wilds of Corsica. For the last three months he had forsworn letters, newspapers, and telegrams. He had tramped about the mountains, admiring from a cautious

distance the wild beauty of Corsican peasant-women, and studying the vendetta in its natural haunt. In such conditions murder seemed not only reasonable, but lovable. Bunter, his confidential man and assistant sleuth, had nobly sacrificed his civilised habits, had let his master go dirty and even unshaven, and had turned his faithful camera from the recording of finger-prints to that of craggy scenery. It had been very refreshing.

Now, however, the call of the blood was upon Lord Peter. They had returned late last night in a vile train to Paris, and had picked up their luggage. The autumn light, filtering through the curtains, touched caressingly the silver-topped bottles on the dressing-table, outlined an electric lamp-shade and the shape of the telephone. A noise of running water near by proclaimed that Bunter had turned on the bath (h. & c.) and was laying out scented soap, bath-salts, the huge bath-sponge, for which there had been no scope in Corsica, and the delightful flesh-brush with the long handle, which rasped you so agreeably all down the spine. "Contrast," philosophised Lord Peter sleepily, "is life. Corsica—Paris—then London.... Good morning, Bunter."

"Good morning, my lord. Fine morning, my lord. Your lordship's bath-water is ready."

"Thanks," said Lord Peter. He blinked at the sunlight.

It was a glorious bath. He wondered, as he soaked in it, how he could have existed in Corsica. He wallowed happily and sang a few bars of a song.

Stanford

The standard sentences in the text were identified correctly. However, the heading and quotes at the beginning of the chapter were grouped together as one sentence.

<i>CHAPTER I "OF HIS MALICE AFORETHOUGHT" "O, Who hath done this deed?"</i>

Whether this is correct or incorrect is debatable, as they are not standard sentences. However, the speaker of the quote (Othello) is added to the first sentence of the text as *Othello Lord Peter Wimsey stretched himself luxuriously between the sheets provided by the Hôtel Meurice.*

Due to space restrictions, the complete part-of-speech tags will not be given here. However, the errors will be noted in the table.

Table 26 Morphological annotation 7 – Stanford

Stanford			
Word	Lemma	Part-of-speech	Comment
I	I	PRP	As this is the roman numeral I, the lemma should be 1. Incorrect part-of-speech, it should be a number (CD).
AFORETHOUGHT	aforethought	NN	Incorrect part-of-speech, it should be an adjective (JJ).
O	O	NN	Incorrect part-of-speech, it should be an interjection (UH).
hath	have	VBP	Incorrect part-of-speech, it should be a verb, 3rd person singular present (VBZ).
Lord	Lord	NNP	Incorrect part-of-speech, it should be a noun (NN). *All instances where Lord are spelled with a capital letter is marked as NNP or NN.
[the] very [crimes]	very	RB	Incorrect part-of-speech, it should be an adjective (JJ).
flat	flat	JJ	Incorrect part-of-speech, it should be a noun (NN).
haunt	haunt	VBP	Incorrect part-of-speech, it should be a noun (NN).
recording	recording	NN	Incorrect part-of-speech, it should be a gerund (VBG).
proclaimed	proclaimed	JJ	Incorrect part-of-speech, it should be a verb, past tense (VBD).
flesh-brush	flesh-brush	JJ	Incorrect part-of-speech, it should be a noun (NN).
handle	handle	VB	Incorrect part-of-speech, it should be a noun (NN).
contrast	contrast	NNP	Incorrect part-of-speech, it should be a noun (NN).

spaCy

The standard sentences in the text were mostly identified correctly. However, the use of quotation marks caused some problems and are sometimes seen as a sentence on its own. For example:

<p><sentence> “</p> <p><sentence> Thanks,” said Lord Peter.</p>

Instead of grouping the chapter heading and quotes together as a single sentence, spaCy splits it into different sentences.

```

<sentence> CHAPTER
<sentence> I
<sentence> “
<sentence> OF HIS MALICE AFORETHOUGHT”
<sentence> “
<sentence> O, Who hath done this deed?” Othello
<sentence> Lord Peter Wimsey...

```

Due to space restrictions, the complete part-of-speech tags will not be given here. However, the errors will be noted in the table.

Table 27 Morphological annotation 7 – spaCy

spaCy			
Word	Lemma	Part-of-speech	Comment
I	I	PRP	As this is the roman numeral I, the lemma should be 1. Incorrect part-of-speech, it should be a number (CD).
“	“	NN	*spaCy often marks quotation marks as NN. Only this instance is shown here.
MALICE	Malice	NNP	Incorrect part-of-speech, it should be a noun (NN).
AFORETHOUGHT	aforethought	NNP	Incorrect part-of-speech, it should be an adjective (JJ).
hath	hath	NN	The lemma should be “have”. Incorrect part-of-speech, it should be a verb, 3rd person singular present (VBZ).
Othello	Othello	NN	Incorrect part-of-speech, it should be a proper noun (NNP).
Lord	Lord	NNP	Incorrect part-of-speech, it should be a noun (NN). *All instances where Lord are spelled with a capital letter is marked as NNP or NN.
picking	picking	NN	Incorrect part-of-speech, it should be a gerund (VBG).
thirty	thirty	CD	*spaCy splits words joined by a hyphen to different tokens. Only this instance is shown here.
-	-	HYPH	
three	three	CD	
flat	flat	JJ	Incorrect part-of-speech, it should be a noun (NN).

forsworn	forsworn	JJ	Incorrect part-of-speech, it should be a verb, past participle (VBN).
let	let	VBN	Incorrect part-of-speech, it should be a verb, base form (VB).
recording	recording	NN	Incorrect part-of-speech, it should be a gerund (VBG).
near	near	RB	Incorrect part-of-speech, it should be a preposition (IN).
Contrast	Contrast	NNP	Incorrect part-of-speech, it should be a noun (NN).
—	—	.	Could be marked as colon or ellipsis, not sentence terminator.
—	—	NNP	Incorrect part-of-speech, it should be punctuation.
....	Could separate the ellipsis and the sentence terminator.

Discussion

Automated encoding on the morphological level is advanced. Most of the tokenisation (word segmentation) is done correctly. It is clear that there are some design decisions that differ between tools. Stanford CoreNLP keeps hyphenated words together, whereas spaCy splits hyphenated words into separate words. Stanford seems to handle punctuation better than spaCy, for example the separation of ellipsis and period. Contractions are typically handled effectively in both tools. In terms of sentence segmentation, standard sentences are extracted correctly, but as expected, sections that are not standard sentences, such as headings or quotes, are problematic. Here it is also clear that different tools make different design decisions. Stanford CoreNLP tends to group items together and spaCy tends to split items.

Both libraries identify the lemma fairly accurately, except for the handling of pronouns in spaCy. All the lemmas of pronouns are indicated as -PRON-. Understandably, a few problems are encountered where the author uses words and punctuation more artistically, for example to indicate dialect. Here spaCy performed better in identifying the correct lemma.

Part-of-speech tagging is done well by both libraries. Simple sentences seem to be tagged without problems. However, as sentences become complex, words that have multiple senses occur or the author of the text purposefully deviates from standard spelling to make a point, some errors can occur. Stanford CoreNLP incorrectly tagged the word *sovereign* as an adjective, whereas in this context it refers to the former British gold coin and both libraries tagged the word *flat* as an adjective, whereas it is used as a noun meaning *apartment* in this example. This meaning is mostly used in British English. It could be that the training data used to train the tools have more

current American texts than older British texts. In the example where multiple senses were used, Stanford CoreNLP assigned the same part-of-speech tag to each word. spaCy performed better and only misidentified the last instance of *well*. The difficulty the libraries have with words that are spelled differently to indicate a different pronunciation is also not surprising and entirely reasonable. The words are not standard words anymore, but are used by the author with artistic freedom. Stanford CoreNLP did not identify the interjections correctly. Both tools seemed to categorise a word as a proper noun when it is capitalised, for example *Lord*. The other errors will not be discussed in detail.

7.2.2. Syntactic encoding

The same tools that were used for the morphological level will be used on the syntactic level, namely, Stanford CoreNLP and spaCy. For syntactic encoding Stanford CoreNLP provides output in Universal Dependencies (v1) (Stanford Parser, n.d.) and Stanford Dependencies. The spaCy parser uses labels from ClearNLP (spaCy – Annotation Specifications, n.d.). The Universal Dependencies (v1) are used in this study and listed in Table 3.

Stanford CoreNLP and spaCy were used to encode various examples on a syntactic level. Six sentences were parsed. The sentences are labeled S1 to S6 for easy referencing.

For each sentence, one table was created that includes the encoding from the two different parsers, enabling easy comparison. No errors are highlighted, nor are comments provided in the tables as such. Various issues will be discussed afterwards. The examples are not exhaustive but are sufficient to make certain observations about automated encoding of syntactic dependencies.

Encoded examples

[S1] *His flat had a beautiful view.*

Researcher's own sentence.

Table 28 Syntactic encoding 1

Word	Lemma	Stanford			spaCy		
		Part-of-speech	Governor	Dependency type	Part-of-speech	Governor	Dependency type
His	he	PRP\$	flat	nmod:poss	PRP\$	flat	poss
flat	flat	JJ	had	nsubj	NN	had	nsubj
had	have	VBD	ROOT	root	VBD	had	ROOT
a	a	DT	view	det	DT	view	det
beautiful	beautiful	JJ	view	amod	JJ	view	amod
view	view	NN	had	dobj	NN	had	dobj
.	.	.	had	punct	.	had	punct

[S2] *The man put the book on the table.*

Researcher's own sentence.

Table 29 Syntactic encoding 2

Word	Lemma	Stanford			spaCy		
		Part-of-speech	Governor	Dependency type	Part-of-speech	Governor	Dependency type
The	the	DT	man	det	DT	man	det

man	man	NN	put	nsubj	NN	put	nsubj
put	put	VBD	ROOT	root	VBD	put	ROOT
the	the	DT	book	det	DT	book	det
book	book	NN	put	dobj	NN	put	dobj
on	on	IN	table	case	IN	put	prep
the	the	DT	table	det	DT	table	det
table	table	NN	put	nmod	NN	on	pobj
.	.	.	put	punct	.	put	punct

[S3] “Does changing the salt and sugar never pall on you as a jest?”

(From *The Innocence of Father Brown* by G.K. Chesterton.)

Table 30 Syntactic encoding 3

		Stanford			spaCy		
Word	Lemma	Part-of-speech	Governor	Dependency type	Part-of-speech	Governor	Dependency type
``	``	``	Does	punct	NN	“	ROOT
Does	do	VBZ	ROOT	root	VBZ	changing	aux
changing	change	VBG	Does	xcomp	VBG	pall	advcl
the	the	DT	salt	det	DT	salt	det
salt	salt	NN	changing	dobj	NN	changing	dobj
and	and	CC	salt	cc	CC	salt	cc
sugar	sugar	NN	salt	conj	NN	salt	conj
never	never	RB	changing	neg	RB	pall	neg
pall	pall	NN	changing	nmod	VB	pall	ROOT

on	on	IN	pall	case	IN	pall	prep
you	you	PRP	pall	dep	PRP	on	pobj
as	as	IN	jest	case	IN	pall	prep
a	a	DT	jest	det	DT	jest	det
jest	jest	NN	changing	nmod	NN	as	pobj
?	?	.	Does	punct	.	pall	punct
"	"	"	Does	punct	"	pall	punct

[S4] “Contrast,” philosophised Lord Peter sleepily, “is life.”

(From *Clouds of Witness* by Dorothy L. Sayers.)

Table 31 Syntactic encoding 4

Word	Lemma	Stanford			spaCy		
		Part-of-speech	Governor	Dependency type	Part-of-speech	Governor	Dependency type
``	``	``	life	punct	NNP	Contrast	compound
Contrast	Contrast	NNP	life	nsubj	NNP	philosophised	npadvmod
,	,	,	philosophised	punct	,	philosophised	punct
"	"	"	philosophised	punct	"	philosophised	punct
philosophised	philosophise	VBD	life	parataxis	VBD	philosophised	ROOT
Lord	Lord	NNP	Peter	compound	NNP	Peter	compound
Peter	Peter	NNP	philosophised	nsubj	NNP	philosophised	dobj
sleepily	sleepily	RB	Peter	advmod	RB	philosophised	advmod
,	,	,	philosophised	punct	,	philosophised	punct
``	``	``	philosophised	punct	NNP	is	punct
is	be	VBZ	life	cop	VBZ	philosophised	ccomp

life	life	NN	ROOT	root	NN	is	attr
.	.	.	life	punct	.	philosophised	punct
"	"	"	life	punct	"	"	ROOT

[S5] *A noise of running water near by proclaimed that Bunter had turned on the bath (h. & c.) and was laying out scented soap, bath-salts, the huge bath-sponge, for which there had been no scope in Corsica, and the delightful flesh-brush with the long handle, which rasped you so agreeably all down the spine.*

(From *Clouds of Witness* by Dorothy L. Sayers.)

Table 32 Syntactic encoding 5

Word	Lemma	Stanford			Word	Lemma	spaCy		
		Part-of-speech	Governor	Dependency type			Part-of-speech	Governor	Dependency type
A	a	DT	noise	det	A	a	DT	noise	det
noise	noise	NN	handle	nsubj	noise	noise	NN	proclaimed	nsubj
of	of	IN	running	mark	of	of	IN	noise	prep
running	run	VBG	noise	acl	running	run	VBG	water	amod
water	water	NN	running	dobj	water	water	NN	of	pobj
near	near	IN	proclaimed	case	near	near	RB	noise	prep
by	by	IN	proclaimed	case	by	by	IN	noise	prep
proclaimed	proclaimed	JJ	running	nmod	proclaimed	proclaim	VBN	proclaimed	ROOT
that	that	IN	turned	mark	that	that	IN	turned	mark
Bunter	Bunter	NNP	turned	nsubj	Bunter	Bunter	NNP	turned	nsubj
had	have	VBD	turned	aux	had	have	VBD	turned	aux
turned	turn	VBN	proclaimed	dep	turned	turn	VBN	proclaimed	ccomp

on	on	RP	turned	compound:prt	on	on	RP	turned	prt
the	the	DT	bath	det	the	the	DT	bath	det
bath	bath	NN	turned	dobj	bath	bath	NN	turned	dobj
-LRB-	-lrb-	-LRB-	h.	punct	((-LRB-	bath	punct
h.	h.	NN	bath	dep	h.	h.	NN	bath	appos
&	&	CC	h.	cc	&	&	CC	h.	cc
c.	c.	NN	h.	conj	c.	c.	NN	h.	conj
-RRB-	-rrb-	-RRB-	h.	punct))	-RRB-	bath	punct
and	and	CC	turned	cc	and	and	CC	turned	cc
was	be	VBD	laying	aux	was	be	VBD	laying	aux
laying	lay	VBG	turned	conj	laying	lay	VBG	turned	conj
out	out	RP	laying	compound:prt	out	out	RP	laying	prt
scented	scented	JJ	soap	amod	scented	scented	JJ	soap	amod
soap	soap	NN	laying	dobj	soap	soap	NN	laying	dobj
,	,	,	soap	punct	,	,	,	soap	punct
bath-salts	bath-salt	NNS	soap	appos	bath	bath	NN	salts	compound
					-	-	HYPH	salts	punct
					salts	salt	NNS	soap	conj
,	,	,	soap	punct	,	,	,	salts	punct
the	the	DT	bath-sponge	det	the	the	DT	sponge	det
huge	huge	JJ	bath-sponge	amod	huge	huge	JJ	sponge	amod
bath-sponge	bath-sponge	NN	soap	appos	bath	bath	NN	sponge	compound
					-	-	HYPH	sponge	punct
					sponge	sponge	NN	salts	conj
,	,	,	soap	punct	,	,	,	sponge	punct
for	for	IN	which	case	for	for	IN	been	prep

which	which	WDT	scope	nmod	which	which	WDT	for	pobj
there	there	EX	scope	expl	there	there	EX	been	expl
had	have	VBD	scope	aux	had	have	VBD	been	aux
been	be	VBN	scope	cop	been	be	VBN	sponge	relcl
no	no	DT	scope	neg	no	no	DT	scope	det
scope	scope	NN	soap	acl:relcl	scope	scope	NN	been	attr
in	in	IN	Corsica	case	in	in	IN	scope	prep
Corsica	Corsica	NNP	scope	nmod	Corsica	Corsica	NNP	in	pobj
,	,	,	scope	punct	,	,	,	sponge	punct
and	and	CC	scope	cc	and	and	CC	sponge	cc
the	the	DT	flesh-brush	det	the	the	DT	brush	det
delightful	delightful	JJ	flesh-brush	amod	delightful	delightful	JJ	brush	amod
flesh-brush	flesh-brush	JJ	scope	conj	flesh	flesh	NN	brush	compound
					-	-	HYPH	brush	punct
					brush	brush	NN	sponge	conj
with	with	IN	long	case	with	with	IN	brush	prep
the	the	DT	long	det	the	the	DT	handle	det
long	long	JJ	flesh-brush	nmod	long	long	JJ	handle	amod
handle	handle	VB	ROOT	root	handle	handle	NN	with	pobj
,	,	,	handle	punct	,	,	,	handle	punct
which	which	WDT	rasped	nsubj	which	which	WDT	rasped	nsubj
rasped	rasp	VBD	handle	ccomp	rasped	rasp	VBD	handle	relcl
you	you	PRP	rasped	dobj	you	-PRON-	PRP	rasped	dobj
so	so	RB	rasped	advmod	so	so	RB	agreeably	advmod
agreeably	agreeably	RB	rasped	advmod	agreeably	agreeably	RB	all	advmod
all	all	DT	spine	dep	all	all	DT	down	advmod
down	down	IN	spine	case	down	down	IN	rasped	prep

the	the	DT	spine	det	the	the	DT	spine	det
spine	spine	NN	rasped	nmod	spine	spine	NN	down	pobj
.	.	.	handle	punct	.	.	.	proclaimed	punct

[S6] *When they were quite ready, the now triumphant Toad led his companions to the paddock and set them to capture the old grey horse, who, without having been consulted, and to his own extreme annoyance, had been told off by Toad for the dustiest job in this dusty expedition.*

From *The Wind In The Willows* by Kenneth Grahame.

Table 33 Syntactic encoding 6

Word	Lemma	Stanford			spaCy		
		Part-of-speech	Governor	Dependency type	Part-of-speech	Governor	Dependency type
When	when	WRB	ready	advmod	WRB	were	advmod
they	they	PRP	ready	nsubj	PRP	were	nsubj
were	be	VBD	ready	cop	VBD	led	advcl
quite	quite	RB	ready	advmod	RB	ready	advmod
ready	ready	JJ	led	advcl	JJ	were	acomp
,	,	,	led	punct	,	led	punct
the	the	DT	Toad	det	DT	Toad	det
now	now	RB	triumphant	advmod	RB	triumphant	advmod
triumphant	triumphant	JJ	Toad	amod	JJ	Toad	amod
Toad	Toad	NNP	led	nsubj	NNP	led	nsubj
led	lead	VBD	ROOT	root	VBD	led	ROOT
his	he	PRP\$	companions	nmod:poss	PRP\$	companions	poss
companions	companion	NNS	led	dobj	NNS	led	dobj

to	to	TO	paddock	case	IN	led	prep
the	the	DT	paddock	det	DT	paddock	det
paddock	paddock	NN	led	nmod	NN	to	pobj
and	and	CC	led	cc	CC	led	cc
set	set	VBD	led	conj	VBD	led	conj
them	they	PRP	set	dobj	PRP	set	dobj
to	to	TO	capture	mark	TO	capture	aux
capture	capture	VB	set	xcomp	VB	set	xcomp
the	the	DT	horse	det	DT	horse	det
old	old	JJ	grey	amod	JJ	horse	amod
grey	grey	JJ	horse	amod	JJ	horse	compound
horse	horse	NN	capture	dobj	NN	capture	dobj
,	,	,	horse	punct	,	horse	punct
who	who	WP	told	nsubjpass	WP	told	nsubjpass
,	,	,	told	punct	,	told	punct
without	without	IN	consulted	mark	IN	told	prep
having	have	VBG	consulted	aux	VBG	consulted	aux
been	be	VBN	consulted	auxpass	VBN	consulted	auxpass
consulted	consult	VBN	told	advcl	VBN	without	pcomp
,	,	,	consulted	punct	,	without	punct
and	and	CC	consulted	cc	CC	without	cc
to	to	TO	annoyance	case	IN	without	conj
his	he	PRP\$	annoyance	nmod:poss	PRP\$	annoyance	poss
own	own	JJ	annoyance	amod	JJ	annoyance	amod
extreme	extreme	JJ	annoyance	amod	JJ	annoyance	amod
annoyance	annoyance	NN	consulted	conj	NN	to	pobj
,	,	,	told	punct	,	told	punct

had	have	VBD	told	aux	VBD	told	aux
been	be	VCN	told	auxpass	VCN	told	auxpass
told	tell	VCN	horse	acl:relcl	VCN	horse	relcl
off	off	RP	told	compound:prt	RP	told	prt
by	by	IN	Toad	case	IN	told	agent
Toad	Toad	NNP	told	nmod	NNP	by	pobj
for	for	IN	job	case	IN	told	prep
the	the	DT	job	det	DT	job	det
dustiest	dustiest	JJS	job	amod	JJS	job	compound
job	job	NN	told	nmod	NN	for	pobj
in	in	IN	expedition	case	IN	job	prep
this	this	DT	expedition	det	DT	expedition	det
dusty	dusty	JJ	expedition	amod	JJ	expedition	amod
expedition	expedition	NN	job	nmod	NN	in	pobj
.	.	.	led	punct	.	led	punct

Discussion

The purpose of this evaluation is to provide illustrative examples of automated syntactic encoding. It is beyond the scope of this study to do a comprehensive evaluation and comparison of the two parsers. However, the examples show what can be expected when using software to annotate texts on a syntactic level. Based on the observations by the researcher, various issues or points of discussion were identified. These issues were grouped together in categories and will be discussed accordingly.

1) Many of the syntactic dependencies and dependency types are evidently correct, and in most cases both parsers agree, for example:

- *Beautiful* as adjectival modifier (amod) that is governed by *view* in S1.
- *View* as direct object (dobj) that is governed by *had* in S1.
- *Had* as root in S1.
- *Man* as nominal subject (nsubj) that is governed by *put* in S2.
- *Book* as direct object (dobj) that is governed by *put* in S2.
- *Put* as root in S2.
- *The* as determiner (det) that is governed by *salt* in S3.
- *Salt* as direct object (dobj) that is governed by *changing* in S3.
- *And* as coordinating conjunction (cc) that is governed by *salt* in S3.
- *Sugar* as conjunct (conj) that is governed by *salt* in S3.
- *A* as determiner (det) that is governed by *noise* in S5.
- *Bunter* as nominal subject (nsubj) that is governed by *turned* in S5.
- *Had* as auxiliary (aux) that is governed by *turned* in S5.
- *Was* as auxiliary (aux) that is governed by *laying* in S5.
- *Scented* as adjectival modifier (amod) that is governed by *soap* in S5.
- *Soap* as direct object (dobj) that is governed by *laying* in S5.

- *Huge* as adjectival modifier (amod) that is governed by *bath-sponge (sponge)* in S5.
- *Quite* as adverbial modifier (advmod) that is governed by *ready* in S6.
- *Now* as adverbial modifier (advmod) that is governed by *triumphant* in S6.
- *Toad* (first occurrence) as nominal subject (nsubj) that is governed by *led* in S6.
- *Horse* as direct object (dobj) that is governed by *capture* in S6.

2) In some cases, the dependency type is correctly identified, but the link to the governor is incorrect, for example:

- *Never* in S3 is correctly labelled as negation modifier (neg) by both parsers, but Stanford incorrectly identifies *changing* as the governor.
- *Sleepily* in S4 is correctly labelled as adverbial modifier (advmod) by both parsers, but Stanford incorrectly identifies *Peter* as the governor. spaCy correctly identifies *philosophised* as the governor.
- *Noise* in S5 is correctly identified as nominal subject (nsubj) by both parsers, but Stanford incorrectly identifies *handle* as the governor. spaCy correctly identifies the governor as *proclaimed*.
- *Agreeably* in S5 is correctly identified as adverbial modifier (advmod) by both parsers, but spaCy incorrectly identifies *all* as the governor. Stanford correctly identifies *rasped* as the governor.

3) In some cases, the correct governor is identified, but the dependency type is incorrect, for example:

- In S4 *Peter* is incorrectly identified as direct object (dobj) in spaCy, but is correctly linked to *philosophised* as governor.

4) Some syntactic dependencies and dependency types are evidently incorrect, for example:

- *Philosophised* in S4 is incorrectly identified as parataxis with *life* as governor by Stanford. It is correctly identified as the root in spaCy.
- *Water* in S5 is incorrectly labelled as direct object (dobj) and incorrectly linked to *running* as governor by Stanford.

5) Incorrect morphological parsing (POS tagging) can lead to incorrect syntactic parsing, for example:

- *Pall* in S3 is tagged as a noun (NN) in Stanford, but it is a verb; this results in an incorrect syntactic parsing where *pall* is regarded as a nominal modifier (nmod) in Stanford. The parsing in spaCy is correct.
- *Handle* in S5 is tagged as a verb (VB) in Stanford, but it is a noun; this results in an incorrect syntactic parsing where *handle* is regarded as root in Stanford. The parsing in spaCy is correct.
- *Proclaimed* in S5 is tagged as an adjective (JJ) in Stanford, but it is a verb in past tense; this results in an incorrect syntactic parsing where *proclaimed* is regarded as a nominal modifier (nmod) in Stanford. The parsing in spaCy is correct.

6) Incorrect POS tagging can nevertheless sometimes result in correct syntactic parsing, for example:

- *Flat* in S1 is tagged as adjective (JJ) by Stanford but is a noun. However, the syntactic parsing is correct.
- *Proclaimed* in S5 is tagged as a past participle (VBN) in spaCy, but it is a verb in past tense (VBD). The POS tagging was, however, only marginally incorrect (past participle of the verb, instead of a past tense of the verb). Since both POS tags refer to verb, this could explain the correct syntactic parsing.

7) In some cases, there is only a slight difference in interpretation or use of different labels, but both analyses could be correct, for example:

- *On* in S2 is labelled as case by Stanford and prep by spaCy. Case is the label used by Stanford for case-marking elements including prepositions. Prep is the label used by spaCy for prepositions.
- *Running* in S5 is marked as acl (clausal modifier of noun (adjectival clause)) in Stanford and amod (adjectival modifier) in spaCy.
- *So* in S5 is correctly identified as adverbial modifier (advmod) by both parsers, but spaCy identifies *agreeably* as the governor, while Stanford identifies *rasped* as the governor.

- *His* in S6 is identified as a possession modifier, Stanford using the label `nmod:poss` and spaCy using the label `poss`.
- In S6 *grey* is identified as compound by spaCy, while Stanford on the other hand sees it as an adjectival modifier (`amod`), both parsers identifying *horse* as the governor.
- Stanford fairly consistently labels a noun after a determiner towards the end of a sentence as a nominal modifier (`nmod`) and spaCy as an object of preposition (`pobj`).

8) Inverted commas and direct speech can be problematic, for example:

- spaCy fairly consistently parsed initial inverted commas (indicating direct speech) incorrectly as `NN` and `root`.
- Direct speech consisting of two separate sections, as in S4, seems to have created problems for both Stanford and spaCy.

It should be clear from the issues that were raised in this discussion that the evaluation of syntactic parsers is a complex activity. Furthermore, these examples are not exhaustive and only highlight categories of issues. However, three important points relevant to this study should be evident. Firstly, both parsers frequently encode the syntactic dependencies correctly. Secondly, both parsers still make errors. This shows that syntactic parsing is fairly well developed, but still does not result in completely accurately tagged texts. Thirdly, it is necessary to understand the theoretical framework as well as tagset that a parser adopts. Some of the differences in interpretation from the two parsers examined in this study highlights this issue.

It should also be noted that errors at the beginning of a sentence do not mean the rest of the sentence will be entirely incorrectly parsed. It is also clear that simple sentences are annotated more accurately, but understandably, there are more issues in longer or complex sentences. Punctuation could also cause some issues.

7.2.3. Semantic encoding

For the semantic level the correct WordNet sense should be identified and PyWSD will be considered. PyWSD is implementations of Word Sense Disambiguation (WSD) technologies in the Python language (Tan, 2014). It includes Lesk algorithms, Maximizing Similarity and Baselines. The sense assigned to a word is from the WordNet database. This package was selected for this study, as it is publicly available

and can be used with minimal configuration and programming and uses senses from WordNet, as is required for this study. In this study, the classic Lesk algorithm, as available through the PyWSD package, was used.

Encoded examples

Three sentences were selected. The encoding for each example is given as in a table.

Though some senses are identified correctly, most senses of the assigned senses are incorrect. It seems that there are too many errors to rely on this package for automated semantic encoding.

Sentence: For the last three months he had forsworn letters, newspapers, and telegrams.

Source: *Clouds of Witness* by Dorothy L. Sayers

Table 34 Semantic encoding 1

Word	Automated encoding		Comment
	Synset	Definition	
For	N/a	N/a	N/a
the	N/a	N/a	N/a
last	last.v.01	persist for a specified period of time	Incorrect last.s.01 (immediately past)
three	trey.n.02	one of four playing cards in a deck having three pips	Incorrect three.n.01 (the cardinal number that is the sum of one and one and one)
months	calendar_month.n.01	one of the twelve divisions of the calendar year	Correct
he	N/a	N/a	N/a
had	suffer.v.02	undergo (as of injuries and illnesses)	Incorrect have.v.01 (have or possess, either in a concrete or an abstract sense) – closest meaning in WordNet
forsworn	abjure.v.01	formally reject or disavow a formerly held belief, usually under pressure	Correct
letters	letter.n.05	an award earned by participation in a school sport	Incorrect letter.n.01 (a written message addressed to a person or organization)
newspapers	newspaper.n.04	cheap paper made from wood pulp and used for printing newspapers	Incorrect newspaper.n.01 (a daily or weekly publication on folded sheets; contains news and articles and advertisements)
and	N/a	N/a	N/a
telegrams	telegram.n.01	a message transmitted by telegraph	Correct

Sentence: A person may be proud without being vain.

Source: *Pride and Prejudice* by Jane Austen

Table 35 Semantic encoding 2

Word	Automated encoding		Comment
	Synset	Definition	
A	N/a	N/a	N/a
person	person.n.03	a grammatical category used in the classification of pronouns, possessive determiners, and verb forms according to whether they indicate the speaker, the addressee, or a third party	Incorrect person.n.01 (a human being)
may	N/a	N/a	N/a
be	be.v.10	spend or use time	Incorrect be.v.01 (have the quality of being; (copula, used with an adjective or a predicate noun))
proud	proud.a.01	feeling self-respect or pleasure in something by which you measure your self-worth; or being a reason for pride	Correct
without	N/a	N/a	N/a
being	be.v.10	spend or use time	Incorrect be.v.01 (have the quality of being; (copula, used with an adjective or a predicate noun))
vain	conceited.s.01	characteristic of false pride; having an exaggerated sense of self-importance	Correct

Sentence: And he threw down half a sovereign and caught up his coat.

Source: *The Innocence of Father Brown* by G.K. Chesterton

Table 36 Semantic encoding 3

Word	Automated encoding		Comment
	Synset	Definition	
And	N/a	N/a	N/a
he	N/a	N/a	N/a
threw	throw.v.14	throw (a die) out onto a flat surface	Incorrect throw.n.02 (move violently, energetically, or carelessly)
down	toss_off.v.02	drink down entirely	Incorrect down.r.01 (spatially or metaphorically from a higher to a lower level or position)
half	one-half.n.01	one of two equal parts of a divisible whole	Correct
a	N/a	N/a	N/a
sovereign	sovereign.s.02	greatest in status or authority or power	Incorrect WordNet does not contain a correct sense
and	N/a	N/a	N/a
caught	watch.v.03	see or watch	Incorrect catch.v.04 (take hold of so as to seize or restrain or stop the motion of)
up	up.v.01	raise	Incorrect up.r.01 (spatially or metaphorically from a lower to a higher position)
his	N/a	N/a	N/a
coat	coating.n.01	a thin layer covering something	Incorrect coat.n.01 (an outer garment that has sleeves and covers the body from shoulder down; worn outdoors)

Discussion

The semantic encoding using the classic Lesk algorithm, as available through the PyWSD package, was not sufficient to do automated encoding. There are too many errors and it cannot be relied upon. It will be better to start with manual encoding than having to work through so many erroneous encodings.

7.3. Conclusion

At the beginning of this chapter, it was stated that the manual encoding of texts is time-consuming and expensive, and that automated encoding has to be considered in order for encoding on a large scale to be feasible. The purpose of this chapter was to investigate to what extent the encoding of texts can be automated according to the metadata suggested in this study.

On the morphological and syntactic level there has been such progress that there are publicly available tools that are relatively easy to use, are stable and are claimed to provide fairly accurate results. As a result, some of these tools could be tested in this chapter. Though not the same confidence can currently be placed in semantic encoding, there are tools available and one of these was tested.

From the examples that were encoded, it seems that both Stanford CoreNLP and spaCy perform well and make few errors in annotating data on the morphological level. However, some errors do occur, especially in sections where the text is not considered standard. If a very large text collection has to be encoded and a few errors are acceptable, then the currently available tools are sufficient to automate the annotation of texts on a morphological level. This seems to be the approach by the Google Books Ngram Viewer in which a large text collection was processed and encoded automatically on a morphological and syntactic level. On such a large scale, it is not possible to correct the encoding manually, even if there must be some errors. In such cases the user must understand that there are some errors in the data collection. However, it seems that there are calls from the field of Digital Humanities to have smaller collections that are of high quality. Therefore, if the purpose is to have a collection where every word is correctly encoded then natural language processing tools are insufficient. However, the error rate is so small that the tools can be used to do the first round of encoding and it can afterwards be corrected by hand. This would be especially important, for example, for sections where there are non-standard sentences, where foreign words are used, or spelling is changed to suit the purpose of the author. It could be useful to compare the results of the two libraries and pay attention to those tokens that are tagged differently. It could also be useful to see if

different training data will produce different results, specifically if the training data are predominantly American texts and the data to annotate will be British texts. A semi-automated approach can improve the quality of the encoding of a collection.

Much of the discussion on morphological encoding seems to apply to the syntactic level as well. Much of the parsing seems to be correct, but errors do occur. It seems that at this stage it will not be possible to rely on software for completely accurate encoding on a syntactic level, but that it could provide a starting point for manual correction and so result in a semi-automated process. Alternatively, if it is sufficient to have an encoded collection that has some errors, automated encoding on a syntactic level can be considered. Furthermore, it is evident that it is important to understand the theoretical principles, design decisions and tagsets (labels) that are used by different parsers. In the same way that hyphenated words are treated differently on a morphological level, different design decisions were made by the different parsers on a syntactical level.

In terms of semantic encoding, it seems that it is currently not feasible to do automated encoding. Reliable semantic encoding must be done manually, but manual semantic encoding is tedious and time-consuming. If no automated tools for this process will be developed, it will most probably not be feasible to encode large collections with semantic information.

There is promising work being done for most of the elements that need to be encoded on the functional level. For example, languages can be detected to a certain extent, work is being done on recognising names in a text (named entity recognition) and on identifying sections in a text (document layout), and more. However, this work is still done on separate tools or is even only done in a research capacity. It will currently not be possible to submit a text to a tool and have it encoded according to the specifications of this research. If more research in this area is being done, it would mean that in future different algorithms or programs can be combined to help with the encoding of texts. Even if a text cannot be encoded completely, it could possibly reach a level of semi-automated encoding, where the tool does the first round of encoding and is manually corrected by hand. Such an integrated tool will require extensive programming.

There is also currently no automated method to extract bibliographic information from a text. However, future work could investigate using existing bibliographic records for texts, for example, MARC records or other records from libraries. The challenge would be to extract the data from the correct fields from the old records and populate the

fields as suggested in this study. This should not be as difficult as trying to determine the bibliographic metadata of a text in an automated manner by using the text only. The main difficulty would be if existing bibliographic records are in different formats and the program has to be changed each time a new format is encountered.

In terms of in-text bibliographic metadata, one of the main arguments of this study is that there can be quotations in a text that have different bibliographic metadata to that of the text in which it is used. Unfortunately, it is not yet possible to automate the encoding of these metadata. It could be possible to explore the use of tools used to detect plagiarism to identify words that have been used in other texts.

Unfortunately, there is no complete and definitive answer to the automated encoding of texts as of yet. However, there are exciting developments in the field of natural language processing and some aspects can already be encoded with some accuracy automatically and there is reason to be optimistic about the automated encoding of many of the elements discussed in this study.

It should be noted that the development of tools in one area has an influence on the performance of tools in another area. In some instances, encoding on one level relies on information from a previous level. For example, tokenisation and POS tagging, which are done at the morphological level, are necessary to do syntactic analysis. The sequential nature of the encoding process is important to recognise. This means that if a part-of-speech tag is incorrectly assigned on the morphological level, the encoding on the syntactic level could be incorrect. The performance of tools used in each level is therefore important, as it could influence the accuracy of encoding in other levels.

It has been shown in this study that automated encoding is not completely accurate, and errors do occur. A decision has to be made about the desired quality of a collection. Sections where the encoding was automated will include errors. The error rate might not be high, but it will not be possible to guarantee perfectly encoded texts. In order to obtain highly accurate encodings, a large amount of manual labour is required. This is not feasible for large collections. Consequently, there is a trade-off between large collections that will inevitably include errors in the encoding and small collections of high quality. If it is necessary to work with large collections, it might be that there are enough data that the errors in encoding do not influence the results significantly. The user of the large collections should take this into consideration when analysing data and presenting results. If precise, accurate retrieval is required then the errors from automated encoding will present a problem.

A decision can also be made about the encoding that is necessary for specific collections and the tools available for those metadata, the accuracy of those tools, the time needed to correct the encoding and the time needed to encode manually. In this study, the most time-consuming part of the encoding was the semantic level, followed by the syntactic level. One could consider encoding only those levels where automated tools have an adequate performance and leave out levels where there are no or no adequate tools available. Where feasible, manual encoding can be added, for example, on the functional level. This will of course only be possible for small to medium-size collections. As was noted, some researchers from the field of digital humanities require collections where the metadata are reliable.

When using automated tools for encoding, it is important to consider the order in which the encoding takes place. This is necessary as one level of encoding could influence another level. Researchers from the field of natural language processing would be better equipped to suggest the ideal pipeline for automated encoding. However, the researcher will make some suggestions for the encoding of the metadata in this study. The researcher suggests that sentence segmentation and word tokenisation should be done first. After that, the morphological encoding should be done. The information from this level is necessary for the syntactic encoding. The semantic encoding could occur after the syntactic encoding. In this study it is suggested that this encoding is added to the first file (refer to chapter 4). In this study, the second file contains the bibliographic and functional encoding. If bibliographic metadata can be extracted in an automated manner, this can be done first, and the structure of the encoded document created. The elements from the functional level that can be detected in an automated manner can then be identified and encoded.

The next chapter will conclude this study with recommendations and suggestions for future work.

8. Conclusion

*Some people will never learn anything, for this reason,
because they understand everything too soon.*

Alexander Pope

8.1. Introduction

The phenomenal increase in information in the world cannot be overemphasised. Not only is information being created digitally at an extraordinary rate, paper-based texts are also being digitised. Various digitisation programs have been established and are contributing to growing digital text collections.

The growing importance of digital text collections has been highlighted in this study. It has been argued that digital text collections do not only offer improved access to material, but that new ways of working with texts can be explored. Improved retrieval and more efficient analysis can be done over larger collections of texts than can be done manually. Methods from other disciplines are being applied to text collections. Tools to analyse and explore digital text collections are available.

However, it has been argued that there is still a need to be able to search and retrieve items from text collections on a detailed level. It is apparent that metadata about a book or volume are not sufficient. More interesting questions can be asked if there is information available about sections and features within a book. This study proposed that if texts are encoded with more detailed metadata then retrieval in digital text collections could be improved.

In this chapter the research question and sub-questions that were presented in chapter 1 are answered. Various recommendations based on the findings of this study will be made, followed by suggestions for future work, before concluding.

8.2. Answering the research question and sub-questions

The main question of this study was the following: how can texts be encoded with detailed metadata to improve retrieval of words or phrases with specific properties from digital text collections?

To answer this question, the following sub-questions needed to be considered:

1) What metadata elements are available with which to describe and encode texts?

- 2) What are the characteristics of some of the tools that are currently used for the retrieval of words from digital text collections?
- 3) To what extent do current tools allow for the retrieval of words or phrases with specific properties from text collections?
- 4) What encoding can be suggested to allow for retrieval on a greater level of granularity and specificity?
- 5) To what extent can a tool (or prototype) be used to retrieve words or phrases with specific properties from texts encoded with detailed metadata?
- 6) What recommendations can be made to automate the encoding of texts?
- 7) How can the process followed in this study be formalised to make a contribution to theory development?
- 8) What recommendations can be made based on the results of this study?

8.2.1. Metadata elements to describe and encode texts

To answer the first sub-question, the researcher identified various types of information that would be useful to encode, and standards that could be used for the encoding of this information. This was done through a literature review and discussed in chapter 2.

Five categories of metadata were identified and discussed. The first three categories pertain to individual words. The morphological level considers the structure of words and the part-of-speech categories to which they belong. On the syntactic level the relationships between words are considered, whereas the semantic level is concerned with the meaning of words. The next two categories are concerned with the structures in the text (functional level) and the text as unit (bibliographic level).

For each category, the type of information that would be useful for retrieval was identified and different encoding standards were discussed.

8.2.2. Characteristics of tools that are currently used for the retrieval of words from digital text collections

In order to answer this sub-question, various tools that enable users to analyse the use of words and phrases in a text collection had to be identified and evaluated. This formed part of the literature review of chapter 2, though it included more than merely reading about the tools. The tools were examined and evaluated through a heuristic evaluation.

The following tools were considered: Google Books Ngram Viewer, HathiTrust+Bookworm, Perseus Project, Voyant Tools, TXM, BNCweb, BYU Corpora. Each tool was evaluated according to their interface design, the metadata in the collection(s) that it works with, the search options that it presents to users, filtering, the way in which results are presented and if users can link to more context, the complexity of use, the clarity of the help files and possible issues with the corpus.

8.2.3. The extent to which current tools allow for the retrieval of words or phrases with specific properties from text collections

The overarching goal of the heuristic evaluation of the search tools was to determine to what extent these tools allow a user to search for words or phrases with specific properties. The evaluation of the different tools revealed that different forms of searching on different levels of granularity with different levels of complexity are possible to a limited extent. It was determined that tools with simpler interfaces (e.g. Google Books Ngram Viewer) are coarse and do not allow for fine-grained filtering. Tools that allow for more detailed filtering (e.g. BNCweb), tend to require knowledge of a query language and/or the encoding of the data.

Thus, though different types of filtering and search options are available, none of the tools evaluated in this study could be used to search on a fine-grained level, without existing knowledge of the underlying data or complex query syntax. The concept of searching on detailed bibliographic properties is not supported.

8.2.4. Encoding that will allow for retrieval on a greater level of granularity and specificity

After reviewing the literature and identifying various types of information about properties of texts and the words in texts, and reviewing standards that could be used to encode this information to make it explicit, the researcher could answer the next sub-question. In chapter 4 the researcher suggested metadata that could be applied to text to enable more fine-grained retrieval. The five categories of metadata that were observed in chapter 2 were followed.

For each category, the researcher suggested what information would be useful to encode and in what way it could be encoded. The importance of standard bibliographic metadata was acknowledged, as it provides useful information about the composition of the data that the user is analysing. Furthermore, these metadata allow for basic filtering of texts. In addition to having information about texts as whole units, it was mentioned that texts could be complex entities. A text could have different sections and different sections could have different features (functional metadata). TEI was selected

as a standard used to encode features in a text. The bibliographic metadata are encoded in the header of the TEI document. Apart from only recognising different sections in a text, an important contention in this study is that different sections in a text can have properties that are distinct from the properties of the text as a whole, the concept of in-text metadata. If different sections in a text are identified and encoded, these sections can have specific bibliographic metadata assigned to them.

Apart from bibliographic metadata (and in-text bibliographic metadata) and functional metadata, this chapter also applied metadata on a word level. This included morphological, syntactic and semantic metadata that are used to make properties about individual words and the relationships between words explicit. On the morphological level the lemma and part-of-speech tag for each word is annotated. On the syntactic level the governor and dependency type of each word is annotated. On the semantic level the senses of words are annotated. The Penn Treebank tagset is used for the part-of-speech categories. The Universal Dependency annotations are used to annotate the dependencies. WordNet senses are used for the semantic level. All this information is encoded in XML.

Furthermore, the researcher showed how all five levels of encoding could be combined. The complexity of the five levels of metadata and the amount of information generated during the annotations meant, at a practical implementation level, that the morphological, syntactic and semantic data were encoded in one file, and the functional and bibliographic metadata in a second file. These two files were combined in a database to be able to use all the metadata in retrieval.

8.2.5. Prototype to retrieve words or phrases with specific properties from texts encoded with detailed metadata

In order to answer the next sub-question, two tasks were required. The first task was to encode a selection of texts with the metadata suggested in chapter 4. This is discussed in chapter 5. The next task was to develop a system to test whether words or phrases with specific properties could be retrieved. This system is discussed in chapter 6. The technical implementation of the system is discussed and how the two files are combined, and the information is stored in the database. The functionality of the system is also discussed. The tool allows searching of words or phrases and filtering on a detailed level, incorporating all metadata suggested in this study. The tool uses a graphical user interface but suggests how a query language could be employed.

8.2.6. Automated encoding of texts

Only a selection of texts was encoded to test the prototype and prove the main argument of this study. However, it is recognised that manually encoding texts is a time-intensive activity and if this system would be expanded, it would be necessary to consider the use of tools to automate the encoding. This sub-question was addressed in two ways. In chapter 2, part of the literature review about metadata looked at what is reported about the use of tools to do automated encoding. In chapter 7 some of the tools that are fairly established were evaluated. Text samples were selected and encoded and comments about the accuracy of the encoding were made. It was established that much progress has been made and some tools could be used, if not entirely, at least as part of a process where the system-generated encodings are checked manually.

8.2.7. A decision support system as contribution to theory development

The purpose of this study was to combine various levels of metadata so that retrieval of words or phrases on a detailed level is possible, both by laypersons and advanced users. As was highlighted, only a selection of metadata useful to prove the concept was used. However, the framework is extensible and the elements that are useful to specific users could be added. If the system were to be used by a group of researchers, their exact needs would need to be determined in order for the framework to be customised. The quality of the encoding that is necessary for the specific customisation is also important, as that will influence the encoding process.

As it is possible to customise the system, it is important to specify the process that needs to be followed to enhance texts with detailed metadata to enable retrieval. As a contribution to theory development, a decision support system was developed to enable users to follow the process needed to encode texts for detailed retrieval as proposed in this study.

A further benefit of the decision support system is that it aids in the replicability of the study. The steps necessary to encode files with detailed metadata are clearly indicated, each decision point is pointed out, as well as the different options one can take.

The decision support system developed to formalise the process of encoding is presented in Figure 180. As this diagram is large, sections will be selected and enlarged to aid readability (Figure 181, Figure 182 and Figure 183). Each section will then be discussed.

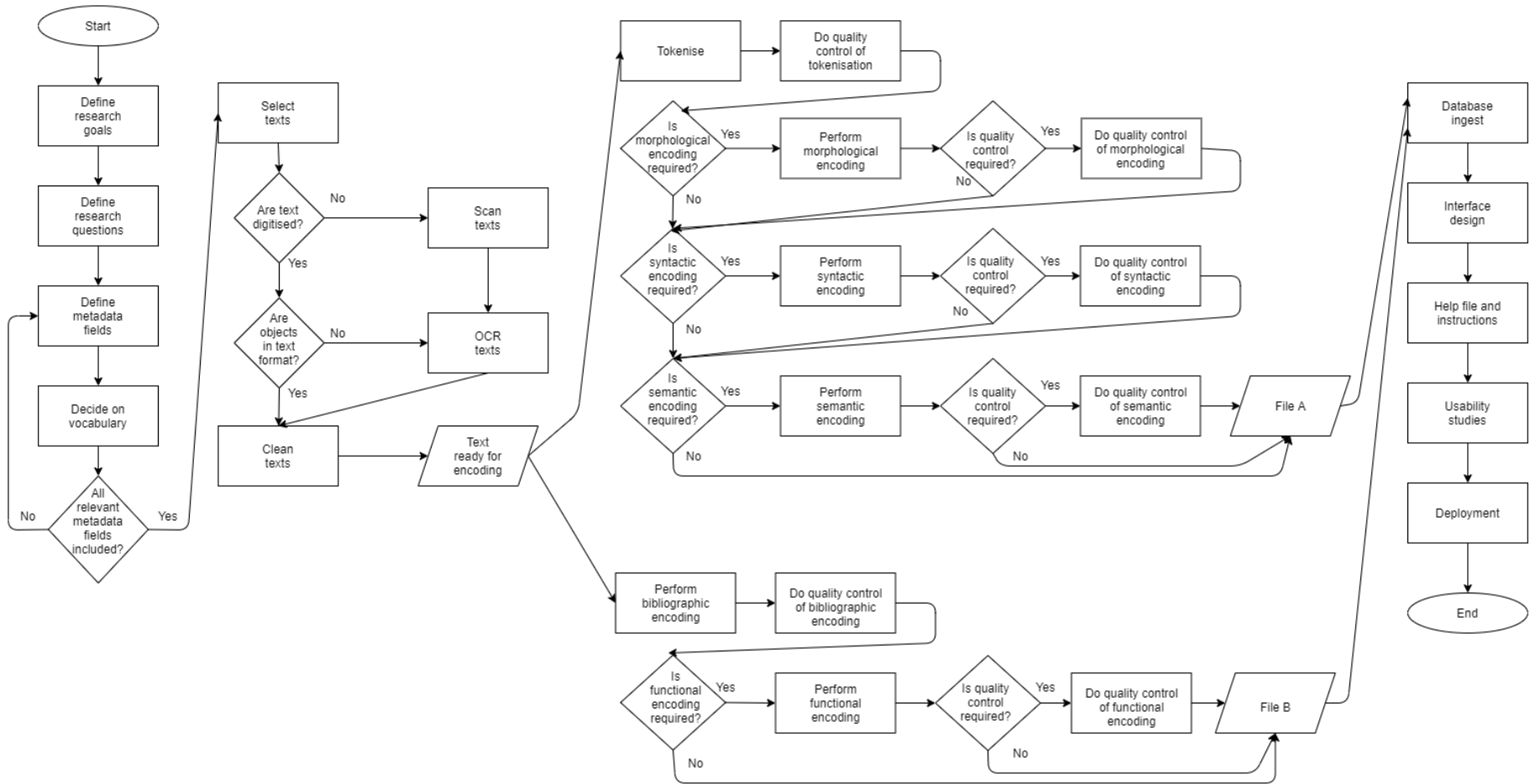


Figure 180 A decision support system to provide guidance when encoding texts with detailed metadata

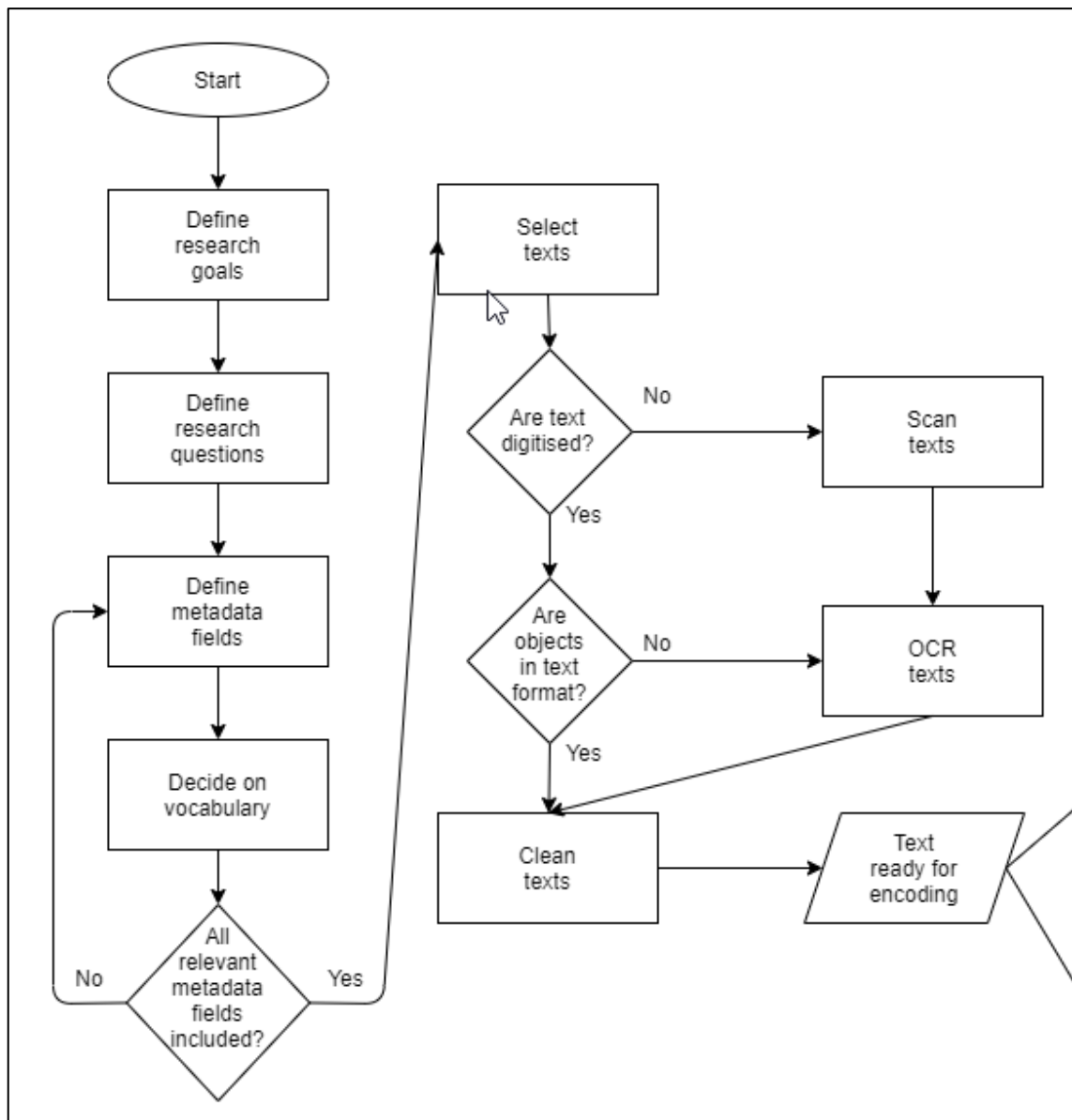


Figure 181 Enlargement: First part of the decision support system

The first step in the process is to define the research goals. Humanities as a discipline is a broad field and there are diverse aspects of texts that can be of interest to different researchers. The goal that a research group would like to achieve with a set of texts should be established. The goal will have a direct influence on the customisation of the system. During this step they should establish the types of entities in the texts that would be of interest. They should consider the number of texts that would be suitable for their research, as well as whether this should be a large or a smaller collection.

In the next step the types of queries that would be executed should be determined. What specific types of questions should this encoded set be able to answer? Are researchers interested in detailed or basic bibliographic detail? Are the researchers

interested in filtering according to detailed structures in the texts? According to what entities should the system filter?

Once the types of queries have been determined, the exact elements needed to answer those queries need to be defined. What information from each level is required? This is particularly important for the bibliographic and functional levels. The bibliographic level could include numerous fields and it would be necessary to narrow down what would be necessary for a particular customisation of the system. If it is necessary for a group of researchers, for example, to search according to the edition or an image caption then it is important to have such metadata, otherwise if this is not of interest then such extra metadata will not only clutter the encoding, but also add to the expense of encoding the data and of customising the prototype.

At this stage it is also important to decide on any controlled vocabulary that will be used. It could be that a specific group uses a particular standard for a certain element or field. In this study for example, the Library of Congress Subject Headings (LCSH) list was used for the subject field; however, this could be changed to suit the needs of the researchers.

The next step is to determine if all the metadata fields that are required to answer the questions identified have been defined. For example, if the research group would like to filter according to direct speech then there should be an element for direct speech in the schema, or if the research group would like to filter according to style of writing, there should be an element for style of writing. If there are still fields required, then the process moves back a step to define the metadata fields; otherwise, one can proceed to the next step.

After all the preparation has been completed, it is important to select the texts that will constitute the collection in the system. The composition of the collection is important and should address the needs of the researchers.

After selecting the texts, one should check if the texts are in digital format, if not they should be scanned. The digitised texts should not be in an image format, but in a format in which encoding can be done (e.g. text file). Optical character recognition could be used to get the content from digitised copies and create objects in text file.

The texts files should be prepared and cleaned. The text files should not contain any formatting. If the texts are obtained from a digital library that had inserted additional information into the text files, this should be removed. This could be additional

information about the library or copyright notices from the library. The result of this step would be text files that can be encoded.

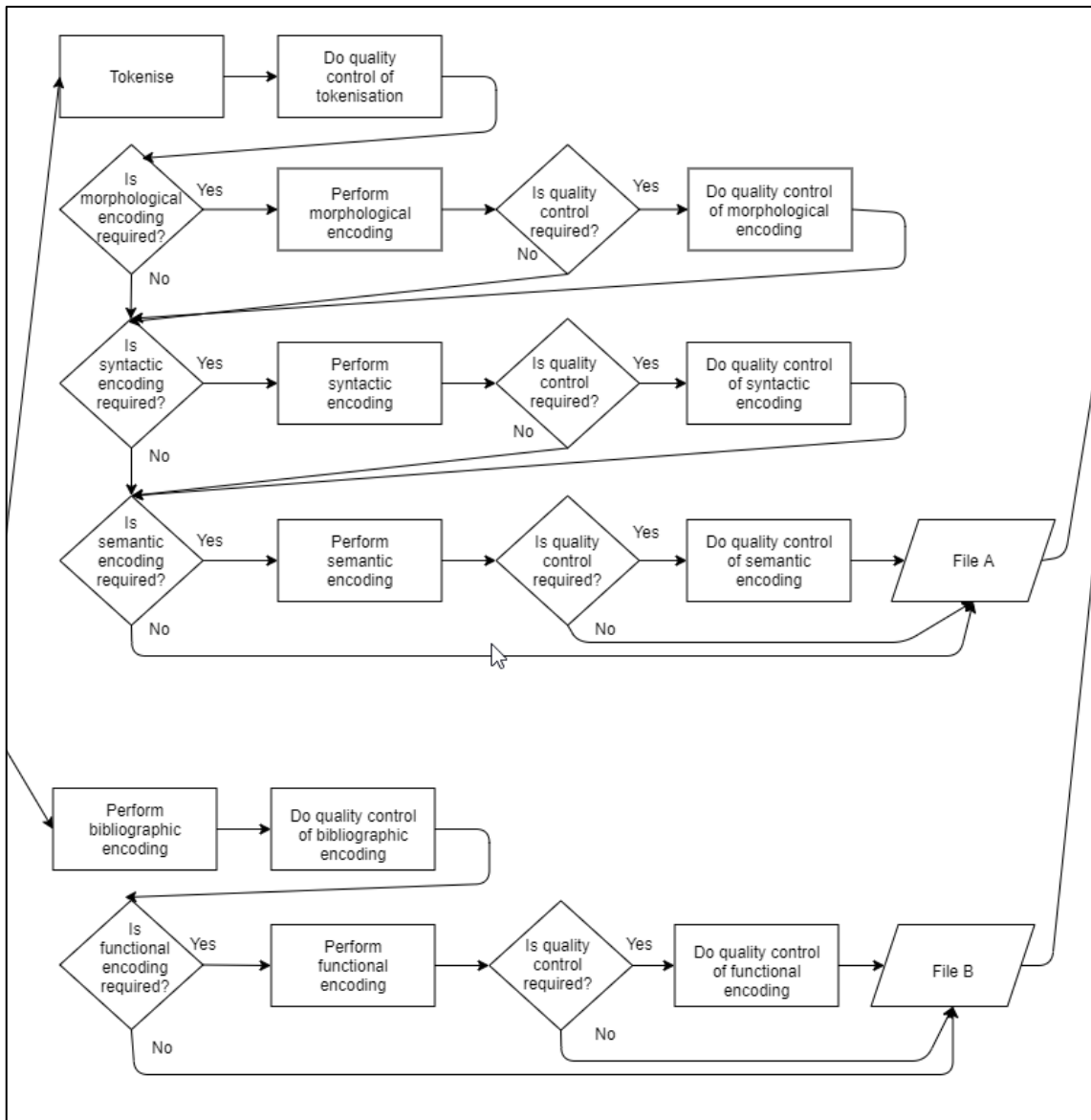


Figure 182 Enlargement: Second part of decision support system

The encoding phase has two parts that can occur in parallel. In the first part, the morphological, syntactic and semantic metadata are encoded, and the first file is produced. In the second part, the bibliographic and functional encoding are performed, and the second file is produced.

When encoding the first file, one should firstly tokenise the text to identify sentences and words. This can be done through an automated program. The correct identification of words is critical to the success of the system, as the system compares the words in

the first file to the words in the second file and so links to the different properties for each word. It is therefore important to check the quality of this.

The first level of encoding for this file is the morphological level. One should first determine if encoding on this level is necessary. If not, one can proceed to the next level, otherwise, the encoding can be done. This can be done through an automated program. One should then decide if quality control should be done or if one should proceed directly to the next level.

The next level of encoding is the syntactic level. Similar to the previous level, one should check if syntactic encoding is required. If not, then one can start with the semantic encoding. The syntactic encoding, if required, can be done through an automated program. Again, one should decide if it is necessary to check the quality of the encoding.

The last level of encoding for this file is the semantic level. If no semantic encoding is necessary, then the first file has been completed and is ready to be passed to the database. The semantic coding, if required, is currently done manually. Any required quality checks should be done and then the first file will complete.

When doing the bibliographic and functional encoding, the first step is to encode the bibliographic metadata of the text. The bibliographic metadata are used to identify the text and the information provided here should be checked for accuracy (quality control).

After encoding the bibliographic metadata, one should check if functional encoding is required. If not, then the second file is complete and ready to be passed to the database, if yes, then functional encoding should be done. This is currently done manually. This step also includes the encoding of any in-text bibliographic metadata. This encoding occurs on this level, as the encoder will identify any sections in the texts that should be encoded and requires additional in-text bibliographic metadata. The next step is to decide if quality control for this level is necessary. Once any required quality checks have been done, the second file has been completed.

It is important to discuss the concept of quality control of encoding. The quality of the encoding of the functional, semantic, syntactic and morphological levels of metadata do not necessarily need to be checked. This is a decision that can be made for each level of encoding. If the quality of the encoding is paramount and few errors can be allowed, then a more rigorous encoding process will be necessary. If some errors are allowed and will not have a significant impact on the results (or can be accommodated in the analysis) then the encoding process could be less expensive and rely more on

automated tools. The size of the collections might have an impact on the decision about the quality of the encoding. A smaller collection could be encoded more strictly, which is less likely to be possible with a larger collection. There are different options for quality control. If manual encoding is done, one encoder can do the first round of encoding and then the second (possibly more experienced) encoder can check the quality and make corrections. If there are subjective issues to consider (e.g., what genre a section is) then this should be discussed and referred to a third party, if necessary. If automated encoding is done, then one or more encoder(s) can check the encoding.

After the encoding process there should be two files that can be submitted to the system.

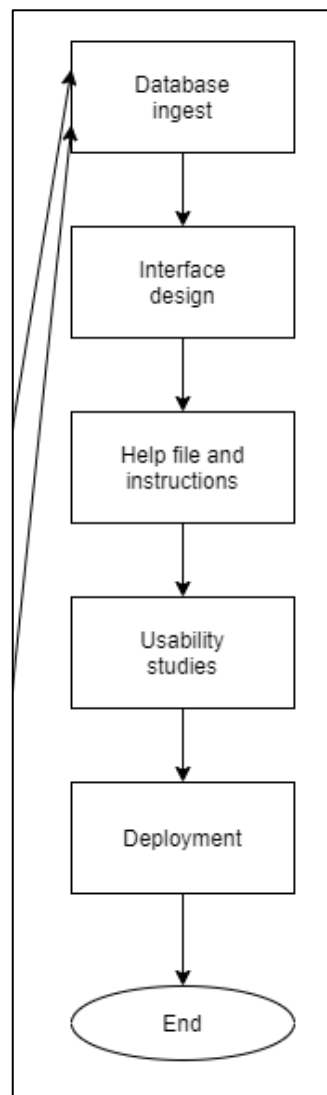


Figure 183 Enlargement: Third part of decision support system

The two files can be submitted and processed by the system so that the required information is stored in the database. The prototype stopped at this step, but three more steps are included in the decision support system to allow for further refinement.

The interface should be designed and customised to accommodate the metadata elements that have been included. (The prototype included a basic interface to allow the system to be tested; however, no visual design was done.)

Help files and examples should be developed to provide clear instructions to the users of the system. It is helpful to refer to clearly worked out examples to see how a system can work. (The prototype has several examples, but not comprehensive help files.)

Usability studies should be conducted to determine how the system is used by a sample set of researchers, to identify and correct usability problems and to suggest any improvements that can be made to the system.

The system can then be deployed to the users.

8.2.8. Recommendations and future research for the encoding of text collections to improve retrieval

The last sub-question looks at recommendations for the encoding of text collections to enable improved retrieval based in the work done in this study. Recommendations and suggestions for future work are discussed in the last part of this chapter.

8.2.9. The extent to which retrieval of words or phrases with specific properties from digital text collections can be improved by detailed encoding

By answering the first seven sub-questions in this study, the researcher can answer the main research question. This study has emphasised that retrieving words or phrases from a large text collection, without being able to filter according to certain properties, is limiting. Texts are complex structures. Although not an exhaustive list, there is information about texts as units, information about structures and features in texts, information about words in texts, and information about the relationships between words. This study has shown that information about texts and words can be made explicit through encoding. Furthermore, this information can be saved in a database and allow a user to retrieve words or phrases with specific properties.

It has been shown that the five categories of metadata suggested in this study allow detailed filtering and the ability to extract very specific instances from a text collection. The bibliographic metadata allow a user to create exactly the subset of data that should be studied, and the composition of the data is transparent. For example, it can enable a

user to select texts from a specific time period or genre. The in-text bibliographic metadata add further value, allowing users to be more specific with the subset of data that is created, especially when a single volume contains different types of text. For example, it can enable a user to exclude foreign languages from a text, or exclude introductory sections and only search in the dramas or poems. The functional metadata add a further level of filtering, allowing users to extract data based on features in the text. For example, to search only in direct speech. The semantic metadata allow a user to differentiate between instances with different meanings. The syntactic metadata encapsulate relationships between words and a user can, for example, search for instances where a word is used as a direct object. The morphological metadata capture properties about the words and enable searches on aspects such as part-of-speech categories.

At this point it might be useful to consider the scenarios used in chapter 1 as examples of searching on a detailed level and see if these examples can in fact be accommodated. It was suggested that a user might want to search for instances in direct speech as opposed to indirect speech; or only in the dramas in an anthology, not the introductory notes; or exclude foreign words used in the texts. It is clear that the fine-grained metadata in this study would allow such searches to be performed, as is illustrated in the examples and screen captures in chapter 6.

It should be mentioned that, though a graphical user interface is offered, some knowledge about linguistic analysis is necessary to make full use of the retrieval capabilities of this tool. It is possible for a user to simply select an option from a dropdown menu. However, if a user would like to get the full benefit of the metadata in the collection, the user would need to have a good linguistic understanding. The user would need to understand detailed part-of-speech categories as well as syntactic dependency grammar. It will also be necessary for the user to understand the theoretical framework used and what decisions were made in the annotation process. Thus, though it is not necessary to understand the encoding or learn a query language, a sound linguistics framework will be necessary to engage with this prototype fully. Furthermore, the prototype shows both the codes and the labels for the codes for the morphological and syntactic level. It will therefore be beneficial for the user to be familiar with the codes used on these levels.

In this way, the prototype caters for different types of users. Users who are only interested in detailed bibliographic metadata without an in-depth understanding of linguistics can still use the system very effectively. Such users can even select basic

options from the dropdown menu (such as noun) and use it in their search. However, users that have a greater understanding of linguistics may do more advanced searches, specifically on a syntactic level.

The fine-grained metadata suggested in this study allow researchers to move beyond dealing with a text as a single entity. A large dataset can be filtered to include only the data relevant to the researcher. Furthermore, the prototype that was developed also demonstrated that it is not necessary to understand a complex query language or know about the structure of the dataset, but that an intuitive interface for retrieval on a detailed level is possible. Such advanced retrieval, without programming expertise, can add significant value in the field of digital humanities. Being able to search and filter using specific properties could enable researchers in the humanities to ask more complex questions and retrieve only relevant information to their specific information need, and so improve the analysis of word usage and text analysis.

8.3. Recommendations and future research

In this section, the researcher will make recommendations for the encoding of text collections with detailed metadata. The recommendations are based on the findings of this study.

8.3.1. Encode texts with detailed metadata to enhance retrieval

This study has shown that it is possible to retrieve words or phrases with specific properties from a digital text collection when texts are encoded with detailed metadata, and thus allow for powerful retrieval. The first recommendation of this study is therefore that texts should be encoded with fine-grained metadata.

This study recommends that metadata from different categories should be used. Metadata applied to texts should not be limited to bibliographic information about a text as a whole. Although such metadata are important, metadata should also reveal features and structures in a text. Metadata should also indicate when sections in a text have different bibliographic properties to the text itself. Some of the examples explored in this study are quotes by other authors, introductions written by other authors or poems included in novels. Furthermore, encoding information about individual words can lead to advanced searches.

8.3.2. Identify more metadata useful for encoding

Detailed metadata can enhance retrieval in text collections. In this study various categories of metadata that could be useful to enhance retrieval were recommended. There is more information that could be encoded, particularly on the functional level.

The encoding used in this study could be extended. It would be an advantage to have metadata that are useful to researchers in different fields and disciplines. It is recommended that the metadata fields that would benefit most researchers are identified and included in the metadata suggested in this study. It is recommended that not all information that can possibly be encoded is used, as this will have an impact on the time that it takes to encode texts as well as the interface that needed to allow a user to search in these fields. A set of elements should be identified and included.

8.3.3. Develop tools that can utilise metadata for improved retrieval

The prototype that was developed in this study demonstrates that retrieval can be enhanced when the metadata from texts are used in the retrieval process. By using metadata, this tool allows users to filter their search according to very specific properties. It is recommended that more similar tools are developed, enabling users to create complex queries. If the data can be manipulated and filtered to create subsets of data exactly to the researcher's requirements, more trustworthy analyses can be made.

It is also recommended that more features be added to make the tool more useful. For example, more ways to view the results could be explored. It has been evident that viewing the frequency of words over time on a graph is useful. Due to the small dataset that was used in this prototype a graph could not be included but could be a valuable feature. Other ways to sort and group results could be investigated when a large number of encoded texts is available. For example, similar results could be grouped together and then when a user clicks on a group the individual instances are visible. More could be done with the results as well. For example, it could be considered to export the results so that analysis outside the system could be done.

8.3.4. Accommodate laypersons and advanced users when designing tools for retrieval

Tools that require a user to write queries in a complex query language or require knowledge of encoding standards or knowledge of the structure of the underlying data, may exclude many people and prohibit laypersons from conducting sensible queries on a large text collection. On the other hand, tools that are very simplistic in nature may prevent advanced users from asking complex questions.

It is recommended that both types of users are accommodated as far as possible. The prototype in this study showed a way in which both laypersons and advanced users could be accommodated. The graphical user interface is available so that users do not need to know codes or a query language syntax but can still build sensible queries by using the dropdown menus and check boxes. Advanced users could also make use of

the graphical user interface but could learn the query language and write queries more efficiently.

8.3.5. Improve tools used for automated encoding

A further recommendation of this study is that ways to automate the encoding of texts should be improved. Much work in this regard has already been done, but as was seen in this study, there are still areas that could be improved. This is particularly important in areas where a type of encoding relies on other types of encoding. For example, syntactic annotation makes use of morphological annotation. Incorrect sentence segmentation or word tokenisation will also influence further annotations. As each level of annotation is improved, further levels could be improved.

8.3.6. Explore the scalability of the solution

This study set out to prove a concept and did so. As was discussed earlier, the system that was developed is therefore not a final product that is ready to be used by the public. Future work could consider the viability of turning such a tool into a fully-fledged system that can accommodate specific text collections in accordance with specific scholars' research requirements. This would require research into the scalability of the system used in this study. In the prototype, many links are created between data in the database. It would be necessary to determine how and if the system is scalable.

8.4. Conclusion

The various digital text collections that are available offer exciting opportunities to users for research in the digital humanities. Technology is allowing users to engage with these collections in ways not previously possible. This study suggests that there could be still more developments to improve searching in digital text collections. This study particularly considered the way in which detailed metadata could be used to improve retrieval. The results of this study showed that detailed metadata could indeed be used to enhance retrieval. Detailed metadata allow users to filter results according to specific properties and retrieve exactly what they need.

This study concludes then that enhancing texts with detailed metadata will allow users to select only that which is relevant in terms of their information need and thus improve retrieval.

9. References

- Acerbi, A., Lampos, V., Garnett, P. & Bentley, R.A. 2013. The Expression of Emotions in 20th Century Books. *PLOS One*, 8(3): e59030.
- ACL Wiki. 2019. *POS Tagging (State of the art)* [Online]. Available: [https://aclweb.org/aclwiki/POS_Tagging_\(State_of_the_art\)](https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)) [Accessed on 26 March 2020].
- Agirre, E., de Lacalle, O.L. & Soroa, A. 2018. The risk of sub-optimal use of Open Source NLP Software: UKB is inadvertently state-of-the-art in knowledge-based WSD. *arXiv preprint arXiv:1805.04277*.
- Agirre, E., López de Lacalle, O. & Soroa, A. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1): 57–84.
- Al Omran, F.N.A. & Treude, C. 2017. Choosing an NLP library for analyzing software documentation: A systematic literature review and a series of experiments. *Proceedings of the 14th International Conference on Mining Software Repositories, 2017*: 187–197. IEEE Press.
- Anthony, L. 2013. A critical look at software tools in corpus linguistics. *Linguistic Research*, 30(2): 141–161.
- Baksik, C. 2006. Fair Use or Exploitation? The Google Books Search Controversy. *Libraries and the Academy*, 6(4): 399–415.
- Band, J. & Gerafi, J. 2015. *The Fair Use/Fair Dealing Handbook* [Online]. Available: <http://infojustice.org/wp-content/uploads/2015/03/fair-use-handbook-march-2015.pdf> [Accessed on 27 October 2020].
- Banerjee, K. & Reese, T. 2018. *Building Digital Libraries*, 2nd edition, Chicago: ALA Neal-Schuman.
- Barnum, C.M. 2010. *Usability Testing Essentials*, Burlington: Morgan Kaufmann.
- Bauman, S. & Flanders, J. 2018. *Overview of TEI Customization* [Online]. Available: https://www.wwp.northeastern.edu/outreach/seminars/_current/presentations/customization/customization_overview_tutorial_00.xhtml [Accessed on 9 January 2019].
- Bauman, S., Hoover, D., van Dalen-Oskam, K. & Piez, W. 2012. *Text Analysis Meets Text Encoding* [Online]. Available: <http://www.dh2012.uni-hamburg.de/conference/programme/abstracts/text-analysis-meets-text-encoding.1.html> http://www.ur.umich.edu/0506/Feb13_06/02.shtml [Accessed on 2 February 2018].

- Bergquist, K. 2006. *Google project promotes public good*. [Online]. Available: http://www.ur.umich.edu/0506/Feb13_06/02.shtml [Accessed on 11 January 2018].
- Berzak, Y., Huang, Y., Barbu, A., Korhonen, A. & Katz, B. 2016. Anchoring and agreement in syntactic annotations. *arXiv preprint arXiv:1605.04481*.
- Bhattacharyya, S., Organisciak, P. & Downie, J.S. 2015. A Fragmentizing Interface to a Large Corpus of Digitized Text: (Post) humanism and Non-consumptive Reading via Features. *Interdisciplinary Science Reviews*, 40(1): 61–77.
- Bird, S., Klein, E. & Loper, E. 2015. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. [Online]. Available: <http://www.nltk.org/book/> [Accessed on 17 May 2018].
- Blevins, J.P. 2013. Word-based morphology from Aristotle to modern WP (word and paradigm models). In: Allan, K. (ed.) *Oxford Handbook of the History of Linguistics*. Oxford: Oxford University Press: 375–395.
- BNC. 2018. *The British National Corpus 2014: User manual and reference guide* [Online]. Available: <http://corpora.lancs.ac.uk/bnc2014/doc/BNC2014manual.pdf> [Accessed on 24 April 2019].
- BNCweb (CQP-edition). 2008. *BNCweb (CQP-edition) – A web-based interface to the British National Corpus* [Online]. Available: <http://corpora.lancs.ac.uk/BNCweb/> [Accessed on 26 June 2018].
- BNCweb (CQP-edition). n.d. *Simple Query Syntax – Cheat Sheet* [Online]. Available: http://bncweb.lancs.ac.uk/bncwebXML/Simple_query_language.pdf [Accessed on 26 June 2018].
- Bode, K. 2017. The equivalence of “close” and “distant” reading; or, toward a new object for data-rich literary history. *Modern Language Quarterly*, 78(1): 77–106.
- Burnard, L. 1995. *Markup and Markup Languages* [Online]. Available: <https://tei-c.org/Vault/ED/EDW25/W25C.htm> [Accessed on 22 January 2021].
- Burnard, L. 2007. *BNC User Reference Guide* [Online]. Available: <http://www.natcorp.ox.ac.uk/docs/URG/index.html> [Accessed on 15 February 2019].
- Burnard, L. 2009. *British National Corpus* [Online]. Available: <http://www.natcorp.ox.ac.uk/corpus/index.xml> [Accessed on 15 February 2019].
- BYU Corpora – EEBO. n.d. *Early English Books Online* [Online]. Available: <https://www.english-corpora.org/eebo/> [Accessed on 24 April 2019].

- Capitanu, B., Underwood, T., Organisciak, P., Cole, T., Sarol, M.J. & Downie, J.S. 2016. *The HathiTrust Research Center Extracted Feature Dataset (1.0)*. HathiTrust Research Center. <http://dx.doi.org/10.13012/J8X63JT3>.
- Choi, J.D., Tetreault, J. & Stent, A. 2015. It depends: Dependency parser comparison using a web-based evaluation tool. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 2015*: 387–396.
- Christenson, H. 2011. HathiTrust: A Research Library at Web Scale. *Association for Library Collections & Technical Services*, 55(2): 93–102.
- CLEF. 2020. *The CLEF Initiative (Conference and Labs of the Evaluation Forum)* [Online]. Available: <http://www.clef-initiative.eu/> [Accessed on 30 January 2021].
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. & Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12: 2493–2537.
- Crane, G. 1998. The Perseus Project and Beyond: How Building a Digital Library Challenges the Humanities and Technology. *D-Lib Magazine*, 4(1).
- Cristani, M., Bertolaso, A., Scannapieco, S. & Tomazzoli, C. 2018. Future paradigms of automated processing of business documents. *International Journal of Information Management*, 40: 67–75.
- Culturomics. 2017. *Culturomics – FAQ* [Online]. Available: <http://www.culturomics.org/Resources/faq> [Accessed on 1 December 2017].
- Cummings, J. 2013. *An Introduction to the Text Encoding Initiative*. [Online]. Available: <https://prezi.com/s8rqk-xdpzdb/an-introduction-to-the-text-encoding-initiative/> [Accessed on 12 January 2018].
- Cummings, J. 2016. *An Overview of TEI Metadata* [Online]. Available: <https://prezi.com/s1e37cij0i0p/an-overview-of-tei-metadata/> http://www.ur.umich.edu/0506/Feb13_06/02.shtml [Accessed on 2 February 2018].
- Davies, M. 2014. Making Google Books n-grams useful for a wide range of research on language change. *International Journal of Corpus Linguistics*, 19(3): 401–416.
- Davies, M. n.d. *corpus.byu.edu* [Online]. Available: <https://corpus.byu.edu/> [Accessed on 21 February 2019].
- Dhandapani, S. 2016. Integration of User Centered Design and Software Development Process. *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016*: 1–5. IEEE.

- Dobрева, M., Kim, Y. & Ross, S. 2013. *Automated Metadata Generation* [Online]. Available: <http://www.dcc.ac.uk/resources/curation-reference-manual/completed-chapters/automated-metadata-extraction> [Accessed on 6 December 2019].
- Drucker, J. 2013. *6A. Text Encoding: Mark-up and TEI*. [Online]. Available: http://dh101.humanities.ucla.edu/?page_id=60 [Accessed on 23 January 2018].
- EEBO. n.d. *Early English Books Online* [Online]. Available: <https://eebo.chadwyck.com> [Accessed on 31 July 2018].
- Elson, D.K. & McKeown, K.R. 2010. Automatic attribution of quoted speech in literary narrative. *Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010*: 1013–1019. AAAI Press.
- ESRC Centre for Corpus Approaches to Social Science (CASS). n.d. *BNC2014* [Online]. Available: <http://cass.lancs.ac.uk/bnc2014/> [Accessed on 15 February 2019].
- Evert, S. 2005. *The CQP Query Language Tutorial* [Online]. Available: <http://www.ims.uni-stuttgart.de/forschung/projekte/CorpusWorkbench/CQPTutorial/cqp-tutorial.2up.pdf> [Accessed on 19 February 2019].
- Fenlon, K., Senseney, M., Green, H., Bhattacharyya, S., Willis, C. & Downie, J. 2014. Scholar-built collections: A study of user requirements for research in large-scale digital libraries. *Proceedings of the Association for Information Science and Technology*, 51(1): 1–10.
- Finlayson, M.A. 2015. ProppLearner: Deeply annotating a corpus of Russian folktales to enable the machine learning of a Russian formalist theory. *Digital Scholarship in the Humanities*, 32(2): 284–300.
- Forster, C. 2015. *A Walk Through the Metadata: Gender in the HathiTrust Dataset* [Online]. Available: <http://cforster.com/2015/09/gender-in-hathitrust-dataset/> [Accessed on 30 July 2019].
- Friginal, E., Walker, M. & Randall, J.B. 2014. Exploring mega-corpora: Google Ngram Viewer and the Corpus of Historical American English. *EuroAmerican Journal of Applied Linguistics and Languages*, 1(1): 48–68.
- Gallant, K., Lorang, E. & Ramirez, A. 2014. *Tools for the digital humanities: A librarian's guide* [Online]. Available: <https://hdl.handle.net/10355/44544> [Accessed on 25 September 2020].
- Gander, L., Lezuo, C. & Unterweger, R. 2011. Rule based document understanding of historical books using a hybrid fuzzy classification system. *Proceedings of the*

- 2011 Workshop on Historical Document Imaging and Processing, 2011: 91–97. ACM.
- Gao, L., Zhong, Y., Tang, Y., Tang, Z., Lin, X. & Hu, X. 2011. Metadata Extraction System for Chinese Books. *2011 International Conference on Document Analysis and Recognition, 2011*: 749–753. IEEE.
- Gilliland, A.J. 2016. Setting the Stage. In: Baca, M. (ed.) *Introduction to Metadata*. Los Angeles: Getty Research Institute.
- Goddard, C. & Schalley, A.C. 2010. Semantic Analysis. In: Indurjha, N. & Damerau, F. J. (eds.) *Handbook of Natural Language Processing*. 2nd ed. Boca Raton, FL: CRC Press: 93–120.
- Gómez-Rodríguez, C., Alonso-Alonso, I. & Vilares, D. 2019. How important is syntactic parsing accuracy? An empirical evaluation on rule-based sentiment analysis. *Artificial Intelligence Review*, 52(3): 2081–2097.
- Gooding, P. 2013. Mass digitization and the garbage dump: The conflicting needs of quantitative and qualitative methods. *Literary and Linguistic Computing*, 28(3): 425–431.
- Gooding, P., Terras, M. & Warwick, C. 2013. The myth of the new: Mass digitization, distant reading, and the future of the book. *Literary and Linguistic Computing*, 28(4): 629–639.
- Gooding, P., Warwick, C. & Terras, M. 2012. *The Myth of the New: Mass Digitization, Distant Reading and the Future of the Book* [Online]. Hamburg, Germany. Available: <http://www.dh2012.uni-hamburg.de/conference/programme/abstracts/the-myth-of-the-new-mass-digitization-distant-reading-and-the-future-of-the-book.1.html> [Accessed on 25 September 2020].
- Goodwin, J. 2015. *Creating a Topic Browser of HathiTrust Data* [Online]. Available: <https://jgoodwin.net/blog/creating-hathitrust-topic-browser/> [Accessed on 30 July 2019].
- Google Books Ngram Viewer Info. 2020. *Google Books Ngram Viewer Info* [Online]. Available: <https://books.google.com/ngrams/info> [Accessed on 18 August 2020].
- Goutte, C., Léger, S. & Carpuat, M. 2014. The NRC system for discriminating similar languages. *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects, Dublin, Ireland, 2014*: 139–145.
- Goyal, A., Gupta, V. & Kumar, M. 2018. Recent named entity recognition and classification techniques: A systematic review. *Computer Science Review*, 29: 21–43.

- Grana, C., Serra, G., Manfredi, M., Coppi, D. & Cucchiara, R. 2016. Layout analysis and content enrichment of digitized books. *Multimedia Tools and Applications*, 75(7): 3879–3900.
- Grant, L.E. 2005. Frequency of ‘core idioms’ in the British National Corpus (BNC). *International Journal of Corpus Linguistics*, 10(4): 429–451.
- Greenberg, J. 2004. Metadata extraction and harvesting: A comparison of two automatic metadata generation applications. *Journal of Internet Cataloging*, 6(4): 59–82.
- Gregory, I. 2014. Challenges and Opportunities for Digital History. *Frontiers in Digital Humanities*, 1(1): 1.
- Gregory, I.N., Atkinson, P.D., Hardie, A., Joulain-Jay, A., Kershaw, D., Porter, C., Rayson, P.E. & Rupp, C.J. 2016. From digital resources to historical scholarship with the British Library 19th Century Newspaper Collection. *Journal of Siberian Federal University: Humanities and Social Sciences*, 9(4): 994–1006.
- Griffis, D., Shivade, C., Fosler-Lussier, E. & Lai, A.M. 2016. A quantitative and qualitative evaluation of sentence boundary detection for the clinical domain. *AMIA Summits on Translational Science Proceedings*, 2016: 88.
- Hanington, B. & Martin, B. 2012. *Universal Methods of Design*, Beverly, Massachusetts: Rockport Publishers Inc.
- Hao, H., Li, N., Tian, Y. & Geng, S. Re-flowable Document Structure Understanding by Comprehensive Use of Features and Rules. *International Conference on Computer, Communication and Network Technology (CCNT 2018), Wuzhen, China, 2018*: 291–297. DEStech Publications.
- Hardie, A. 2012. CQPweb – combining power, flexibility and usability in a corpus analysis tool. *International journal of corpus linguistics*, 17(3): 380–409.
- Hardie, A. 2014. Modest XML for Corpora: Not a standard, but a suggestion. *ICAME Journal*, 38(1): 73–103.
- HathiTrust. 2020. *HathiTrust Digital Library About* [Online]. Available: <https://www.hathitrust.org/about> [Accessed on 27 July 2020].
- HathiTrust Digital Library – Bibliographic metadata specifications. n.d. *HathiTrust Digital Library – Bibliographic metadata specifications* [Online]. Available: https://www.hathitrust.org/bib_specifications [Accessed on 12 June 2018].

- HathiTrust Digital Library – Guidelines for Digital Object Deposit. 2011. *Guidelines for Digital Object Deposit* [Online]. Available: https://www.hathitrust.org/deposit_guidelines [Accessed on 3 August 2018].
- HathiTrust Digital Library – Our Digital Library. n.d. *Our Digital Library* [Online]. Available: https://www.hathitrust.org/digital_library [Accessed on 3 August 2018].
- HathiTrust Digital Library – Our Partnership. n.d. *Our Partnership* [Online]. Available: <https://www.hathitrust.org/partnership> [Accessed on 3 August 2018].
- HathiTrust Research Center. n.d. *HathiTrust Research Center – About* [Online]. Available: <https://analytics.hathitrust.org/about> [Accessed on 25 September 2020].
- Haynes, D. 2018. *Metadata for Information Management and Retrieval: Understanding Metadata and Its Use*, London: Facet Publishing.
- Heiden, S. 2010. The TXM platform: Building open-source textual analysis software compatible with the TEI encoding scheme. *24th Pacific Asia conference on language, information and computation, Sendai, Japan, 2010*: 389–398. Institute for Digital Enhancement of Cognitive Development, Waseda University.
- Henry, C. & Smith, K. 2010. Ghostlier Demarcations: Large-Scale Text Digitization Projects and Their Utility for Contemporary Humanities Scholarship. In: Bishop, A., Clotfelter, C., Friedlander, A., Gift, D. M., Holly, A., Lynch, C., McPherson, M., Moore, C., Nichols, S. & Williams, J. (eds.) *The Idea of Order: Transforming Research Collections for 21st Century Scholarship*. Washington, DC: Council on Library and Information Resources: 106–115.
- Heuser, R., Moretti, F. & Steiner, E. 2016. *The Emotions of London. Pamphlet 13*. [Online]. Stanford Literary Lab. Available: <https://litlab.stanford.edu/LiteraryLabPamphlet13.pdf> [Accessed on 2 August 2018].
- Heyman, S. 2015. *Google Books: A Complex and Controversial Experiment*. [Online]. Available: <https://www.nytimes.com/2015/10/29/arts/international/google-books-a-complex-and-controversial-experiment.html> [Accessed on 6 October 2017].
- Hippisley, A. 2010. Lexical analysis. In: Indurjha, N. & Damerau, F. J. (eds.) *Handbook of Natural Language Processing*. 2nd ed. Boca Raton, FL: CRC Press: 31–58.
- Hitchcock, T. 2013. Confronting the digital: Or how academic history writing lost the plot. *Cultural and Social History*, 10(1): 9–23.

- Hodge, G. 2001. *Metadata made simpler* [Online]. NISO. Available: http://qjfb0520.sid.inpe.br/col/dpi.inpe.br/banon/2004/04.21.12.47/doc/Metadata_simpler.pdf [Accessed on 21 January 2021].
- Hoffmann, S. & Evert, S. 2006. BNCweb (CQP-edition): The marriage of two corpus tools. *Corpus technology and language pedagogy: New resources, new tools, new methods*, 3: 177–195.
- Hofstee, E. 2006. *Constructing a good dissertation: A practical guide to finishing a Master's, MBA or PhD on schedule*, Sandton, South Africa: EPE.
- Hovy, E. & Lavid, J. 2010. Towards a 'science' of corpus annotation: A new methodological challenge for corpus linguistics. *International Journal of Translation*, 22(1): 13–36.
- Howard, J. 2017. *What Happened to Google's Effort to Scan Millions of University Library Books?* [Online]. Available: <https://www.edsurge.com/news/2017-08-10-what-happened-to-google-s-effort-to-scan-millions-of-university-library-books> [Accessed on 30 January 2020].
- Iacobacci, I., Pilehvar, M.T. & Navigli, R. 2016. Embeddings for word sense disambiguation: An evaluation study. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 2016*: 897–907.
- Internet Archive. n.d. *Internet Archive* [Online]. Available: <https://archive.org> [Accessed on 17 August 2020].
- Interset. n.d. *Interset: Interlingua for Morphosyntactic Tagsets* [Online]. Available: <http://ufal.mff.cuni.cz/interset> [Accessed on 27 August 2018].
- Jauhiainen, T.S., Lui, M., Zampieri, M., Baldwin, T. & Lindén, K. 2019. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65: 675–782.
- Jenkins, T. 2013. *Don't count on big data for answers*. [Online]. Available: <https://www.scotsman.com/news/opinion/tiffany-jenkins-don-t-count-on-big-data-for-answers-1-2785890> [Accessed on 23 July 2019].
- Jett, J., Cole, T.W., Maden, C. & Downie, J.S. 2016a. The HathiTrust Research Center Workset Ontology: A Descriptive Framework for Non-Consumptive Research Collections. *Journal of Open Humanities Data*, 2(e1): 1–7.
- Jett, J., Nurmikko-Fuller, T., Cole, T.W., Page, K.R. & Downie, J.S. 2016. Enhancing scholarly use of digital libraries: A comparative survey and review of bibliographic metadata ontologies. *Proceedings of the 16th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL), New York, USA, 2016b*: 35–44. ACM.

- Jockers, M.L. 2010. *Unigrams, and Bigrams, and Trigrams, Oh My*. [Online]. Available: <http://www.matthewjockers.net/2010/12/22/unigrams-and-bigrams-and-trigrams-oh-my/> [Accessed on 11 January 2018].
- Jockers, M.L. 2011. Detecting and Characterizing National Style in the 19th Century Novel. *Digital Humanities 2011, Stanford University, Stanford, CA, USA, 19-22 June 2011*: 159–160. Stanford University Library.
- Juola, P. 2013. Using the Google N-Gram corpus to measure cultural complexity. *Literary and linguistic computing*, 28(4): 668–675.
- Jurafsky, D. & Martin, J.H. 2017. *Speech and Language Processing* [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/> [Accessed on 3 April 201].
- Kahle, B. 2018. *Digital Books on archive.org* [Online]. Available: <https://blog.archive.org/2018/01/24/digital-books-on-archive-org/> [Accessed on 27 July 2018].
- Kahle, B. & Vadillo, A.P. 2015. The Internet Archive: An Interview. *19: Interdisciplinary Studies in the Long Nineteenth Century*, 21: 1–15.
- Kennedy, G. 2003. Amplifier collocations in the British National Corpus: Implications for English language teaching. *Tesol Quarterly*, 37(3): 467–487.
- Keuleers, E., Brysbaert, M. & New, B. 2011. An evaluation of the Google Books ngrams for psycholinguistic research. In: Würzner, K. & Pohl, E. (eds.) *Potsdam Cognitive Science Series 3 – Lexical Resources in Psycholinguistic Research*. Postdam, Germany: Universitätsverlag Postdam: 23–27.
- Khemakhem, M., Foppiano, L. & Romary, L. 2017. Automatic extraction of TEI structures in digitized lexical resources using conditional random fields. *eLex 2017: Proceedings of eLex 2017 conference, Electronic lexicography in the 21st century, Leiden, Netherlands, 19-21 September 2017*. Lexical Computing CZ.
- Kinnaman, A. & Koehl, E.D. 2018. *Extracted Features Dataset* [Online]. Available: <https://wiki.htrc.illinois.edu/display/COM/Extracted+Features+Dataset> [Accessed on 18 March 2019].
- Kirschenbaum, M. 2013. What is Digital Humanities and What's it doing in our English Departments? In: Terras, M., Nyhan, J. & Vanhoutte, E. (eds.) *Defining Digital Humanities*. Surrey, England: Ashgate: 195–204.
- Kitchin, R. 2014. Big Data, new epistemologies and paradigm shifts. *Big data & society*, 1(1): 1–12.

- Kjeldskov, J., Skov, M.B. & Stage, J. 2010. A longitudinal study of usability in health care: Does time heal? *International Journal of Medical Informatics*, 79(6): e135–e143.
- Klimczak, E. 2013. *Design for Software: A Playbook for Developers*, West Sussex: John Wiley & Sons, Incorporated.
- Koch, S., John, M., Wörner, M., Müller, A. & Ertl, T. 2014. VarifocalReader—in-depth visual analysis of large text documents. *IEEE transactions on visualization and computer graphics*, 20(12): 1723–1732.
- Koplenig, A. 2017. The impact of lacking metadata for the measurement of cultural and linguistic change using the Google Ngram data sets—Reconstructing the composition of the German corpus in times of WWII. *Digital Scholarship in the Humanities*, 32(1): 169–188.
- Kumar, R. 2014. *Research methodology: A step-by-step guide for beginners*, 4th edition, Los Angeles: SAGE.
- Lansdall-Welfare, T., Sudhakar, S., Thompson, J., Lewis, J., FindMyPast Newspaper Team & Cristianini, N. 2017. Content analysis of 150 years of British periodicals. *Proceedings of the National Academy of Sciences*, 114(4): E457–E465.
- Lee, D. 2001. Genres, registers, text types, domains and styles: Clarifying the concepts and navigating a path through the BNC jungle. *Language Learning and Technology*, 5(3): 37–72.
- Leech, G., Barnett, R. & Kahrel, P. 1996. *EAGLES Recommendations for the Syntactic Annotation of Corpora EAG-TCWG-SASG/1.8 – Phrase structure vs dependency* [Online]. Available: <http://www.ilc.cnr.it/EAGLES96/segsasg1/node44.html> [Accessed on 22 August 2018].
- Leedy, P.D. & Ormrod, J.E. 2014. *Practical research. Planning and design*, 10th ed, New Jersey: Pearson Education.
- Leedy, P.D. & Ormrod, J.E. 2020. *Practical Research: Planning and Design, Global Edition*, Essex: Pearson Education Limited.
- Leetaru, K. 2008. Mass book digitization: The deeper story of Google Books and the Open Content Alliance. *First Monday*, 13(10).
- Leetaru, K. 2011. Culturomics 2.0: Forecasting large-scale human behavior using global news media tone in time and space. *First Monday*, 16(9).

- Library of Congress. 2016. *Metadata Object Description Schema (MODS)* [Online]. Available: <http://www.loc.gov/standards/mods/mods-overview.html> [Accessed on 28 September 2018].
- Lin, Y., Michel, J., Aiden, E.L., Orwant, J., Brockman, W. & Petrov, S. 2012. Syntactic Annotations for the Google Books Ngram Corpus. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju, Republic of Korea, 8-14 July 2012*: 169–174. Association for Computational Linguistics.
- Lopatin, L. 2006. Library digitization projects, issues and guidelines. *Library Hi Tech*, 24(2): 273–289.
- Loria, S. 2013. *Tutorial: What is WordNet? A Conceptual Introduction Using Python* [Online]. Available: <https://stevenloria.com/wordnet-tutorial/> [Accessed on 24 August 2018].
- Love, R., Demby, C., Hardie, A., Brezina, V. & McEnery, T. 2017. The Spoken BNC2014. *International Journal of Corpus Linguistics*, 22(3): 319–344.
- Lu, X., Kahle, B., Wang, J.Z. & Giles, C.L. 2008. A metadata generation system for scanned scientific volumes. *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries, Pittsburgh, USA, June 2008*: 167–176. ACM.
- Lui, M., Lau, J.H. & Baldwin, T. 2014. Automatic detection and language identification of multilingual documents. *Transactions of the Association for Computational Linguistics*, 2: 27–40.
- Manning, C.D. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In: Gelbukh, A. F., (ed.) *Proceedings of CICLing (Computational Linguistics and Intelligent Text Processing), Tokyo, Japan, 20-26 February 2011*: 171–189. Berlin, Heidelberg: Springer.
- Manning, C.D. 2014. *Dependency Grammar – Introduction* [Online]. Available: <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1162/handouts/SLoSP-2014-4-dependencies.pdf> [Accessed on 18 September 2018].
- Manning, C.D., Raghavan, P. & Schütze, H. 2008a. *Introduction to Information Retrieval – Stemming and lemmatization*. [Online]. Available: <https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html> [Accessed on 9 April 2018].
- Manning, C.D., Raghavan, P. & Schütze, H. 2008b. *Introduction to Information Retrieval – Tokenization*. [Online]. Available: <https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html> [Accessed on 3 April 2018].
- Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J. & McClosky, D. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of*

52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, Maryland, USA, June 2014: 55–60.

- Marcus, M.P., Marcinkiewicz, M.A. & Santorini, B. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2): 313–330.
- Marinai, S. 2009. Metadata extraction from PDF papers for digital library ingest. *Proceedings of the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 26-29 July 2009: 251–255.* IEEE.
- Marr, B. 2018. *How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read* [Online]. Available: <https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read> [Accessed on 2 October 2020].
- Mason, A. 2015. *Text Analysis and the TEI*. [Online]. Available: <http://blogs.carleton.edu/hacking-humanities/2015/01/29/8-text-analysis-and-the-tei/> [Accessed on 23 January 2018].
- McGregor, W.B. 2009. *Linguistics: An Introduction*, London: Continuum International Publishing Group.
- McNamee, P. 2005. Language identification: A solved problem suitable for undergraduate instruction. *Journal of Computing Sciences in Colleges*, 20(3): 94–101.
- Meyer, E.T. & Eccles, K. 2016. *The Impacts of Digital Collections: Early English Books Online & House of Commons Parliamentary Papers* [Online]. London: Jisc. Available: <https://ssrn.com/abstract=2740299> [Accessed on 14 August 2018].
- Michel, J., Shen, Y.K., Aiden, A.P., Veres, A., Gray, M.K., The Google Books Team, Pickett, J.P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M.A. & Lieberman, A.E. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014): 176–182.
- Miller, D.C. & Salkind, N.J. 2002. *Handbook of Research Design & Social Measurement*, Thousand Oaks, CA: SAGE Publications, Inc.
- Miller, T., Erbs, N., Zorn, H., Zesch, T. & Gurevych, I. 2013. DKPro WSD: A generalized UIMA-based framework for word sense disambiguation. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Sofia, Bulgaria, August 2013: 37–42.* Association for Computational Linguistics.
- Mimno, D. 2014. *Word counting, squared*. [Online]. Available: <http://www.mimno.org/articles/wordsim/> [Accessed on 30 July 2019].

- Mittelbach, A. & Rahtz, S. 2018. *TEI Roma: Generating validators for the TEI* [Online]. Available: <http://roma.tei-c.org> [Accessed on 10 January 2019].
- Mitton, R., Hardcastle, D. & Pedler, J. 2007. BNC! Handle with care! Spelling and tagging errors in the BNC. *Paper presented at the Fourth Corpus Linguistics Conference, Birmingham, U.K., 27-30 July 2007.*
- Mohit, B. 2014. Named Entity Recognition. In: Zitouni, I. (ed.) *Natural Language Processing of Semitic Languages*. Berlin: Springer-Verlag: 221–246.
- Molich, R. & Dumas, J.S. 2008. Comparative usability evaluation (CUE-4). *Behaviour & Information Technology*, 27(3): 263–281.
- Moretti, F. 2003. Graphs, Maps, Trees. *New Left Review*, 24: 67–93.
- Moro, A., Raganato, A. & Navigli, R. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2: 231–244.
- Mouton, J. 2001. *How to succeed in your master's and doctoral studies : A South African guide and resource book*, Pretoria: Van Schaik.
- MUC-7. 2007. *MUC-7 (State of the art)* [Online]. Available: [https://aclweb.org/aclwiki/MUC-7_\(State_of_the_art\)](https://aclweb.org/aclwiki/MUC-7_(State_of_the_art)) [Accessed on 3 December 2019].
- Murdock, J., Jett, J., Cole, T., Ma, Y., Downie, J.S. & Plale, B. 2017. Towards publishing secure capsule-based analysis. *Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries, Toronto, Ontario, Canada, June 2017*: 261–264. Piscataway, NJ, USA: IEEE Press.
- Nadeau, D. & Sekine, S. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1): 3–26.
- Navigli, R. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2): 10.
- Nicholson, B. 2012. Counting culture; or, how to read Victorian newspapers from a distance. *Journal of Victorian Culture*, 17(2): 238–246.
- Nicholson, B. 2013. The Digital Turn: Exploring the methodological possibilities of digital newspaper archives. *Media History*, 19(1): 59–73.
- Nielsen, J. 1995. *Ten usability heuristics*. [Online]. Available: <http://www.nngroup.com/articles/ten-usability-heuristics/> [Accessed on 20 August 2020].

- Nivre, J., De Marneffe, M., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R.T., Petrov, S., Pyysalo, S. & Silveira, N. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, May 2016*: 1659–1666. European Language Resources Association (ELRA).
- Nolas, S. 2011. Grounded Theory Approaches. In: Frost, N. (ed.) *Qualitative Research Methods in Psychology: Combining Core Approaches*. Berkshire: McGraw-Hill Education: 16-43.
- Nunberg, G. 2009. *Google Books: A Metadata Train Wreck* [Online]. Available: <https://languagelog.ldc.upenn.edu/nll/?p=1701> [Accessed on 23 July 2019].
- Nunberg, G. 2010. *Counting on Google Books* [Online]. Available: <https://www.chronicle.com/article/Counting-on-Google-Books/125735> [Accessed on 23 July 2019].
- O'Grady, W. 2010. *Contemporary linguistics: An introduction*, 6th ed., Boston, Massachusetts: Bedford/St. Martins.
- Ophir, S. 2016. Big data for the humanities using Google Ngrams: Discovering hidden patterns of conceptual trends. *First Monday*, 21(7).
- Organisciak, P., Capitanu, B., Underwood, T. & Downie, J.S. 2017. Access to billions of pages for large-scale text analysis. *iConference 2017*. Wuhan, China.
- Palmer, D. 2010. Text Preprocessing. In: Indurjha, N. & Damerau, F. J. (eds.) *Handbook of Natural Language Processing*. 2nd ed. Boca Raton, FL: CRC Press: 9–30.
- Park, J. & Brenza, A. 2015. Evaluation of semi-automatic metadata generation tools: A survey of the current state of the art. *Information technology and libraries*, 34(3): 22–42.
- Parry, M. 2010. *The Humanities Go Google* [Online]. Available: <https://www.chronicle.com/article/The-Humanities-Go-Google/65713> [Accessed on 23 July 2019].
- Pechenick, E.A., Danforth, C.M. & Dodds, P.S. 2015. Characterizing the Google Books corpus: Strong limits to inferences of socio-cultural and linguistic evolution. *PLOS One*, 10(10): e0137041.
- Perseus Digital Library. n.d.-a. *Greek and Roman Documents* [Online]. Available: <http://www.perseus.tufts.edu/hopper/collection?collection=Perseus:collection:Gr-eco-Roman> [Accessed on 18 September 2018].

- Perseus Digital Library. n.d.-b. *Perseus Digital Library – About* [Online]. Available: <http://www.perseus.tufts.edu/hopper/about> [Accessed on 12 June 2018].
- Petrov, S., Das, D. & McDonald, R. 2012. A Universal Part-of-Speech Tagset. In: Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J. & Piperidis, S., (eds.) *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey, 21-27 May 2012*: 2089–2096. European Language Resources Association.
- Pickard, A.J. 2017. *Research Methods in Information*, London: Facet Publishing.
- Pinto, A., Gonçalo Oliveira, H. & Oliveira Alves, A. 2016. Comparing the performance of different NLP toolkits in formal and social media text. *Proceedings of the 5th Symposium on Languages, Applications and Technologies (SLATE'16), Maribor, Slovenia, 20-21 June 2016*: 3:1–3:16. Schloss Dagstuhl.
- Pomerantz, J. 2015. *Metadata*, Cambridge: MIT Press.
- Pouliquen, B., Steinberger, R. & Best, C. 2007. Automatic detection of quotations in multilingual news. *Recent Advances in Natural Language Processing (RANLP), Borovets, Bulgaria, 27-29 September 2007*.
- Preece, J., Rogers, Y. & Sharp, H. 2011. *Interaction Design. Beyond human-computer interaction*, 3rd ed., Chichester: John Wiley & Sons.
- Project Gutenberg. n.d. *Project Gutenberg* [Online]. Available: <https://www.gutenberg.org> [Accessed on 20 August 2020].
- Pustejovsky, J. & Stubbs, A. 2012. *Natural Language Annotation for Machine Learning*, Sebastopol, California: O'Reilly Media.
- Qin, W., Elanwar, R. & Betke, M. 2018. LABA: Logical Layout Analysis of Book Page Images in Arabic Using Multiple Support Vector Machines. *Proceedings of the 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), London, United Kingdom, 12-14 March 2018*: 35–40. IEEE.
- Raganato, A., Camacho-Collados, J. & Navigli, R. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, April 2017*: 99–110. Association for Computational Linguistics.
- Rahman, M.M. & Finin, T. 2017. Deep Understanding of a Document's Structure. *Proceedings of the 4th IEEE/ACM International Conference on Big Data*

Computing, Applications and Technologies, Austin, Texas, USA, December 2017: 63–73. Association for Computing Machinery.

- Read, J., Dridan, R., Oepen, S. & Solberg, L.J. 2012. Sentence boundary detection: A long solved problem? *Proceedings of COLING 2012, Mumbai, India, December 2012*: 985–994. The COLING 2012 Organizing Committee.
- Reason, P. & Bradbury, H. 2006. *Handbook of action research: The concise paperback edition*, London: SAGE.
- Renear, A.H. 2004. Text Encoding. In: Schreibman, S., Siemens, R. & Unsworth, J. (eds.) *A Companion to Digital Humanities*. Oxford: Blackwell.
- Riley, J. 2017. *Understanding Metadata: What is Metadata, and What is it For?: A Primer* [Online]. Available: <https://www.niso.org/publications/understanding-metadata-2017> [Accessed on 24 January 2019].
- Roller, R., Uszkoreit, H., Xu, F., Seiffe, L., Mikhailov, M., Staeck, O., Budde, K., Halleck, F. & Schmidt, D. 2016. A fine-grained corpus annotation schema of German nephrology records. *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP), Osaka, Japan, December 2016*: 69–77. The COLING 2016 Organizing Committee.
- Rydberg-Cox, J.A., Chavez, R.F., Smith, D.A., Mahoney, A. & Crane, G.R. 2000. Knowledge Management in the Perseus Digital Library. *Ariadne*, 25.
- Salkind, N.J. 2010. *Encyclopedia of research design*, Thousand Oaks, CA: SAGE Publications, Inc.
- Schmidt, D. 2012. The role of markup in the digital humanities. *Historical Social Research*, 37(3): 123-146.
- Schuster, S. & Manning, C.D. 2016. Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. *10th International Conference on Language Resources and Evaluation Conference (LREC 2016), Portorož, Slovenia, 23-28 May 2016*: 2371–2378. European Language Resources Association.
- Shneiderman, B. & Plaisant, C. 2010. *Designing the user interface. Strategies of effective human-computer interaction*, 5th ed., Boston: Pearson Education.
- Shwartz, V. 2016. *Linguistic Analysis of Texts* [Online]. Available: <https://veredshwartz.blogspot.co.za/2016/06/linguistic-analysis-of-texts.html> [Accessed on 9 April 2018].

- Sicilia, M. 2014. Metadata Research: Making Digital Resources Useful Again? In: Sicilia, M. (ed.) *Handbook Of Metadata, Semantics And Ontologies*. Singapore: World Scientific Publishing.
- Sinclair, S. & Rockwell, G. 2016. *Voyant Tools* [Online]. Available: <http://voyant-tools.org/> [Accessed on 18 April 2019].
- Sketch Engine. n.d. *Sketch Engine* [Online]. Available: <https://www.sketchengine.eu> [Accessed on 22 August 2018].
- SourceForge.net. n.d. *TXM* [Online]. Available: <https://sourceforge.net/projects/txm/files/corpora/leviathan/> [Accessed on 17 September 2018].
- South Africa. 1978. *Copyright Act 98 of 1978* [Online]. Available: https://www.gov.za/sites/default/files/gcis_document/201504/act-98-1978.pdf [Accessed on 26 October 2020].
- spaCy – Annotation Specifications. n.d. *Annotation Specifications* [Online]. Available: <https://spacy.io/api/annotation> [Accessed on 19 October 2020].
- spaCy. n.d. *Industrial-Strength Natural Language Processing in Python* [Online]. Available: <https://spacy.io/> [Accessed on 25 August 2020].
- Stanford Parser. n.d. *Software > Stanford Parser* [Online]. Available: <https://nlp.stanford.edu/software/lex-parser.shtml> [Accessed on 19 October 2020].
- Suranto, B. 2015. Software prototypes: Enhancing the quality of requirements engineering process. *Proceedings of the 2015 International Symposium on Technology Management and Emerging Technologies (ISTMET), Langkawai Island, Kedah, Malaysia, 25-27 August 2015*: 148–153. IEEE.
- Svensson, P. 2013. Humanities Computing as Digital Humanities. In: Terras, M., Nyhan, J. & Vanhoutte, E. (eds.) *Defining Digital Humanities: A Reader*. Surrey, England: Ashgate: 159–186.
- Tan, L. 2014. *Pywsd: Python Implementations of Word Sense Disambiguation (WSD) Technologies [software]* [Online]. Available: <https://github.com/alvations/pywsd> [Accessed on 30 March 2020].
- Taylor, J.R. 2003. Near synonyms as co-extensive categories: ‘high’ and ‘tall’ revisited. *Language Sciences*, 25(3): 263–284.
- TCP. n.d. *Text Creation Partnership: EEBO, ECCO and Evans texts* [Online]. Available: <https://ota.ox.ac.uk/tcp/> [Accessed on 15 August 2018].

- TEI – News. 2020. *TEI Guidelines – Version 4.0.0* [Online]. Available: <http://www.tei-c.org/News/> [Accessed on 17 August 2020].
- TEI – Projects Using the TEI. 2017. *Projects Using the TEI*. [Online]. Available: <http://www.tei-c.org/Activities/Projects> [Accessed on 12 January 2018].
- TEI – Text Encoding Initiative. 2016. *TEI: Text Encoding Initiative – Home*. [Online]. Available: <http://www.tei-c.org/index.xml> [Accessed on 12 January 2018].
- TEI – Text Encoding Initiative. 2020. *P5: Guidelines for Electronic Text Encoding and Interchange* [Online]. Available: <https://tei-c.org/release/doc/tei-p5-doc/en/html/index.html> [Accessed on 24 January 2021].
- TEI – Text Encoding Initiative. n.d. *TEI – Getting Started with P5 ODDs* [Online]. Available: <http://www.tei-c.org/guidelines/customization/getting-started-with-p5-odds/> [Accessed on 9 January 2019].
- TEI wiki – TXM. 2016. *TXM* [Online]. Available: <https://wiki.tei-c.org/index.php/TXM> [Accessed on 26 June 2018].
- TEI wiki – XAIRA. 2007. *XAIRA* [Online]. Available: <https://wiki.tei-c.org/index.php/Xaira> [Accessed on 26 June 2018].
- Terras, M. 2016. A Decade in Digital Humanities. *Journal of Siberian Federal University, Humanities & Social Sciences*, 9(7): 1637–1650.
- Terras, M., Baker, J., Hetherington, J., Beavan, D., Zaltz Austwick, M., Welsh, A., O'Neill, H., Finley, W., Duke-Williams, O. & Farquhar, A. 2017. Enabling complex analysis of large-scale digital collections: Humanities research, high-performance computing, and transforming access to British Library digital collections. *Digital Scholarship in the Humanities*, 33(2): 456–466.
- Terras, M., Nyhan, J. & Vanhoutte, E. 2013. Introduction. In: Terras, M., Nyhan, J. & Vanhoutte, E. (eds.) *Defining Digital Humanities - A Reader*. Surrey, England: Ashgate: 1–12.
- Text Creation Partnership. 2019. *EEBO-TCP: Early English Books Online* [Online]. Available: <https://textcreationpartnership.org/tcp-texts/eebo-tcp-early-english-books-online/> [Accessed on 24 April 2019].
- The Global WordNet Organization. n.d. *Wordnets in the World* [Online]. Available: <http://globalwordnet.org/wordnets-in-the-world/> [Accessed on 24 August 2018].
- The iSchool at Illinois. 2014. *Exploring the Billions and Billions of Words in the HathiTrust Corpus with Bookworm: HathiTrust + Bookworm Project* [Online]. Available: <https://ischool.illinois.edu/research/projects/hathitrust-bookworm-project> [Accessed on 11 January 2018].

- The Library of Congress. 2017. *MARC Genre Term List* [Online]. Available: <https://www.loc.gov/standards/valuelist/marcqt.html> [Accessed on 17 May 2019].
- The Walt Whitman Archive. n.d. *Do you know what music does*. [Online]. Available: <https://whitmanarchive.org/manuscripts/transcriptions/tex.00088.html> [Accessed on 27 February 2018].
- Thomas, J. 2007. *Project Gutenberg Digital Library Seeks To Spur Literacy* [Online]. Available: <https://japan2.usembassy.gov/e/p/2007/tp-20070723-89.html> [Accessed on 18 September 2018].
- TREC. 2020. *Text REtrieval Conference (TREC)* [Online]. Available: <https://trec.nist.gov/> [Accessed on 25 January 2021].
- Trumpener, K. 2009. Critical response I. Paratext and genre system: A response to Franco Moretti. *Critical Inquiry*, 36(1): 159–171.
- Tucker, N. 2019. *Branch Rickey Crowdsourcing Project: It's Outta Here!* [Online]. Available: <https://blogs.loc.gov/loc/2019/03/branch-rickey-crowdsourcing-project-its-outta-here/?loclr=ealocb> [Accessed on 8 April 2019].
- TXM User Manual. 2018. *TXM User Manual* [Online]. Available: <http://textometrie.ens-lyon.fr/files/documentation/TXM%20Manual%200.7.pdf> [Accessed on 26 June 2018].
- UCLA Research Guide. n.d. *Text Encoding (TEI)* [Online]. Available: <http://guides.library.ucla.edu/tei> [Accessed on 12 January 2018].
- Underwood, T. 2015a. *A dataset for distant-reading literature in English, 1700-1922*. [Online]. Available: <https://tedunderwood.com/2015/08/07/a-dataset-for-distant-reading-literature-in-english-1700-1922/> [Accessed on 23 July 2019].
- Underwood, T. 2015b. *Understanding Genre in a Collection of a Million Volumes* [Online]. University of Illinois, Urbana-Champaign. Available: <https://hcommons.org/deposits/item/hc:12277/> [Accessed on 30 July 2019].
- Underwood, T., Bamman, D. & Lee, S. 2018. The Transformation of Gender in English-Language Fiction. *Journal of Cultural Analytics*, 1(1).
- Universal Dependencies – v1. n.d. *Universal Dependencies* [Online]. Available: <https://universaldependencies.org/docsv1/u/dep/index.html> [Accessed on 19 October 2020].
- Universal Dependencies. n.d. *Universal Dependencies* [Online]. Available: <http://universaldependencies.org> [Accessed on 23 August 2018].

- University of Oxford IT Services. 2015a. *British National Corpus* [Online]. Available: <http://www.natcorp.ox.ac.uk/> [Accessed on 26 June 2018].
- University of Oxford IT Services. 2015b. *Using BNC with Xaira* [Online]. Available: <http://www.natcorp.ox.ac.uk/tools/> [Accessed on 26 June 2018].
- University of Oxford IT Services. n.d. *Using Xaira under Windows* [Online]. Available: <http://projects.oucs.ox.ac.uk/xaira/Doc/refman.xml?ID=X01> [Accessed on 26 June 2018].
- Ustalov, D., Teslenko, D., Panchenko, A., Chernoskutov, M., Biemann, C. & Ponzetto, S.P. An unsupervised word sense disambiguation system for under-resourced languages. *11th International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 2018*. European Language Resources Association (ELRA).
- Vaknin, S. 2005. *The Ubiquitous Project Gutenberg: Interview with Michael Hart, Its Founder* [Online]. Available: <http://samvak.tripod.com/busiweb46.html> [Accessed on 18 September 2018].
- Van den Branden, R., Terras, M. & Vanhoutte, E. 2017. *TEI By Example* [Online]. Available: <http://teibyexample.org/modules/TBED00v00.htm> [Accessed on 12 January 2018].
- Vanhoutte, E. 2013. The Gates of Hell: History and Definition of Digital | Humanities | Computing. In: Terras, M., Nyhan, J. & Vanhoutte, E. (eds.) *Defining Digital Humanities: A Reader*. Surrey, England: Ashgate: 119–156.
- Viiri, S. 2014. *Digital Humanities and Future Archives* [Online]. London, United Kingdom: Finnish Institute in London. Available: https://www.fininst.uk/wp-content/uploads/2017/09/Digital_Humanities_and_Future_Archives.pdf [Accessed on 29 September 2020].
- Walker, M., Takayama, L. & Landay, J.A. 2002. High-fidelity or low-fidelity, paper or computer? Choosing attributes when testing web prototypes. *Proceedings of the human factors and ergonomics society annual meeting*, 46(5): 661–665.
- Welsh, M.E. 2014. Review of Voyant tools. *Collaborative Librarianship*, 6(2): 96–98.
- Wilson, M. 2011. Interfaces for information retrieval. In: Ruthven, I. & Kelly, D. (eds.) *Interactive Information Seeking, Behaviour and Retrieval*. London: Facet Publishing: 139–170.
- Wissler, L., Almashraee, M., Díaz, D.M. & Paschke, A. 2014. The Gold Standard in Corpus Annotation. *IEEE GSC 2014*. Pasau, Germany.

- WordNet. n.d. *WordNet – A Lexical Database for English* [Online]. Available: <https://wordnet.princeton.edu/> [Accessed on 24 August 2018].
- Wynne, M. 2014. *Changes to the distribution of the British National Corpus* [Online]. Available: <http://blogs.it.ox.ac.uk/martinw/2014/01/13/changes-to-the-distribution-of-the-british-national-corpus/> [Accessed on 26 June 2018].
- Xiao, R. 2010. Corpus Creation. In: Indurjha, N. & Damerau, F. J. (eds.) *Handbook of Natural Language Processing*. 2nd ed. Boca Raton, FL: CRC Press: 147–166.
- Yang, X., Yumer, E., Asente, P., Kralej, M., Kifer, D. & Giles, L.C. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, USA, 21-26 July 2017*: 5315–5324. IEEE.
- Yin, R.K. 2009. *Case study research: Design and methods*, 4th edition, Los Angeles, California: Sage Publications.
- Yuan, D., Richardson, J., Doherty, R., Evans, C. & Altendorf, E. 2016. Semi-supervised word sense disambiguation with neural models. *Proceedings of COLING 2016, Osaka, Japan, 2016*: 1374–1385. The COLING 2016 Organizing Committee.
- Zeng, M. & Qin, J. 2016. *Metadata*, 2nd edition, Chicago: Neal-Schuman.
- Zeyrek, D., Mendes, A., Grishina, Y., Kurfalı, M., Gibbon, S. & Ogrodniczuk, M. 2019. TED Multilingual Discourse Bank (TED-MDB): A parallel corpus annotated in the PDTB style. *Language Resources and Evaluation*, 54: 587–613.
- Zhang, A.B. & Gourley, D. 2008. *Creating Digital Collections: A practical guide*, Oxford: Chandos Publishing.
- Zhong, Z. & Ng, H.T. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. *Proceedings of the ACL 2010 system demonstrations, Uppsala, Sweden, July 2010*: 78–83. Association for Computational Linguistics.
- Zhou, X. & Marslen-Wilson, W. 2000. Lexical representation of compound words: Cross-linguistic evidence. *Psychologia*, 43(1): 47–66.