

Corpus Linguistics Software:
understanding their usages and
delivering two new tools



Andressa Rodrigues Gomide

Thesis submitted for the degree of Doctor of Philosophy

August 2020

Department of Linguistics and English Language

Para Tia Elenice

Declaration

This thesis has not been submitted in support of an application for another degree at this or any other university. It is the result of my own work and includes nothing that is the outcome of work done in collaboration except. Many of the ideas in this thesis were the product of discussion with my supervisor Dr. Andrew Hardie.

Abstract

The increasing availability of computers to ordinary users in the last few decades has led to an exponential increase in the use of Corpus Linguistics (CL) methodologies. The people exploring this data come from a variety of backgrounds and, in many cases, are not proficient corpus linguists. Despite the ongoing development of new tools, there is still an immense gap between what CL can offer and what is currently being done by researchers. This study has two outcomes. It (a) identifies the gap between potential and actual uses of CL methods and tools, and (b) enhances the usability of CL software and complement statistical application through the use of data visualization and user-friendly interfaces. The first outcome is achieved through (i) an investigation of how CL methods are reported in academic publications; (ii) a systematic observation of users of CL software as they engage in the routine tasks; and (iii) a review of four well-established pieces of software used for corpus exploration. Based on the findings, two new statistical tools for CL studies with high usability were developed and implemented on to an existing system, CQPweb. The Advanced Dispersion tool allows users to graphically explore how queries are distributed in a corpus, which makes it easier for users to understand the concept of dispersion. The tool also provides accurate dispersion measures. The Parlink Tool was designed having as its primary target audience beginners with interest in translations studies and second language education. The tool's primary function is to make it easier for users to see possible translations for corpus queries in the parallel concordances, without the need to use external resources, such as translation memories.

Acknowledgements

I would like to thank everyone who somehow helped me throughout my PhD.

Special thanks go to:

CAPES, for financial support

CASS and LAEL, for the academic support

Andrew, for helping me become a stronger person

Elena, without whom I would not have completed my PhD

Cláudia, who helps me find serenity even in the hardest moments

Deise, an educator for all time, for her constant guidance and inspiration

My mother, who taught me, with lots of love, how to be independent

My father, who teaches me how to see good in everything

Fernanda, my best friend and safe harbour

Fred, “é tão 10, que junto todo o stress é miúdo”

Adriana, Bruno, Giulia, my (non) quixotic friends

Tanjan, Pri, Mat and Isolde, who brought joy to work

Ana Flávia, Clarissa, Gubs, Laís, Lorena, friends for a lifetime

Cris, Camila, Fernando, Erika, Pedro, a bit of Brazilian warmth in the British winter

Angela, mia musicalità

Contents

1 INTRODUCTION.....	1
1.1 Background	1
1.2 Research aims	4
1.3 Thesis structure	5
2 LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Language and computers: a historical view	9
2.2.1 <i>Ancient times</i>	9
2.2.2 <i>The Middle and Modern Ages</i>	11
2.2.3 <i>Twentieth Century</i>	11
2.3 Digital Computing and Language	18
2.3.1 <i>Digital Humanities</i>	18
2.4 Corpus Linguistics	20
2.4.1 <i>Brown Corpus</i>	20
2.4.2 <i>Four generations of tools</i>	21
2.4.3 <i>Implications for Linguistic Theory Considerations</i>	27
2.4.4 <i>Annotation, textual mark-up, and encoding</i>	30
2.4.5 <i>Consistency and tools</i>	32
2.5 Discussion	34
3 AN EXEMPLAR-BASED REVIEW OF CORPUS ANALYSIS SOFTWARE	
TOOLS.....	36
3.1 Introduction	36
3.2 Framework for the software review	36
3.2.1 <i>The criteria</i>	37
3.3 Selection of tools for review	43
3.4 The tools.....	45
3.4.1 <i>Standalone tools: AntConc and #LancsBox</i>	45
3.4.2 <i>Quanteda and other script-based tools</i>	49
3.4.3 <i>CQPweb</i>	54
3.5 Discussion and conclusion	59
4 CORPUS-BASED STUDIES: A LITERATURE INVESTIGATION.....	62
4.1 Introduction	62
4.2 The databases	64
4.2.1 <i>Choosing the sources</i>	64
4.2.2 <i>LLBA</i>	66
4.2.3 <i>AHD</i>	67
4.3 Procedures	68
4.3.1 <i>Article retrieval and processing</i>	68
4.3.2 <i>Investigation Methods</i>	69
4.4 Analysis.....	79

4.4.1 Difficulties in finding mentions of the tools	79
4.4.2 Tool cost and availability.....	82
4.4.3 More than concordance lines.....	83
4.4.4 Complementing CL methods	84
4.5 Discussion	85
5 TARGET AUDIENCE: CONTEXTUAL DESIGN AND USABILITY	86
5.1 Introduction	86
5.2 Overview	86
5.3 The Contextual Design Approach.....	87
5.3.1 Contextual design.....	87
5.3.2 Contextual inquiry	88
5.3.3 Contextual analysis and user needs and requirements extraction.....	93
5.3.4 Design-informing models.....	96
5.3.5 Tool development: iteration with users.....	97
5.4 Other types of user observations	98
5.4.1 Workshops, lectures and seminars.....	98
5.4.2 Talks.....	99
5.4.3 Web analytics	99
5.5 User needs and requirements	101
5.5.1 Some caveats.....	102
5.6 Summary	103
6 ADVANCED DISPERSION	104
6.1 Dispersion: definition, measures, applications	104
6.2 Graphical visualization of dispersion.....	114
6.2.1 AntConc: Concordance Plot.....	116
6.2.2 WordSmith Tools: Text Plot.....	118
6.2.3 Quanteda: Textplot x-ray.....	120
6.2.4 CLiC: concordance plot.....	122
6.2.5 Voyant Tools: corpus terms and trends	123
6.3 Criteria for a new visualization	126
6.3.1 Usability.....	126
6.3.2 Functionality.....	129
6.3.3 Implementation	130
6.4 Prototypes.....	131
6.4.1 Mock corpus.....	132
6.4.2 Prototype one: parallel coordinates	132
6.4.3 Prototype two: histogram and scatterplot	135
6.4.4 Prototype three: time-series style	138
6.4.5 Prototype four: scatter-plot and barcode style.....	139
6.5 Implementation of the visualization.....	141
6.5.1 Dispersion Overview: text frequency.....	142
6.5.2 New query and new action menu	144
6.5.3 Single-text dispersion.....	145
6.6 Conclusion: observations and next steps	146
7 PARLINK: A TOOL FOR PARALLEL CORPORA.....	147

7.1 Introduction	147
7.2 Parallel Corpus Linguistics: a literature review	148
7.2.1 <i>Key Terms</i>	148
7.2.2 <i>Current Methods</i>	151
7.2.3 <i>Prominent tools and data</i>	154
7.3 Limitations and Motivation.....	165
7.3.1 <i>Indirect use of parallel corpora</i>	166
7.3.2 <i>CL tools and methods</i>	168
7.3.3 <i>A new solution to the users</i>	170
7.4 The tool	171
7.4.1 <i>Conceptualization of the tool</i>	171
7.4.2 <i>Parallel Link</i>	174
7.4.3 <i>Creation Process</i>	174
7.4.4 <i>The tool development</i>	183
7.5 Conclusion	188
8 CONCLUSION	190
8.1 Thesis Summary.....	190
8.2 Answers to the research questions	191
8.3 Significance and Contribution	192
8.3.1 <i>Theoretical contribution</i>	192
8.3.2 <i>Practical contribution</i>	193
8.4 Limitations	193
8.5 Future work.....	194
8.6 Concluding remarks	195
REFERENCES.....	196

List of Tables

Table 4.1: tools from existing lists found in the corpus.....	72
Table 4.2: CL tools found in the corpus and their categories	74
Table 4.3: CL-related software found in the corpus	75
Table 4.4: absolute frequency of CL software mentioned across fields	77
Table 5.1: user classes.....	90
Table 5.2: profiles of the five participants in the contextual inquiry.....	91
Table 5.3: possible questions for use in the contextual inquiry	93
Table 5.4: Top 40 queries at CQPweb Lancaster.....	100

List of Figures

Figure 2.1: screenshot of a concordance in BYU corpora	29
Figure 2.2: screenshot of a concordance in CQPweb	29
Figure 2.3: screenshot of a concordance in Sketch Engine.....	29
Figure 3.1: example of search breadcrumbs	42
Figure 3.2: example of word cloud made with Quanteda	51
Figure 3.3: example of a frequency pot made with Quanteda	52
Figure 3.4: example of word keyness plot made with Quanteda	52
Figure 4.1: overall frequency of CL software.....	74
Figure 4.2: percentage of software type (pricing) across fields.....	77
Figure 4.3: percentage of software type (environment) across fields	78
Figure 4.4: percentage of software type (functionality) across fields.....	78
Figure 5.1: requirement statement format.....	94
Figure 5.2: affinity diagram for the interviews/observations.....	95
Figure 5.3: selected and primary personas.....	96
Figure 6.1: relative frequency for ‘six’ for individual texts in the BNC 2014	109
Figure 6.2: screenshot of a concordance plot from AntConc	117
Figure 6.3: screenshot of Text Plot, in WordSmith Tools	119

Figure 6.4: screenshot of Word Positions, in WordSmith Tools	119
Figure 6.5: screenshot of Uniform View, in WordSmith Tools.....	120
Figure 6.6: Text Plot x-ray, in Quanteda (relative scale).....	121
Figure 6.7: Text Plot x-ray for two words, in Quanteda (absolute scale).....	122
Figure 6.8: screenshot of Concordance Plot in CLiC	123
Figure 6.9: screenshot of Corpus Terms, in Voyant Tools	124
Figure 6.10: screenshot of Trends, in Voyant Tools.....	125
Figure 6.11: prototype one - parallel coordinates	134
Figure 6.12: prototype two - histogram and scatterplot.....	137
Figure 6.13: Prototype three - time series style	139
Figure 6.14: prototype four - scatter plot and barcode style	141
Figure 6.15: screenshot of dispersion - step 1.....	142
Figure 6.16: screenshot of dispersion overview	143
Figure 6.17: dispersion - tool multiple queries	144
Figure 6.18: dispersion - text view function	145
Figure 6.19: dispersion - text view with bars.....	145
Figure 7.1: concordance lines for parallel corpora using CQPweb	155
Figure 7.2: multiple languages concordance lines (OPUS).....	156

Figure 7.3: a screenshot of OPUS word alignment database tool.....	156
Figure 7.4: ACTRES query composer	157
Figure 7.5: ACTRES query results	157
Figure 7.6: EPTIC corpora on NoSketchEngine.....	158
Figure 7.7: a screenshot of InterCorp in KonText	159
Figure 7.8: a screenshot of Treq.....	160
Figure 7.9: a screenshot of PaGeS	162
Figure 7.10: a screenshot of Multilingwis	162
Figure 7.11: a screenshot of CLUVI.....	163
Figure 7.12: screenshot of Linguee.....	164
Figure 7.13: screenshot of Reverso Context	165
Figure 7.14: possible Portuguese translations for phrases with the verb to take	167
Figure 7.15: example of comparable zones	173
Figure 7.16: Agile Process	175
Figure 7.17: Bars & Dots Prototype	177
Figure 7.18: Arrows and Bow prototype	178
Figure 7.19: Heatmap + Sidebar	179
Figure 7.20: Mountain View Prototype	181

Figure 7.21: Sum up Flower prototype	182
Figure 7.22: screenshot of the parlink tool	187
Figure 7.23: screenshot of the pop-up table for parlink.....	189

List of Abbreviations and Acronyms

RAF	Adjusted Frequency
AFP	Adjusted Frequency Based on Corpus Parts
AFD	Adjusted Frequency Based on Distance
AD	Affinity Diagram
AZ	Aligned Zones
AQL	ANNIS Query Language
AHD	Arts & Humanities Database
ALD	Average Logarithmic Distance
APA	Application Programming Interface
ARF	Average Reduced Frequency
AWT	Average Waiting Time
BE06	British English 2006
BNC	British National Corpus
COGS	Concordance Generating System
CLAWS	Constituent-Likelihood Word-Tagging System
CL	Corpus Linguistics
COCA	Corpus of Contemporary American English
CQL	Corpus Query Language
CQP	Corpus Query Processor
CWB	Corpus Workbench
CBTS	Corpus-Based Translation Studies
D3.js	Data-Driven Documents Javascript Library
DP	Deviation of Proportion
DH	Digital Humanities
DOI	Digital Object Identifier
ISO	International Organization for Standardization
KWIC	Key Word in Context
LLBA	Linguistics and Language Behavior Abstracts
NLP	Natural Language Processing
NSUs	Non-Specialist Users of Corpus Data and Methods
OCP	Oxford Concordance Program
OED	Oxford English Dictionary
POS	Parts-Of-Speech
RS	Requirement Statement
SC	Source Corpus
SEU	Survey of English Use
TC	Target Corpus
TACT	Text Analysis Computing Tools
TEI	Text Encoding Initiative
PaGeS	The Parallel Corpus German/Spanish
TLL	<i>Thesaurus Linguae Latinae</i>

TML	Translation Markup Language
TM	Translation Memories
TTR	Type/Token Ratio
USAS	Ucrel Semantic Analysis System
UX	User Experience
wpm	Words Per Million

1 Introduction

1.1 Background

As a discipline, Corpus Linguistics (CL) has long dealt with the principles and practice of using machine-readable textual data for different research purposes since the origin of the field in its modern form in the early 1960s. CL has played a major role in areas such as lexicography (Halliday et al. 2004), grammar (Biber 1999), language teaching (Aijmer 2009), and discourse analysis (McEnery & Baker 2015). The increasing availability of computers to ordinary users in the last three to four decades has led to a massive increase in the use of CL methodologies by linguists of all kinds. Moreover, CL has now been utilised in a wide variety of fields beyond linguistics, such as psychology, information science and law (see Šarčević 2016; Bowker 2018; Maia & Santos 2018). Thus, it has proven to be a useful methodology for a wide variety of disciplines, promoting inter- and multi-disciplinary research.

The use of CL specifically in the field of language teaching has also been steadily growing since Tim Johns' initial work on *data-driven learning* (Johns 1986, 1991, 1994). In this approach, learners of a second or foreign language work with corpus data, performing corpus analysis with or without the aid of their teachers, as part of

the process of becoming proficient in the language they are learning, the *target language*. Learners thus become “more active participants in the learning process” (Tognini-Bonelli 2001:43), in charge of their own investigations and learning in an autonomous way (Bernardini 2002). However, direct use of corpus data and software by students and teachers is not the only way that CL is used in language teaching. Corpora can also be employed indirectly – for example, to design materials, syllabi and class activities; to describe the linguistic characteristics of different learner groups; or to identify important characteristics of the target language (Granger 2002; Gabrielatos 2005).

From these observations on CL’s interdisciplinarity and application in language teaching, it follows that – whatever way corpus data is approached, and for whatever purpose – the people exploring this data come from a variety of backgrounds and, in many cases, are not proficient corpus linguists. They may, for instance, be language teachers or students, or researchers in areas other than linguistics, rather than trained corpus researchers. Although a very heterogeneous group, these non-specialist users of corpus data share some or all the following characteristics. They are not CL experts, and hence are not likely to (wish to) invest a great amount of time into learning how to use a *tool* – any computer application or subsystem thereof used to achieve a specific goal – just to access corpus data that they wish to use. They are likely to be more concerned with language in use than with the details of statistical analyses that corpus linguistic research often involves. Their investigations typically do not require or utilise techniques specially designed for their purposes; rather they are likely to stick to what we might call *off-the-shelf* methods. Their work is usually collaborative, with the same corpus being used by multiple people. The corpus data is, ultimately, no more than a means for such users to achieve some aim that may well

not have to do with CL at all. More practically, such users are also likely to have limited or even no awareness of the full range of currently available corpus data resources (Gries 2015; Diaz-Negrillo et al. 2013); in addition, they are likely to experience great difficulties in using CL software tools, as indeed previous research has found many such users to report (Boulton 2012).

Unlike commercial software, which is developed by programmers whose primary professional concern is the design and implementation of software, CL tools are created by linguists, and implement or reflect design decisions made solely on the basis of developer preference. This creates the potential for idiosyncrasy, which can negatively affect the final user experience. The resulting software may not be intuitive or may not address the real needs of the full range of potential users. For example, despite having been available for almost 20 years (Anthony 2002), and despite its status as one of the most widely used CL tools at the present time, AntConc (Anthony 2019) still exhibits room for improvement (see 3.4.1). There has been a growing number of attempts to develop tools with greater usability. For instance, #LancsBox (Brezina et al. 2015) aims to improve understanding of collocations by allowing the user to explore them through visualization as a graph of interlinked nodes. Besides visualizations, improvements in user interfaces have also been observed. The web-based corpus search engine Kontext (Machálek 2020), for example, presents a clear and easy to navigate interface with features which enhance the user experience with the tool.

Despite the effort to enhance user experience observed in emerging tools like #LancsBox and Kontext there are still shortcomings that need addressing. As some researchers have pointed, there is excess of bias found in corpus-based research

(especially with data cherry-picking), inaccuracy of statistical application, and limited use or even unawareness of currently available resources (Gries 2015; Diaz-Negrillo et al. 2013). Moreover, there are many researchers who do not use CL methodologies due to difficulties with the tools (Boulton 2012). There is, then, still an immense gap between what CL can offer and what is currently being done by the researchers mentioned above.

1.2 Research aims

This study has two main goals:

1. To identify how CL methods and tools usage is reported in the literature and how users deal with these tools.
2. To demonstrate how to enhance usability of CL software and complement statistical application through the use of data visualization and user-friendly interfaces.

My aim with this thesis is to contribute with the following: (a) to reveal the gap between potential and actual uses of CL methods; (b) to deliver two new statistical tools for CL studies with high usability.

To achieve the first aim, we need to understand the users. The first step is then to define the targeted user group or groups. This is because different groups have different needs, and one effort cannot address all. In this study, the main target user group is beginner users of CL in the field of language teaching and learning, as introduced in section 1.1. This includes language students and teachers who use computer-assisted techniques; language learning materials designers who use corpora

to inform their work; and academics conducting research on language teaching who are not CL specialists.

From this last group, a secondary target user group emerges. In many universities, language, linguistics and literature courses are offered within the same department, and it is common to see undergraduate students taking modules in different language-related disciplines. For instance, a student majoring in Literature might be introduced to CL while taking a module related to language teaching. Such potential CL users form the secondary target user group. The motivation for adding this second user group is the potential that CL offers for an audience that are not strictly connected to linguistics but might benefit from CL tools. The two groups together are referred to in this thesis as *non-specialist users of corpus data and methods* (NSUs).

1.3 Thesis structure

With the target user group defined as consisting of NSUs, the next step is to understand the user group's needs. My thesis addresses this issue in two ways. First, I undertake an investigation of how the use of CL methods is reported in the literature in Social Science and Humanities publications. This literature analysis investigates what tools are used, what kind of analyses are performed, and how they differ according to the disciplinary area. A dataset of over 4,000 academic journal articles was retrieved from two different academic databases; and CL methods were used for the analysis. Second, I undertake a direct study of users' needs. I use a contextual design approach to observe researchers using corpus tools within their own routines and contexts. This systematic observation of users allows their needs and requirements to be identified.

To begin addressing the second goal, I needed to develop a more comprehensive picture of the presently available corpus tools, and their use in the literature. The first step for that was to understand the capabilities of presently popular corpus software in the light of the improved understanding of users and the tools they use that has previously been arrived at. I did this through a literature and performance review of four specific pieces of software. Based on this review, two new visualizations for CL are developed as extensions to an existing system, CQPweb (Hardie 2012). I opted for this approach for four main reasons. First, CQPweb is open-source, meaning that its underlying code is freely available and can be modified. It is web-based, which improves accessibility. It is currently used by researchers in many different countries¹, which makes it a relatively well-established piece of software. Finally, it has an interface that has already been well-received by many researchers, having been developed following the model of BNCweb (Hoffmann & Evert 2006), widely used to retrieve textual data from the British National Corpus 1994.

In the course of developing these new visualizations, I draw regularly on approaches from the area of (software) user-experience research – an important branch of contemporary computer science and also an important tool in the software industry. To date, and to my knowledge, there have been no studies in CL software development that draw on the user-experience methodologies employed in professional software design. Hartson and Pyla (2012) argue that when developing a

¹ There are many installations of CQPweb on public servers around the world, such as <https://corpling.uis.georgetown.edu/cqp/> (USA); <http://cqpw-prod.vip.sydney.edu.au/CQPweb/> (Australia); <https://coct.naer.edu.tw/cqpweb/> (Taiwan)

software product, it is crucial to understand the needs and characteristics of the users. They go on to observe that the target user group needs to be carefully studied and understood before the design itself takes place; moreover, a dialogue with current and/or potential users must continue through the whole process of software development. According to Hartson and Pyla, products designed with the support of these user-experience methods have been shown to require very little learning time and to attract a higher number of users, compared to programs which do not utilise that approach. This methodology aligns well with desired outcomes of this study: (a) NSUs will be able to spend less time learning how the software functions and more time on linguistic analysis; and (b) new users will not be put off by the software.

Developing two new tools and implementing them on CQPweb allowed me to explore the tool design strategy presented in this thesis to address the needs of the NSUs. Although each tool presents solutions to different issues, both provide developers of CL tools with concrete examples of how to identify users' needs and present solutions to them. In chapters six and seven I will also discuss some issues I faced when I was developing (writing scripts in PHP and JavaScript) both tools.

2 Literature Review

2.1 Introduction

This chapter presents a literature review of how technologies for research into language have affected, or been affected by, the investigations in which they have been utilized. First, I briefly discuss how the methods used to research language have evolved from ancient times through to the mid twentieth century (2.2). I then cover the advent of computer-based language studies, and how the use of new technologies helped scholars address the limitations of the techniques available to them prior to this time (2.3). Section 2.4 deals with the emergence of Corpus Linguistics (CL) and its methods and tools; I consider how computer technology came to be an indispensable asset in support of empirical research on language. I conclude this chapter with a discussion of the new directions presently being explored in the information technologies used in the various disciplinary approaches to computer-based language studies (2.5).

2.2 Language and computers: a historical view

2.2.1 Ancient times

Dealing with large masses of data is currently a concern across many different disciplines. This is probably due to the new ease with which scholars can access, gather and process data driven by the ongoing revolution in digital technology. However, interest in gathering and studying large amounts of texts goes back to ancient times. The reason for the present review to consider work very far back in the past is twofold. Ancient work may have brought up potentially fruitful ideas not at the time – or even yet – explored. Moreover, considering the distant past can shed some light on how and why the methods currently employed in analysing language came to be adopted. In sum, the goal of this part of the literature review is to identify, so far as possible, what questions scholars through the ages have asked when studying language through real data; which techniques have been used so far; and how the technologies available in each period have limited and or expanded research by the scholars of that period.

Pāṇini has undoubtedly played a massive role in the development of language studies. He is mainly known for his work on the *Aṣṭādhyāyī*, an early, yet complex grammar of Sanskrit (Thomas 2011:2-6). Pāṇini's grammar formal system has also greatly influenced on the design of computer programming languages (Bhate & Kak 1991). Despite recognising the importance of Pāṇini and other Eastern scholars, due to the scope of this work, I will concentrate this literature review on Western studies.

The origin of the western tradition of the study of language units is attributed to scholars in Ancient Greece. It was probably at that time that what are now referred to as parts-of-speech (POS) were identified and defined for the first time (Robins 2013).

By breaking language into countable segments, the scholars of that time were leading a path to a tangible view of language.

In terms of text production, as the text collections were getting bigger and bigger, the need to better organise archives of documents arose. The concern was how to collect, store and access text systematically. A classic example of this collection process is the Library of Alexandria, where enormous numbers of scrolls were stored and managed. The organisation of scrolls was not nearly as easy as the organisation of today's books. Because of the high costs of a single scroll, and the consequent undesirability of leaving any free space going to waste, one scroll would often contain more than a single work, which made a system to manage the different pieces of work necessary (Olesen-Bagneux 2014). It was during the era in which scrolls were used that measuring techniques such as stichometry (line numbers) and colometry (verses) emerged (Pawłowski 2008). Studies of mode, register, and stylistics, for instance, would not be possible if there was not a means to navigate documents according to such divisions. Today, the frequently used divisions are not verses and lines but rather elements such as paragraphs or sections. These metrics might seem trivial nowadays when these concepts are so established in our lives. However, choosing the measures and units to be used in a corpus analysis is still an area of difficulty in language research. For example, the definition of the word as a linguistic unit (see, e.g., Gries 2009:12) and the appropriate transcript methods for spoken corpora are still disputed. It has always been a struggle to define terms in a fine way so that its quantification can be as accurate as possible, considering the restraint of the technology available.

2.2.2 The Middle and Modern Ages

In the Middle Ages, a common method to facilitate access to the content of large collections of texts was the use of tables of contents and subject indexes, both of which are, of course, still in use today (Lerner 1999). Because indexes facilitate information retrieval, their implementation meant the readers could focus more on the text than on the searching process (Lerner 1999).

Better systems to organise and retrieve data also meant support for new collections. Several scholars adopted the concept of gathering texts for their specific purposes. Many of these projects had their final goal linked to lexicographical purposes as in the case of Samuel Johnson's dictionary, published in 1755. Over the course of nine years of work on this effort, Johnson wrote letters to many different recipients to collect actual examples of word usage to include in his *A Dictionary of the English Language* (Reddick & Johnson 1996).

A similar process was followed by James Murray (1837-1915), the lexicographer who was the first primary editor of the Oxford English Dictionary (OED). To compile the dictionary, Murray had to manually organise the massive quantity of letters that he would receive for his text collection (Murray 2001). These stories illustrate how laborious data collection and management were. Scholars like Murray would dedicate their lives to collecting and organising large textual amount, loosely associated to the modern concept of corpus, whose contents would, most likely, not be explored further than its lexicographical aspect.

2.2.3 Twentieth Century

In the twentieth century, more applications emerged for collections of language data beyond solely lexicographical description. Several researchers began to use corpora to

describe language. Thorndike (1921), for instance, identified the most frequent word types in English based on a corpus of 4.5 million words. The publication of this list triggered changes in language teaching, both first and second language, in the USA and Europe. Thorndike's study is the likely inspiration of the commonly-encountered approach of teaching learners the most frequently used words first (Graves 2016). An updated version of Thorndike's list, containing 30 thousand word types based on a larger corpus (18 million words), was published some twenty years later (Thorndike & Lorge 1944). Following the same trend, the *General Service List of English Words*, a publication which lists the two thousand most frequent words in English (West 1953), was created. This publication is probably one of the most famous lexicon descriptions of English pre-computer (Brezina & Gablasova 2015). Some examples of non-English vocabulary lists are the *Frequency Dictionary of Spanish Words* (Juilland & Chang-Rodriguez 1964), the *Frequency Dictionary of French Words* (Juilland et al. 1970), and *Frequency Dictionary of Italian Words* (Juilland & Traversa 1973). Scholars like Thorndike West and Juilland played, thus, an essential role in applying linguistics methods in education.

It was also at this period that several linguistic laws such as Zipf's Law (Zipf 1935, 1949), and the work of scholars such as Franz Boas, Leonard Bloomfield, Edward Sapir emerged. Bloomfield (1914, 1926), among other scholars, supported the use of scientific procedures to analyse linguistic data. Such work was relevant in setting the ground for a methodological approach that was to come, namely Corpus Linguistics. However, it was only in 1953, after the invention of, but shortly before the introduction into general use of electronic computers, that there began a project which was probably the main effort responsible for shaping the current structure of corpora. The Survey of English Use, or simply SEU, was a non-electronic corpus, compiled by

Randolf Quirk and his team in London. It has been said that most of the influential descriptive grammars in the 20th century based their description on the SEU (Jensen 2014).

The SEU was designed to have one million words of authentic language structured in a fixed number of texts (200), each containing the same number of words per text (5,000). These texts were organised onto paper cards, each of which contained one word of the corpus together with a certain amount of its original textual context (17 lines of text). Each word was grammatically analysed and assigned a category. The system of categories derived from these (manual) annotations was later used as a reference for the development of one of the earliest automatic POS taggers (Greene & Rubin 1971), which in turn was the ancestor of many if not most POS-tagging programs still in use today. Besides its importance for the creation of POS taggers, SEU structure was also a reference for the electronic corpora that were to come (see 2.4).

2.2.3.1 Father Busa

In 1946, the Catholic priest Roberto Busa decided to attempt something which, to my knowledge, had not been done before: to perform searches within the complete works of Saint Thomas Aquinas by using pre-computer punched-card tabulating machines (Winter 1999). Busa's ultimate goal was to publish, as a set of printed volumes, an index of the more than 11 million tokens of surviving Medieval Latin documents, Aquinas being merely the first phase of this more ambitious enterprise. However, he was aware that this would, if undertaken manually, be an enormous – and enormously time-consuming – task. He envisioned the use of new technologies as a means to reduce the length of time that this project would require, and also as a way to improve the accuracy of word retrieval (Jones 2016).

2.2.3.1.1 Busa's work

The task to which Busa set himself, which was accomplished in 30 years, consisted of fully lemmatising all the words in the Latin writings of Saint Thomas Aquinas. Lemmatisation consists of annotating each word in a text with that word's lemma, that is, the citation form or 'headword' of a group of inflectionally related word types, such that, for example, *went* is tagged as *go* and *houses* as *house*. The idea of lemmatising, i.e. grouping headwords and its inflected form, complete works was not new. Busa himself cited (Busa 1987), for instance, the *Thesaurus Linguae Latinae* (TLL).² which relied on contracted workers to process its ten million cards with lemmatised words. The procedure consisted of writing the lemma, the POS and the position in text for every token found in the collected texts onto paper cards. The TLL eventually created a total of around ten million cards (Corbeill 2007). Projects like the TLL addressed the incredible length of time that manual compilation of full textual index data required for such large bodies of text simply by accepting that the results would arrive in decades if not generations (TLL was scheduled to take from 1894 to 2050). By contrast, Busa's index rely on the use of pre-computer punch card machines to reduce the amount of manual labour required, and, thus, the length of time over which the project would run. Punch cards, or punched cards, are a paper-based data storage medium used by the mechanical tabulating machines which preceded, and were eventually replaced by, electronic computers. Data is coded on punch cards by the presence or absence of holes in specific positions. Once a complete data store is encoded on punch cards, a tabulation machine can be used to access the data and

² <http://www.thesaurus.badw.de/en/project.html>

perform (simple) operations such as counting by processing the cards. Thus, many important steps in the compilation of an output such as Busa's index to Aquinas can be performed automatically rather than manually (Jones 2016).

Despite becoming famous for his work with punch cards, Busa's computer knowledge was not what led him to success. He did not have any knowledge in this field before he started his work at IBM. What made his index special was the fact that he "knew the nature of the task and knew what he was looking for" (Winter 1999:9). Busa could not rely on any prior methods for his project because none existed at that point, as the relevant technology was only just emerging (Busa 1987). Hence, he had to test and develop his method as work progressed and technology evolved. One reason for his work to excel was that it was driven by his questions, rather than by the technology available. He would adapt the technology to suit his needs, rather than the other way around (Jones 2016).

Busa was not the only scholar interested in using machines to deal with very large amounts of texts. For instance, in 1957 Rev. John W. Ellison presented what he claimed to be the world's first computer-generated concordance (Jones 2016). According to an article in the non-academic press, it took Ellison "only" 400 hours to process 80 miles of tape containing the 783,137 words of the Revised Standard Version of the Bible (LIFE magazine 1957). The resulting concordance was then published in a book more than 300,000 words of running text in length.

A parallel development to these, made by the computer scientist Hans Peter Luhn, was the *Key Word in Context*, better known by the acronym given it by Luhn (1966). The concept of KWIC was based on the that of the *key word in titles* system, first proposed in 1859 by the librarian Andrea Crestadoro. The key word in titles system is an

approach in which librarians enhance the catalogue of their library by adding to the record of each book a set of ‘key’ words which provide some indication of its topic or content. This system is used to record books with words beyond the ones found in the title (Manning & Schütze 1999). It is debated whether Busa’s work influenced Luhn (Burton 1981). Either way, it is certain that these efforts undertaken during the 1950s were the foundation which would lead to the rise of academic (sub-)disciplines such as machine translation and corpus linguistics (Jones 2016).

2.2.3.1.2 Reasons to be neglected

Despite his importance, Busa did not have at his time the prestige that is now attributed to him. This might be because (i) the contemporary ideas to his work highly criticised structuralism; (ii) it took him too long to finish what he had proposed and (iii) many researchers did not fully understand what Busa was trying to accomplish.

Busa’s work was released in an era where empirical studies were not exactly appreciated. It was the beginning of the use of computers in research centres, but it was also the moment when rationalist ideas were strong. Chomsky (1957) drastically affected the dominant linguistics paradigm at that time. Chomsky’s view is that the data necessary for language analysis is accessible through introspection, and there was no reason for collecting massive amounts of data. Collecting data would only be done in order to study performance rather than linguistic competence.

There was also a shared perception among language researchers that using digital techniques for language analysis was a form of manual labour, the performance of which lacked the prestige afforded the research of the rationalist elite (Fillmore 1992; Svartvik 2007; Jemsén 2015). In fact, in the 1960s and 1970s, “corpus work was, indeed, little else but donkey work” (Leech 1991:25) due to the primitive nature of the

digital technology of that era. On top of that, empirical studies received sharp criticism (e.g. Abercrombie 1965:114-115) regarding their data collection, which was claimed to be biased and to lack credibility (Leech 1991).

Busa's proposal to rely on machines comes to address possible human errors in data preparation. Use of machines guaranteed, to a certain extent, quantitative certainty and labour and time efficiency. Although Father Busa had the support of IBM, he still had to deal with technological restrictions, such as the limited processing capacity of the machines available to him, at that time (Jones 2016). Moreover, the use of new technologies in humanities research has always been limited by availability, as cutting-edge technologies are not exactly required to answer humanities research questions. This happens because in many cases the technology is only used to reduce work time. For instance, when the index started being compiled, there were many more efficient alternatives to the punch card system used by Father Busa. Nevertheless, the punch card system, a more traditional and accessible system at that time, met the work aims, even if slower (Jones 2016).

A final issue that might have prevented Busa's works from having an impact on early CL studies is that not many people understand the main idea behind the creation of his final work, the *Corpus Thomisticum Index Thomisticus*. As Jones (2016) points out,

(f)or Busa, mechanisation was to serve hermeneutics. He aimed to interpret, to reveal meaningful patterns, different dimensions of the language. It's just that some dimensions were too extensive (while their evidence was too minute) to be grasped by the unassisted eye and mind of the reader across an oeuvre of over ten million words. Philosophical and

theological questions drove his linguistic research, and this led him to develop the complex process of literary data analysis. (Jones 2016:90)

Despite the low initial impact, Busa's work had a certain contribution to computer-based language studies. One benefit from this system designed by Busa was the drastic reduction in time necessary to process texts and also the possibility of 'atomising' them. By doing so, the text could be disassembled and then reassembled, making this a crucial moment of humanities computing. The consequence of that would be to allow human readers to visualise texts outside their linear presentation and group their elements into meaningful patterns. This feat could not be quickly done without computational aid (Jones 2016).

2.3 Digital Computing and Language

Despite the practical difficulties of dealing with large collections of texts, various scholars accomplished good results. Several dictionaries were created (e.g. OED); complete works were manually lemmatised (e.g. the Index Thomisticus).

2.3.1 Digital Humanities

Busa's automation-assisted research, and that of others in the early post-war decades, was the beginning of the field which today is called *Digital Humanities* (DH). Different names have been given to the field, such as *Humanist Informatics*, *Literary and Linguistic Computing* and *Humanities Computing*, with slight variations in their definitions (Nyham & Flinn 2016). In this thesis, DH is broadly defined as that field of academic research that relies on computational methods to address research questions within the humanities (and, for some DH scholars, also the social sciences). The methods adopted vary widely from textual corpus analysis to 3-D modelling of

historical sites (Koh 2014). From the DH perspective, computers are powerful tools which need accurate directions and well-prepared data to render optimal results.

In its beginnings (from 1949 to the early 1970s) and for a long time, DH was restricted to centres and institutions that could afford the necessary equipment, professional technicians, and maintenance (Nyham & Flinn 2016, Hockey 2004). Pioneers like Busa had the support of big institutions or companies, which allowed them to access to (mechanical) computers at a time before the personal computer. This beginning phase was crucial in emphasising to researchers the need for a well-defined methodology for the compilation and maintenance of electronic texts (Hockey 2004). The methods developed at Busa's time would be later accessible to low-resourced institutions and even individual researchers (Jones 2016). One example is a project with an approach similar to Busa's, initiated by Martin Abegg in the late 1980s. Abegg alone was able to complete extensive work on indexing the Dead Sea Scrolls using the HyperCard software (Atkinson 1987) on his personal computer (Abegg et al. 2002). Even at the beginning of the personal computing age, individual researchers were interested in processing large amounts of digital text (Hockey 2004).

Despite the initial ecstasy, from the 1970s to the mid-1980s, within DH "there was little really new or exciting in terms of methodology and there was perhaps less critical appraisal of methodologies than might be desirable" (Hockey 2004:10). However, in CL, that was the period during which important early corpora were compiled and published, and tools such as frequency list generators and concordancers started to be acknowledged (Hockey 2004:3-5).

2.4 Corpus Linguistics

2.4.1 Brown Corpus

The advent of mainframe computers in the 1960s contributed to the development of research on language. One example is the release of the first machine-readable corpus, known as the Brown Corpus (Francis & Kučera 1964), marking the beginning of CL. The Brown Corpus consists of one million words, which even for today is a decent size. This corpus incorporates 500 samples of American English texts published in 1961, of approximately 2,000 words each. The use of equally sized text samples follows the structure utilised in the construction of the SEU. The texts are distributed across numerous categories of text, with different genres weighted by their perceived importance.

Initially, the corpus existed only in the form of raw text. Later, the corpus was automatically tagged for POS, using the program TAGGIT (Greene & Rubin 1971). This tagger had a low accuracy rate of 77%, so that post hoc manual adjustments to the annotated text were required. The Brown Corpus was later used as training data for a later POS tagger, the Constituent-Likelihood Word-tagging System, better known as CLAWS (Garside et al. 1987).

The ground-breaking Brown Corpus has been widely used for different purposes, and its POS tagging enabled more sophisticated analysis to be carried out. Noteworthy applications of this corpus include the *American Heritage Dictionary of the English Language* (Morris 1969) and Francis et al.'s (1982) study of lexicon and grammar usage in English based on the Brown Corpus word frequency data (Francis et al. 1982).

Like Busa's work, the Brown Corpus did not receive the level of recognition at the time of its release in 1964. Its importance might have been obfuscated by the widespread belief that language description should not be based on real data (e.g. Chomsky 1957). The work carried out on mainframe computers had several limitations, especially regarding access. Due to its cost and size, mainframe computing was restricted to affluent institutions (Kennedy 2014:7). However, because it was the first electronic linguistic corpus (with an impressive number of words for the time), the Brown Corpus triggered the beginning of Corpus Linguistics.

2.4.2 Four generations of tools

After the milestone creation of the earliest machine-readable corpora, different paths were emerging to suit different needs, as different access methods were necessary to address different problems coming from a variety of knowledge source. However, in the late 70s and early 80s most of the tools were restricted to universities and big research centres (McEnery & Hardie 2012). It was, in fact, over twenty years after the launch of the Brown Corpus that CL analysis tools started to become available to ordinary users. In this section, I adopt McEnery and Hardie's (2012) model of four generations of corpus analysis software to discuss the evolution of CL tools.

2.4.2.1 First and second generations

The first generation (which McEnery and Hardie identify as having taken place in the late 1970s to early 1990s) are marked by tools which ran on mainframe computers; which were mainly available only at prominent institutions, such as universities; that offer minimal functionalities, and which, like most mainframe software, were not especially user-friendly. Second-generation tools came at late 1980s and differ from the first generation in the sense that they "were enabled by the spread of machines of

one type in particular across the planet – IBM-compatible PCs” (McEnery & Hardie 2012:39) and did not run on mainframe computers. These tools could mostly produce (and sort) concordance lines and word frequency data.

An example of the first generation is the COCOA concordance software (Russell 1967). This program, developed for British universities, as well as generating concordances and word frequency lists could also deal with text mark-up and metadata. However, COCOA was not exactly a user-friendly tool.

In 1978, CLOC (Reed 1978) was released by the University of Birmingham. This tool distinguished itself from the previously launched ones by being among the first such program to present a more straightforward user interface, as it was created to be used by linguists instead of computer scientists. For instance, the query syntax used to search for words by spelling patterns was “simply and easily understood”, and collocation analysis could be done quickly (Burnard 1980).

Other similar tools emerged at the same time CLOC was launched, in different universities and countries, for instance, The Concordance Generating System (COGS) (Bradley 1978) and the Text Analysis Computing Tools (TACT) (Bradley et al. 1989) in Canada and Oxford Concordance Program (OCP) at the University of Oxford (Hockey & Martin 1987).

Those tools, considered first-generation tools (McEnery & Hardie 2012), were mainly restricted to institutional usage. But it did not take long for new tools (second generation tools) to become also available for less privileged institutions (first and second generations) or individuals (second generation) (McEnery & Hardie 2012). Six years after the release of OCP, its micro-computer implementation, Micro-OCP (Hockey & Martin 1987), was made available for personal use. The software allowed

the generation of word lists, concordances and indexes of texts in different languages. Although Micro-OCP could not properly perform lemmatised concordance, as the OCP did, its advantages included a friendly interface, as described in a review of the software:

Micro-OCP allows for excellent flexibility in the definition of the input and output formats as well as the type of index or concordance to be made. The program is completely menu-driven and requires no programming experience or technical understanding of the computer. (Jones 1989:131)

An influential tool that emerged around the same time as the Micro-OCP is MicroConcord (Johns 1986) (second generation). This tool had a focus on usability, as its target audience was language learners and teachers (Johns 1986), a public that may well not have extensive computer expertise.

Other concordancers from the second generation are the Longman Mini-Concordancer (Chandler 1989), the Kaye concordancer (Kaye 1990), and the Simple Concordance Program (Reed 1997). Although important at the time of their release, these tools had several limitations.

For instance, the Longman Mini-Concordancer had the downside of covering only Latin letters, having a high cost, and being able to deal only with very small corpora. However, it had a user-friendly interface, especially when compared to its competitors (Johnson 1992), such as MicroConcord, and led the way to a new generation of tools to come.

2.4.2.2 Third generation

The increasing prevalence of personal computers over the early to mid-1990s allowed more researchers to have access to corpus linguistics. An example of a third-generation tool is Mike Scott's WordSmith Tools.

WordSmith Tools stands out as one of the few pieces of corpus analysis software created before the 2000s (Scott 1996) that is still maintained and updated and in widespread use. Since it had many users from different backgrounds, Scott, its developer, had access to substantial feedback, which was crucial for the tool's subsequent improvement. As Scott points out,

The aspect of unpredictability came in with my increasing realisation that, again like Margaret Thatcher I was very often wrong. For example, I had simply assumed that any wordlist would necessarily fold all cases into one, let us say upper case, until some people asked me not to. (Scott 2008:101)

The frequent release of updates to WordSmith Tools in the late 1990s, and its growing popularity in the period, put Scott in the position to create novel tools and techniques within the WordSmith suite. In the process, Scott coined a number of CL terms and concepts such as 'consistency', 'standardised type-token ratio', 'cluster', and 'key key word' (KKW). *Consistency*, also known as *range*, deals with how regularly a word is found in different text-types. The standardised type-token ratio is a statistical measure of lexical variation that is not vulnerable (unlike the original type-token ratio metric) to influence from the length of the text that is being measured. It is calculated as the mean of a set of separate type/token ratios calculated for equally-sized short sections of the text or corpus. A cluster is any group of words in sequence. Scott notes that "the

term key-key word was probably a failure” (Scott 2008:105), since, contrary to expectation, most KKWs do not reveal much about the data. As of 2020, WordSmith Tools is in its eighth released version and still very widely used by researchers.

In the late 1990s and early 2000s, other third-generation tools were released with the same principle as WordSmith, that is, to make corpus linguistics more accessible to users. These other third-generation tools include MonoConc (Barlow 2002, 2004) and AntConc (Anthony 2002). Programs like these give the user extensive control of the search queries, allowing for the use of regular expressions, POS search and complex text mark-up. These programs also permit the extraction of collocation and comparison of word lists.

One of the reasons the tools in the section are considered third generation is the fact they can process considerably large corpora (as large as one million words) on the user’s computer. A more sophisticated piece of software, that is also considered third-generation, is XAIRA (XML Aware Indexing and Retrieval Architecture) (Xiao 2006). This open-source tool can deal with any XML corpus, and it is an evolution of SARA (Aston & Burnard 1998), the retrieval software issued as part of the first distribution of the British National Corpus (BNC). XAIRA is a borderline tool because its system relies on a client/server split but still on the user’s same machine. This client/server mechanism is the basic mode of functioning of the corpus analysis tools of the following fourth generation.

2.4.2.3 Fourth generation

The tools within the fourth generation differ from the others by the fact they operate on client–server model. This means that the tools are accessed via a web-browser and most of the workload is done on the server side. Hence, users can access very large

corpora even from computers with limited power. Lehmann et al. (2000) developed BNCweb using, at first, SARA server software to search the corpus. This web-based program allows users to access the BNC and its metatextual annotation through a web browser. Later, to process the corpus, BNCweb used a now open-source collection of tools called *Corpus Workbench* (CWB) (Christ 1994). CWB is a toolkit for indexing, managing, and querying large text corpora of up to 2.1 billion words. It has a specific focus on supporting corpora that are linguistically annotated. CWB was not designed for beginners and required the user to be at least somewhat familiar with Unix command-line tools (Evert and Hardie 2011).

Hence, BNCweb was an important landmark in the design of corpus linguistic tools. It allowed users to fully explore a heavily annotated corpus without requiring extensive knowledge of computing (Hoffmann et al. 2008:25). Other tools have followed the example of BNCweb by creating a user-friendly interface to CWB, such as AC/DC (Santos & Bick 2000); IntelliText (Wilson et al. 2010); TeiTok (Janssen 2018); and CQPweb (Hardie 2012; see further 3.4.3).

Another prominent program of the fourth generation is SketchEngine (Kilgarriff et al. 2004). This tool was originally designed for a primary user group consisting of lexicographers. In fact, Oxford University Press (publishers of the Oxford English Dictionary) was the first user of SketchEngine. Macmillan publishers was subsequently the first user of the *Word Sketches* tool which gives SketchEngine its name. A Word Sketch is a summary of the collocational and grammatical behaviour of a given word (Kilgarriff et al. 2004), derived by calculating collocates classified according to the grammatical relationship they stand in to the word being sketched. Its back-end system, Manatee (Rychlý 2007), is very similar to CWB (and Manatee's

support of linguistic annotation in the same manner as CWB is what underpins Word Sketches). In consequence, the functionalities and performance of CQPweb and SketchEngine are much alike despite differences of user interface. However, unlike the tools mentioned in the previous paragraph, SketchEngine is a commercial product. Because it is a paid service, more attention is given to the software's user-friendliness. SketchEngine aims to, and does, attract user from wide range of research areas, including lexicography, Natural Language Processing (NLP), translation, discourse analysis, language teaching, and terminology (Kilgarriff et al. 2014).

Another fourth-generation tool is the BYU online corpus platform (Davies 2004-). Unlike most of the software covered in this section, the BYU platform does not allow the user to install their own corpora or download the software for their own use. However, it makes a growing number of online corpora available for free. According to the BYU corpora website, an average of 130,000 unique people accesses the platform each month³.

2.4.3 Implications for Linguistic Theory Considerations

As the amount of research using CL methods, resources and tools increased (McEnery & Wilson 2001), more and more linguists adopted some paradigm of linguistic analysis in which both association and frequency matter. In such approaches, linguistic features are not seen as events that happen by chance. Instead, quantitative investigation of linguistic units can make use patterns evident. Researchers who use CL methodologies generally accept that the usage-patterns or discursive behaviour of

³ <https://www.english-corpora.org/users.asp>

a linguistic unit can be identified by investigating its association patterns (Biber et al. 1998:5).

Language patterns thus identified also present regularities and are stable in distinct moments. That is to say that the patterns have comparable frequency when different events are being observed. They may also present systematic variation across textual varieties, genres, dialects, time, etc. To draw conclusions about language based on these patterns, it is necessary to test hypotheses. From textual frequencies, we can estimate theoretical probabilities (Sardinha 2000). Therefore, to fully understand the use of these patterns of association and frequency, it is necessary to investigate how regularly they occur through quantitative analysis, before then moving on to a qualitative analysis of linguistic features.

Qualitative analyses are often made with the use of concordances, the functionality to generate which was present in the earliest CL software. A basic concordance tabulates the hits for some corpus query, each together with some small amount of the preceding and following co-text, so that the item queried can be studied in its real context. Such basic concordances vary little in different pieces of corpus analysis software. However, advanced tools can display concordance data in more sophisticated ways. For example, different colours can be applied to the text in a concordance to visually indicate analytically significant annotation (see figure 1.1, where colour indicates); tooltips can show text metadata (figure 1.2); and XML tags can be displayed or hidden, according to the users' need (figure 1.3). No matter how many functions a CL program offer, concordance lines is usually the ultimate means of analysis, as they reveal the context of occurrence, which is key for linguistic analysis.

21	2002	OLTL	A	B	C	, Nigel . # Rae: Do you really want to be like Asa Nigel: Sir , please , if there 's any
22	2011	GH	A	B	C	at Pentonville . I know what happened ; You dangled Kristina like bait in front of her loving daddy . He went berserk and
23	2009	ATWT	A	B	C	is different . Lily: Be honest with yourself ; Dusty: I like being around Meg and her kid . I like being around kids
24	2009	YR	A	B	C	ask you out on a real date ; There 's nothing like being harangued ; It 's too bad , ' cause I was
25	2004	ATWT	A	B	C	the ground . (Alison-laugh) Aaron: I 'm serious ; It was like being in that ring , right there and having some big dude
26	2011	DAYS	A	B	C	know why I 'm -- all I know is that I like being with this sane person a lot . Jennifer: I know .
27	2009	GH	A	B	C	saying . You have a child and you start owning things like burp cloths ; How frightening . No offense . I just do
28	2008	YR	A	B	C	, bye-bye . Wow . You are smooth ; Nick: Smooth like buttah ; Victoria: You really know how to go in there and
29	2001	AMC	A	B	C	been there , and so I -- Chris: Makes you feel like Cinderella ; Erica: No . Actually , Cinderella got to go to
30	2009	OLTL	A	B	C	the shoes back by midnight . # Wes: So you feel like dancing ; Marty: Hmm-hmm . There 's this big charity event at
31	2010	ATWT	A	B	C	right . Liberty: But first things first ; Janet: Yeah ; like foie gras ; Liberty: Mm-hmm . Janet: Because we can . Mm-hmm
32	2011	OLTL	A	B	C	No . This was a real woman who looked so much like Gigi ; she had to have been -- Aubrey: Rex , it

Figure 2.1: screenshot of a concordance in BYU corpora

15	S2K7	539	seventeen something S0337: yeah S0339: and one of the rebuilds is S0337: >>a white horse at the end --UNCLEARWORD random S0339: >>but the horse standing there S0337: ju
16	S2K7	540	is S0337: >>a white horse at the end --UNCLEARWORD random S0339: >>but the horse standing there S0337: just yeah --UNCLEARWORD the front door S0338: --UNCLEAR
17	S2L4		we would say like S0423: I 'm so hungry I could eat a horse S0421: I could eat a horse yeah that 's weird as well is
18	S2L4		'm so hungry I could eat a horse S0421: I could eat a horse yeah that 's weird as well is n't it ? why would
19	S2L4		as well is n't it ? why would you be eating a horse ? S0423: >>you 'd think the French would say that though S0421: yeah S0423: they
20	S2L4		guess cos it 's quite a big animal is n't it a horse ? S0421: yeah S0423: probably where it 'll be coming from (...) S0421: I think the
21	S2TP	231	also so the site where erm the whole estate is an old horse hospital and there 's quite a lot of bones S0320: an old () horse

Figure 2.2: screenshot of a concordance in CQPweb

1	<input type="checkbox"/>	Written books and ... hat the Great Witcombe mosaic combines the two: simple guilloche for the axes of the grid, and a three-strand guilloche for its border. </p></s><s><p> A m
2	<input type="checkbox"/>	Written books and ... tilted squares is preserved, this supports a guilloche knot (aligned with the axes of the grid framework). </s><s> Parallels for this treatment include the tiltec
3	<input type="checkbox"/>	Written books and ... incentives for green investment. </s><s> All such considerations have been axed from the final bill which permits drilling along the Californian coast and in th
4	<input type="checkbox"/>	Written miscellaneo... DARK SECRET would be revealed. </p></s><s><caption> Neolithic stone axe </caption></s><s> IMAGINE HOW A CELT FELT WITHOUT GUINNESS <
5	<input type="checkbox"/>	Written books and ... ial farming. </s><s> The virgin lands of America were cleared with fire and axe , as in the middle ages; explosives for removing tree-stumps were at best a
6	<input type="checkbox"/>	Written books and ... I </s><s> Advising the Ripper to dekind casual love </s><s> With hammer, axe and sharpened tool, </s><s> To gouge and bash those girls just out of schc
7	<input type="checkbox"/>	Spoken context-go... 's><s> I it's, it's very hard when you, when you've, when you're grinding an axe to get the true mood of the meeting. </s><s> Could I just have a small poll,
8	<input type="checkbox"/>	Spoken context-go... the marks be found? </s><s> What would you get marks for? </s><s> The axes . </s><s> Okay, X and Y. </s><s> The points </s><s> Which parti particular
9	<input type="checkbox"/>	Spoken context-go... resting parts of the shape? </s><s> Where it crosses through the X and Y axes . </s><s> Okay, X and Y axes, anything else that might be an interesting pc
10	<input type="checkbox"/>	Spoken context-go... 's><s> Where it crosses through the X and Y axes. </s><s> Okay, X and Y axes , anything else that might be an interesting point? </s><s> Erm </s><s> . <

Figure 2.3: screenshot of a concordance in Sketch Engine

Quantitative analysis can vary from the generation of the simplest frequency lists to advanced statistical calculations. Frequency lists are probably the most commonly adopted means of corpus analysis (Gries 2010), and certainly among the oldest. Such tools generate displays listing all words or sequence of words that occur in the corpus being analysed, usually ordered by descending frequency. These lists have proven useful in giving a general overview of the corpus being analysed; however, it is rather simplistic.

A word's behaviour in a concordance, or frequency in the corpus, can be considered to constitute a very basic kind of pattern. In order to perceive more nuanced or unusual patterns, more sophisticated quantitative methods can be used. Different CL concepts,

methods and tools have been created to identify less evident patterns and allow a more in-depth analysis. Such techniques include the investigation of collocations, keywords and n-grams.

New technologies have made it easier to create multi-billion-word corpora. This kind of corpus is exemplified by iWeb, a 14-billion-word corpus (Davies 2018-). Corpora such as iWeb allow patterns of language use to be identified that it would not be possible to observe even with relatively good size corpora such as COCA (Davies 2008-) (Davies & Kim 2019). There are still issues with corpora of this size. For instance, querying such large corpora can be slow. Another issue is that this big data is normally presented as a single mass of text, without the possibility of dividing the dataset into subcorpora according to, for example the type of registers.

Another means of dealing with patterns, is to observe them not only with the words and group of words themselves but also with their classifications. For this type of analysis, it is necessary to have an annotated corpus with textual mark-ups.

2.4.4 Annotation, textual mark-up, and encoding

2.4.4.1 Annotation

Corpus annotation is highly variable in nature. Forms of annotation include semantic (e.g. Piao et al. 2015); morphological (e.g. Schmid 1994); syntactic (e.g. Marcus et al. 1993); morphosyntactic (e.g. Bick 2014); discourse-pragmatic (e.g. Kirk 2016) and problem-oriented (e.g. Kirk 1994). Tagging all the tokens of a corpus for their POS and a small number of related grammatical features, a process called POS tagging or more formally morphosyntactic annotation, is the most common type of annotation applied to English texts and corpora in CL.

When corpora began to be annotated, the levels of annotation applied were simple. However, as the tools evolved, more levels of linguistic knowledge started to be incorporated into the texts and corpora. However, corpus annotation programs are not very popular among language researchers and linguists, as they require considerably higher computer expertise.

2.4.4.2 Textual mark-up

Annotation is not the only way of enriching a corpus. Elements of the appearance of the original document such as paragraphs, titles, or font rendering can also be indicated within the body of a corpus text: the symbols that encode such information, as well as the process of introducing them into the text, is referred to as *mark-up* (or, more precisely, *textual mark-up*). Some of the earliest marks, such as the > and the * are still present in current systems of text mark-up. Since its emergence, different kinds of mark-up have been developed. For instance, Busa's encoded text utilised the asterisk before a true upper-case letter, as the system he was using did not support an uppercase/lowercase distinction, Latin words were used to indicate different positions in the text (Tasman 1958). More recently, standards like XML have emerged and facilitated the management of text-mark-up (Hardie 2014).

2.4.4.3 Encoding

Human civilisation has spawned a plethora of writing systems, many of which exhibit somewhat (or very) illogical structures (Moron & Cysouw 2018:1). Because of that, the representation within a computer program or data storage of the full array of characters used in these writing systems can be problematic. For many years, character encodings – mappings between numeric values stored in computer memory, and the written symbols they represent – were limited to a fairly small number of character codes because of hardware memory limits. One-byte character encodings

can code only 256 characters. ISO 8859-1, an example of a one-byte encoding, can cover the Latin alphabet (plus punctuation and some mathematical symbols) only. The relevance of this issue in the present context is that corpus analysis tools that used character encodings of this type – which most did, because that was what the computers supported – were effectively bound to a small subset of the languages of the world and unable to work with texts in languages outside that subset. A tool programmed to process ISO 8859-1 would be completely unable to deal with Chinese characters, for example.

The advent of the Unicode Standard solved this problem (Moran & Cysouw 2018:3) by abandoning the use of character sets limited to 256 characters or some equally small number. Unicode can represent up to 1,114,112 possible characters, of which 143,859 characters have been defined to date (The Unicode Consortium 2020). They also provide compatibility with previous systems and early standards.

2.4.5 Consistency and tools

As the previous sections have shown, technology for managing linguistic data has evolved greatly. Linguistic research is no longer drastically limited by what computers can do. Rather, in many cases, the factor restricting corpus research methods is limitations in what researchers know how to do with the computer and software at their disposal. In part, this is merely a consequence of the lack of general computer expertise among researchers outside computationally-oriented disciplines. But another reason for the limits in researchers' knowledge of the computer-based techniques available to them is disagreement regarding standards.

Since the beginning of DH, researchers have attempted to establish standards and also of providing easily retrieved linguistic data. Busa, for instance, had the ambition to

create and maintain international centres around the world (Jones 2016). The need for standardization is real and still challenging to this day. Different consortia and initiatives have emerged to set a standard such as the Text Encoding Initiative (TEI). There also exists a plethora of software that can interconvert different text formats, encodings, and annotation schemata, such as Pepper (Zipser et al. 2011), and AntFileConverter (Anthony 2017).

Such conversion tools are in many cases enough to prepare a corpus for processing by some specific piece of software. However, they are typically not very user-friendly. In order not to become obsolete and also to meet the needs of non-expert users, some CL software is able to accept a variety of input formats by incorporating a reformatting tool which converts everything to the tool's preferred data format, without troubling or even informing the user. This capability is present in SketchEngine and #LancsBox (Brezina et al. 2018), both of which also automatically annotate the input data for POS after it has been reformatted as necessary.

Another issue is that “while persistent XML representations and nomenclature have advanced substantially in their coverage power and adaptability to new uses, corpus search systems have lagged somewhat behind” (Krause & Zeldes 2016:1). While programs developed for specific projects (e.g. CLiC – see 6.2.4) are able to fully exploit the XML mark-up of the corpus or corpora they target, software that supports fully XML aware queries for “any generic” corpus are very rare (Krause & Zeldes 2016). Corpus query formalisms that can refer to XML structures include are SketchEngine's Corpus Query Language (CQL), the CQP syntax used in CWB (of which CQL is a minor variant), and the ANNIS Query Language (AQL). But these query languages tend to be too complex for the non-expert user. In order to get results

as accurate as possible, the queries in this type of systems have to be very specific, which requires learning the query language, knowledge that most users do not have.

2.5 Discussion

This chapter has shown that CL software is now reasonably well-established. Many different tools have evolved to offer accessible means to reorganise corpus data, retrieve meaningful information, and offer new perspectives on language.

CL methods are now used for highly varied purposes, including language description (e.g. Biber & Finegan 1988; Biber et al. 1999), second language education (e.g. Granger 1996, 1998), discourse studies (e.g. Hardt-Mautner 1995; Partington 2004; Baker et al. 2008), and stylistics (e.g. Mahlberg 2012).

This broad range of fields exposes the immense applicability of CL. Given this, it is unsurprising that we may observe different users with different research questions making use of the same tool(s) and data for their different purposes. For instance, while a lexicographer might be interested in studying a corpus of Shakespeare's work to find out about the evolution of a word's meaning, a stylistician might use the same data and even the same software to address questions in the area of literary studies. What is irrelevant to one type of research might be important evidence in another (Owens 2011). That being the case, corpus tools should be simple enough to grant users a certain level of flexibility – so that they can manage, query, and visualise their data according to their respective needs.

The growing size of corpus data and its increasing number of types of annotation and mark-ups also require that new CL software be able to deal, in a user-friendly way, with many different layers of information. If in the past it was difficult to recognise

patterns in the data, now, with so many variables, it can be even harder. Relying on more elaborate statistical exploration and visualisation has become a common response to this problem. In the past, researchers worked with fewer variables. For instance, Busa worked with texts written by the same author, in a single language, in a specific context and in a single genre. With corpora such as the Brown Corpus and the BNC, more variables were present in the corpus data, such as genre, mode, and speaker age. Currently, there are many big corpora, with highly complex metadata.

Back in Busa's time, compiling a corpus was the milestone. Now, the breakthrough needed in linguistics is to see all the data at the same time and derive meaningful information from it. Therefore, CL techniques and tools should be as straightforward to use as possible, so as not to obstruct text analysis itself.

3 An exemplar-based review of corpus analysis software tools

3.1 Introduction

This chapter discusses four tools used in Corpus Linguistics (CL). A tool (or software) can be understood here as set of elements put together to perform a task. The goal of this chapter is to identify prominent practices in the software frequently used for corpus studies, mainly by non-specialist users of corpus data and methods (NSUs).

The chapter first presents the framework that I use for the review. The subsequent sections discuss each piece of software. The chapter concludes with an evaluative discussion of the tools reviewed.

3.2 Framework for the software review

Because this chapter covers tools from different natures, it is not reasonable to review them within a fixed structure. Instead, I use a set of predetermined principles, outlined in this section, to guide each software review. I set forth these principles on the basis

of previous research into tools, common practice in academia, and established standards (e.g. Wiechmann & Fuhs 2006; Hardie 2012; ISO standards).

3.2.1 The criteria

Wiechmann and Fuhs (2006) use *functionality*, *performance* and *usability* as the three main points of analysis in their review of ten concordancers⁴. I return to a detailed discussion of these criteria in 3.2.1.1 to 3.2.1.4; they are broadly related to the software's *usefulness*, *reliability* and *simplicity*, respectively. Wiechmann and Fuhs highlight the assets of each tool and describe its potential target users. Tools that do better in terms of usability are regarded as ideal for beginners in CL, while tools with more functionalities are ideal for proficient users. While good performance is important for any kind of user group, expert users favour tools with more functionalities, and usability is crucial for beginners. Because Wiechmann and Fuhs's article was published some 14 years ago, some of the tools that they discuss have fallen out of use or are no longer maintained by their creators. Thus, despite the valuable example that Wiechmann and Fuhs provide of how to undertake a software review for corpus analysis tools, the review itself cannot be relied on as a guide to current needs in CL research. For instance, the Concordance Software,⁵ which was created in 1996, became unavailable for download in 2016.

⁴ MonoConc Pro 2.2, WordSmith Tools 4, Concordance, Multi Language Corpus Tool, ConcApp 4, AntConc 1.3, Aconcorde, Simple Concordance Program, Concordancer for Windows 2.0 and TextSTAT 2.6

⁵ <https://www.concordancesoftware.co.uk/>

Another important point to consider when evaluating software is *flexibility*. A flexible tool should be able to deal with various types of corpus and annotation. It should also be sophisticated enough to allow fine-grained research (Hardie 2012; Soehn et al. 2008: 27). This characteristic is beneficial even for beginners, who might experience different types of linguistic knowledge before specialising in a definitive area. The analysis is carried considering four elements described below: functionality, performance, flexibility and usability.

3.2.1.1 Functionality

Functionality can be defined as what users can do with the tool. The usefulness of a given research tool to the user over the course of their work with it, from data input to analysis, is thus dependent on its functionality (Weik 2000).

One criterion of functionality is the *input data format* that the software accepts. To assess this, we may consider elements such as: which data format the tool can accept as input (whether just plain text files, or binary presentation formats such as Portable Document Format (PDF); or word processor files such as Microsoft Word documents); what mode of management of corpus texts the tool employs (whether the tool treats each input file as equivalent to a text, or instead respects text-boundary mark-up, or treats the entire corpus a single undifferentiated entity); whether, and how, the tool is able to utilise textual metadata (be it embedded in the corpus files or as a separate file); how wide a range of character encodings, writing systems, and languages the tool supports; and to what extent the tool is aware of and able to process different annotation schemes that may be present in corpus (or, alternatively, any capacity it has to annotate data itself. One way of assessing this aspect of corpus tool functionality is to evaluate how much effort is necessary to prepare the data for input into the software.

After the corpus is loaded, the software needs to deal with *data querying and retrieval*. To assess how well a tool does on this task, we can, for instance, evaluate the query language that it exposes. We may consider: the power of the query language (that is, the affordances it makes available, such as wildcards, regular expressions, match strings, flexible searches); and means of dealing with the metadata, annotation and results. A tool that permits elaborate queries to be composed that may reference different annotation layers as well as the forms of tokens and token fragments will be evaluated as having greater functionality than a tool that only allows for simple queries.

Functions more advanced than simple queries should also be considered. The existence of *extra functionalities* such as creating keyword lists, generating n-grams, calculating collocations, and rendering statistical summaries of corpus frequencies is a factor that will necessarily lead to a more positive evaluation. The more functions a tool offers, the better its functionality.

3.2.1.2 Performance

Functionality is not everything. A piece of software that does many complex things poorly is far inferior to one that does one simple thing well. In addition to considerations of accuracy of outputs, the performance of a program may be defined as how efficiently it makes use of the limited computer resources of memory (RAM), processor time, and disk (or network) read/write bandwidth. A program which minimises its use of these resources when carrying out a given task will seem, to the user, to work more quickly and smoothly than a program that does not. The former program can be described as having a higher *performance* than the latter. Equivalently, a higher-performance program can process *more* data or carry out a

more complex operation with the *same* resource requirements as a lower-performance program.

However, evaluating performance of corpus software is not easy, because there is no agreed single corpus size which all tools must be able to handle, nor an agreed minimum time within which all tools must be able to complete any given task. Both these factors vary according to the user's needs as well as the main purpose of the software (for instance, if a tool is mostly designed for collocation statistics, poor performance in rendering a concordance may be quite acceptable). For this reason, when evaluating this criterion, the proposed or main usage of the software ought also to be considered.

Consistency, on the other hand, is crucial. The results from a specific query in a corpus should always be the same, no matter which computer is being used, for instance. If a programme offers different results when, say, using different versions of the software or installing in different operating system, it is then not considered reliable.

3.2.1.3 Flexibility

Hardie (2012:403) defines flexibility as “the possibility of using a tool with any corpus (...) word-level annotation, or none; with any amount of text-level metadata, or none; in any language and any writing system”.

Similarly, flexibility here is related to the ability of the software to process any (to a certain extent) corpus; to support different types of annotation and text metadata. I would suggest the additional criteria that a flexible tool should be accessible from multiple different operating systems and devices (PC, tablet, etc.). Finally, although I treat them as different criteria, flexibility and usability sometimes intersect. For

example, a tool that provides easy text metadata management is both flexible and user-friendly.

3.2.1.4 Usability

This point mainly refers to how users receive the tool. It is used to evaluate how user-friendly the software is. The quality of user-friendliness is defined as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (ISO 9241-11). The more a software works in the way that a naïve user would expect it to, the higher its usability.

User experience cannot be quantitatively measured. However, some techniques exist that can be used to verify quality in user-experience. There are some design principles to be considered when developing search interfaces and information retrieval for searching systems such as web search engines (Google, Bing, and the like) or library search engines, which like web search are now typically accessed within a browser.

For instance, Hearst (2009) suggests some principles to follow when designing search interfaces for text. First, the tool should provide informative and efficient feedback. That means that (i) the results should be returned quickly if not immediately; (ii) the query term should be indicated somehow (e.g. in bold, highlighted) within the results returned; (iii) sorting the results should be possible, so users can easily identify the results that are relevant for them.

Second, users should have some kind of control of the search mechanism but should not be overloaded with options. The system should provide default procedures that fit the typical user’s needs. In the context of CL, an example practice which follows this

principle is that, in many tools, queries are case-insensitive by default, but the user has access to an option to change that if they need a case-sensitive query.

Third, an interface should minimise the need for the user to remember all the settings used and all the steps taken to arrive at a certain result. For instance, the interface can provide the users with traces (often called *breadcrumbs*) of all the steps the user has taken to reach the current display of results. An example of this in action is the Google web search engine's use of breadcrumbs to indicate criteria added to an image search; these breadcrumbs are highlighted in yellow in figure 3.1.

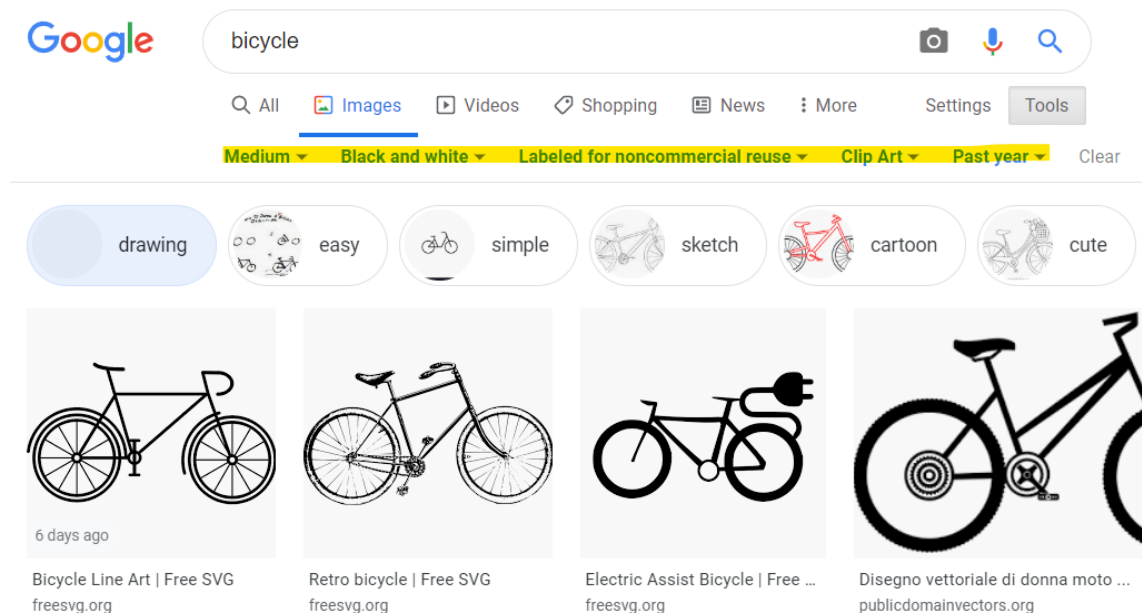


Figure 3.1: example of search breadcrumbs

Fourth, a search interface should include shortcuts and hyperlinks to enable users to go back and forth through the results quickly. An example of an application of this principle is when users can right-click on a result and open new tabs with the result in its integrity.

The fifth principle is that small details are essential. For instance, it is known that bigger entry boxes in forms presented in the interface prompt users to type longer

queries (Franzen & Karlgren 2000), all else equal. So, if in a given application, users are expected to type long queries in normal use, the box into which the query is typed should be of a larger size than the default.

Finally, Hearst (2009)'s last principle refers to aesthetics' crucial role. Parush et al. (1998) showed that users might take up to twice as long to perform tasks using an unappealing search layout as opposed an attractive one. Aesthetics are also crucial in users' decisions on whether or not to use a given tool in the first place (Hassenzahl 2004; Lindgaard and Dudek 2003). Aesthetics can be quite subjective, but there are established means to make an interface more visually appealing. For instance, the interface page presenting the results should be cleanly laid out, using a typeface with good readability. A clean interface is defined as one that follows principles such as *information hierarchy*, i.e. that the more important any given part of the content is, the more it should stand out stands out, and the Gestalt Principles, a set of rules regarding human perception of visual objects (Tidwell et al. 2020).

Despite the aforementioned principles to enhance interface aesthetics, usability might be abstract and dependent on user reception. For this reason, the review presented here will not extensively focus on it. The issue of software usability will arise again, in more detail, in 5.5.

3.3 Selection of tools for review

The selection of a small number of specific tools to be discussed in this review was made in light of the target audience described in 1.1, which is mainly NSUs. Therefore, I opt to discuss tools that NSUs are more likely to use. For the sake of the present review, I make the following four assumptions about NSUs.

First, they are mainly beginners, or else not specialists in CL. For this reason, they are unlikely to be willing or able to invest money in corpus analysis software. The same would apply even to NSUs whose access to research software is via a university or other educational institution, since the university might equally well lack the resources to purchase or license corpus analysis software for NSUs (or have other spending priorities than corpus software). Hence, my first criterion was that this survey should look only at tools available at no cost.

Because software becomes obsolete very fast (Ford & Richards 2020), the second criterion was to select for analysis either recently launched software, or older tools that, despite their age, are frequently updated and actively maintained. Such tools may be reasonably assumed to be, or to aim to be, in line with current needs of NSUs.

A further point to bear in mind is that NSUs could come from different areas in language studies, and, therefore, have different reasons for beginning to work with CL methods. But tools may differ in terms of which (sub)set of the disciplines in question they are intended to appeal to. For instance, Kilgariff et al. (2012) suggest that the Sketch Engine is mainly used by lexicographers, whereas the BYU corpora (Davies 2004-) are mostly used in second language education (e.g. Poole 2018; Bennet 2010). For this reason, the third criterion was to include in the review tools with support for at least some range of possible purposes, research goals, or applications. Thus, I aimed to select tools with one of the following focuses: quantitative analysis; simplified analysis; and powerful linguistic investigation.

The final criterion was that the selected tools should differ in terms of the environment they are used in. Thus, the review will encompass at least one piece of software that can be accessed via the following three architectures: web browsers; local installation

and bespoke software purpose-designed by the researcher for their own needs. This is because different platform can also entail different software behaviour.

Based on the aforementioned criteria, I chose to analyse the following software: CQPweb 3.2.42 (Hardie 2012), AntConc 3.5.8 (Anthony 2019), #LancsBox 4.5 (Brezina et al. 2018), and Quantedata 2.0.1 (Benoit et al. 2018). The versions in question were the most recent stable versions of each program as of this writing. All five pieces of software also offered guides and documentation on how to use them. Other tools will be touched on only briefly when relevant to the detailed consideration of these four.

3.4 The tools

3.4.1 Standalone tools: AntConc and #LancsBox

3.4.1.1 WordSmith Tools: the beginning

As I will demonstrate in chapter four, WordSmith Tools is, by far, the most cited piece of software in CL. It was one of the first CL tools made available for individual researchers, rather than institutions. Since its launch in 1996, WordSmith has evolved through eight major versions. It started as a simple concordancer, and then over time new functionalities were created and implemented. These include minimal pair identification, which helps finding typos and minimally differing pairs of words; an alignment tool for parallel corpora; and a corpus checker, which looks for file corruption, duplicate files and boilerplate. WordSmith Tools has also served as a reference for many other tools such as AntConc and #LancsBox.

The importance of WordSmith Tools in CL research is immeasurable, and its use is prominent in the field of linguistics. Extensive research has been undertaken on, or done with, WordSmith Tools (e.g. Wiechmann & Fuhs, 2006; Smith et al. 2008;

Rodríguez-Inés 2010; Wilkinson 2011); substantial documentation on the software is available (Scott 2012; Scott 2020). Because this literature already reports on common practices adopted by users of WordSmith Tools, I opted not to include it in this review. Moreover, as a commercial piece of software, WordSmith Tools does not meet the criteria stated in 3.2.1. Instead, this section will investigate two freeware tools, #LancsBox, a relatively recent development; and AntConc, which is widely used across the world (see 4.3).

3.4.1.2 Importing the corpora

AntConc and #LancsBox differ in the way that they import corpus data. When a corpus is imported for the first time, #LancsBox first annotates and indexes the corpus. When texts are indexed, a map of the tokens in those texts is created, making information retrieval more efficient, as it is not necessary to search the whole original text from start to end (Gupta et al. 2014). Depending on the size of the corpus, the specifications of the computer and whether the corpus is annotated or not, this process can take a long time. This is not ideal, especially in classroom use, where time is limited. However, indexing makes subsequent queries much faster. By contrast, loading files in AntConc is immediate, as it does not index the data. However, in consequence, subsequent queries are slower than they would be in #LancsBox. Moreover, AntConc does not provide corpus information such as type and token count when the files are loaded, but only when a word list is generated.

#LancsBox offers some built-in functions that make the software ideal for beginners in CL who do not know how to obtain an existing corpus or create one of their own, or how to deal with text annotation, formatting and encoding. After downloading the program, users of #LancsBox can immediately use it without having to create or obtain corpus to import into it. As of this writing, 12 built-in corpora and ten wordlists

were freely available in #LancsBox. If the users want to create or use their corpora, #LancsBox also uses Apache Tikka (The Apache Software Foundation 2020) to automatically detect the format and character encoding of the uploaded text. This means that users can easily load files in a wide variety of formats such as word and pdf documents, without need to change any of the import options. These affordances are applied automatically and are concealed in the interface, which makes it simple and easy to use, but also offers users with advanced settings, should they need them.

The uploaded texts are automatically POS tagged and lemmatised with TreeTagger (Schmid 1994). Users can choose among 23 languages, English being the default. The language option is clearly displayed, as appropriately given that this is most likely the only setting that a NSUs will be willing to change.

Like #LancsBox, AntConc detects the character encoding automatically but also allows users to select from a vast list of encodings. This helps ensuring that the right encoding was chosen (when this selection is made manually) at the same time as usability (for the automatic encoding detection). AntConc also has a simple interface for changing settings such as token, tag and wildcard definitions, although the default settings are very likely to work well for beginners in CL.

3.4.1.3 Tools within the application

AntConc and #LancsBox offer essential tools in CL, such as the generation of concordance lines; collocations; and frequency lists of types, n-grams and keywords (McEnery & Hardie 2012). Both pieces of software have a similar graphical user interface based on tabs. This helps users navigate among different tools. A difference is that in #LancsBox, unlike AntConc, more than one tab for a given tool can be open at the same time. This allows users to see multiple analyses at the same time.

#LancsBox also allows for the import of more than one corpus at the same time. The tab system is particularly helpful for corpus comparison, as the user can navigate through tabs with the same queries for different corpora.

The purposes of the tools within each program are very similar, although they differ in small aspects. For example, AntConc relies on visualisation to show the dispersion of query results within the texts of a corpus (see 6.2.1), while #LancsBox graphically displays collocation networks. Although the graphical display in #LancsBox might first attract new users, the display of collocation on a table is much more efficient and clearer.

Both tools were designed by and for linguists. Hence, they offer basic but relevant statistical calculations. Because #LancsBox and AntConc run on users' local computers, the performance of both is affected by the hardware. Hence, processing large corpora might run well on some machines but halt in others.

Overall, AntConc is lighter than #LancsBox (and for this reason require less computer processing usage) and offers more flexibility in terms of altering the settings. That makes it an excellent tool for users with some basic knowledge already. AntConc also has the benefit of having been around for almost twenty years (Anthony 2002). It is well-established and has tutorials in nine different languages⁶. It also has other related software that complement one another. For instance, FireAnt (Anthony & Hardaker 2017), a tool to download tweets, and AntPConc (Anthony 2017), a tool to visualise parallel corpora. #LancsBox, on the other hand, concentrates all the functionalities,

⁶ <https://www.laurenceanthony.net/software/antconc/>

and even seeing two corpora at the same time via a split screen, in the same software, in an attempt to make it easier for the users.

3.4.2 Quanteda and other script-based tools

3.4.2.1 Scripting environments for language investigation

While the previous section dealt with tools that are created to facilitate the researchers' work via a user-friendly interface, this section deals with the opposite: tools to be used in a scripting environment, which I shall refer to henceforth as script-based tools. The idea with this approach is that researchers develop their tools according to their needs. Some authors (e.g. Biber et al. 1996; Gries 2008, 2010; Weisser 2009) claim that this approach gives more flexibility to an investigation and autonomy to researchers.

Because the scope of this thesis is user-friendly tools, it might seem odd that it should review a tool that requires the user to have, at least, basic knowledge of programming. In terms of usability, the tools in this category are by far the worst. However, there is a tendency for young linguists to start learning programming at the undergraduate level. Moreover, more software libraries specifically designed for language investigations are being developed and made available, especially for programming languages as R and Python. Such libraries present a collection of previously created scripts with functions that are likely to be used often. Good libraries have well-documented and easy-to-understand functions so even users with shallow computational knowledge can benefit from it. Hence, it is useful to study this type of tool.

Quanteda, in my view and to my knowledge, is the easiest programming library that allows user to import and investigate corpora, by relying on Natural Language Processing (NLP) and CL techniques. The functions included in the package are well-

documented and the supporting website⁷, which is constantly updated, includes a guide, references, and examples of its application within the social sciences. As the authors claim

While using `quanteda` requires R programming knowledge, its API [interface] is designed to enable powerful, efficient analysis with a minimum of steps. By emphasising consistent design, furthermore, `quanteda` lowers the barriers to learning and using NLP and quantitative text analysis (Benoit et al. 2018:1)

3.4.2.2 Functionalities

With `Quanteda`, users can perform conventional NLP and CL operations such as segmenting texts by words, sentences and paragraphs; to tokenising texts; stemming words; and retrieving n-grams. It also permits corpus management via metadata: users can filter and subset the corpus according to text-level variables in order to create subcorpora.

One useful aspect of `Quanteda` is its use of dictionaries. With dictionaries, a list of words can be easily searched in a corpus or subcorpora. Users can also use a built-in dictionary (Young & Soroka 2012) to perform sentiment analysis. Sentiment analysis generates an overview of subjective information, such as opinion and sentiments, from a collection of texts (Pozzie et al. 2016). Although sentiment analysis is not widely used in CL, it might be an alternative for corpus investigation when automatic semantic tagging such as USAS (Piao et al. 2016) is not available.

⁷ <https://quanteda.io/>

Other Quanteda functions provide data visualisations that are more related to CL methods. For instance, word frequencies can be plotted as the infamous word clouds (Gambette & Véronis 2010) (figure 3.2)⁸ or as frequency plots (figure 3.3). Word keyness, the extent to which a word is more statistically significantly present in a corpus when contrasted to another corpus of the same size or larger (Baker et al. 2006), can also be plotted by showing or not the corpus used for comparison (figure 3.4). Although users typically understand tables for keyword analysis, this visualisation might make it easier to spot significant differences in frequency. Another useful visualisation is the plot for lexical dispersion, which is discussed in 6.2.3.



Figure 3.2: example of word cloud made with Quanteda

⁸ All Quanteda screenshots were retrieved from <http://quanteda.io/>

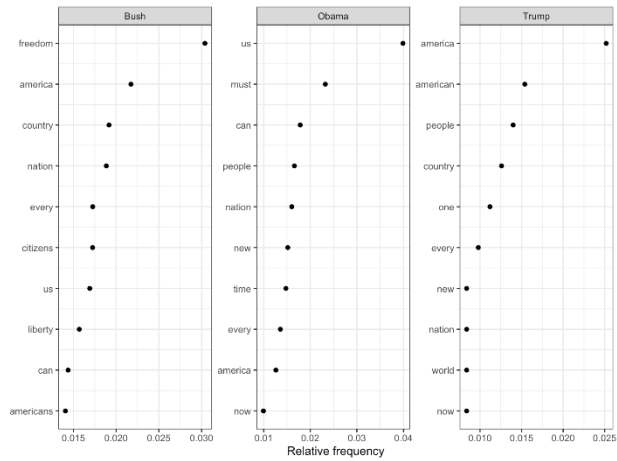


Figure 3.3: example of a frequency pot made with Quanteda

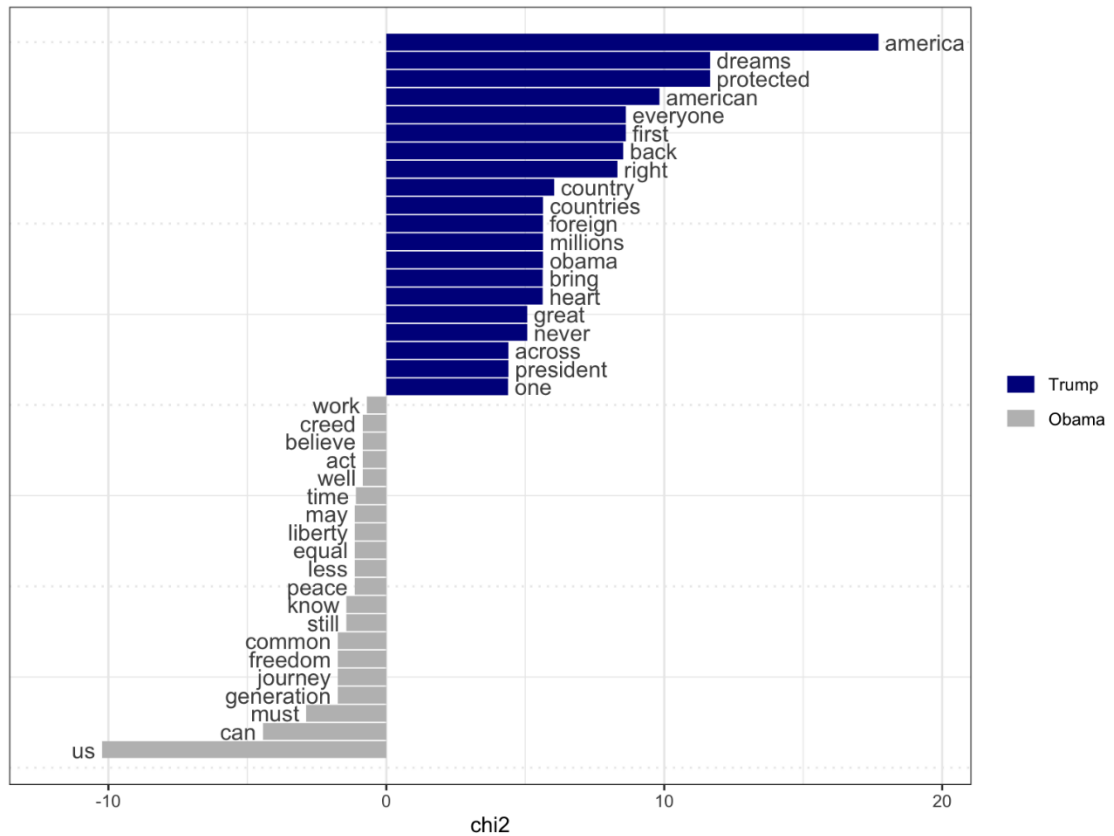


Figure 3.4: example of word keyness plot made with Quanteda

3.4.2.3 Pros and Cons of script-based tools

If used properly, scripting tools can ensure reproducibility. When authors make available the data and scripts used in their study, all the steps taken to achieve the final analysis can be repeated by other researchers. They are also highly customizable. For

example, instead of requiring the researcher to change the data format to fit a tool, the tool can be altered to suit the research or data in question.

Another benefit is that specific statistical models can be applied in the same environment, without the use of a combination of tools. Common scripting languages (e.g. R and Python) are open-source, which contributes to the availability of the packages made for those environments.

Script-based tools are, sometimes, the only option for certain types of analysis. For example, there is a paucity of user-friendly tools in CL that deal with dependency-parsed and constituency-parsed corpora. Script-based tools are the primary resources available for users who want to work with such data.

Despite the advantages of using script-based tools, there are many shortcomings. Wrongly calling a function, for example, can lead to inaccurate statistical calculations. Computer-based software like that discussed in section 3.4.1 has been around for many years and has several users. If something in the calculation of, say, keywords was wrong, someone would likely have found and reported the error. However, if scripts are created for a unique piece of research, even with the support of well-established packages, the chances of having errors and mistakes are high (Peng 2015). Scripts developed for a single piece of research lack the extensive testing and control undergone by well-established software packages.

Another issue, as pointed by Hardie (2012:383) is that such “programs may run slowly if they do not incorporate the ‘tricks’, such as indexing, needed for high speed on large datasets”. That is true for Quanteada. For example, querying the five-million-word corpus described in the previous chapter proved to be a rather slow task. The query system and its syntax, which was not powerful enough, were also an issue.

Simple things, such as searching for a multi-word sequence, are not straightforward. There are ways of using sophisticated corpus systems like the IMS Corpus Workbench (CWB) to scripting languages like Perl and R⁹. Although that approach allows the user to access all the packages in the same environment (R, in the case of Quanteda), it is still far from user friendly.

Finally, script-based tools are not easy for ordinary users. Even if there are plenty of tutorials available, together with pre-made example scripts and proper documentation of the application programming interface (API), it requires much more computer expertise than the standalone CL software.

3.4.3 CQPweb

3.4.3.1 History

CQPweb works on top of a system that was first created nearly three decades ago. As more annotated corpora became available back in the 1990s, it became evident that a system that allowed a precise query was needed. To address this need, Christ (1994:23) proposes a corpus query system that would increase the precision in the way the corpus was investigated and, at the same time, it would reduce the amount of manual browsing necessary, in contrast to the existing query systems at that time. To account for different types of knowledge, the system should also include a general-purpose query language. To deal with limitations of computer hardware, the system should allow corpora to be stored on a more powerful remote computer and be queried

⁹ <https://github.com/PolMine/cwbtools/blob/master/R/cwb.R>

on a computer that would otherwise not be able to process large corpora (Christ 1994:5).

The system proposed by Christ (1994) was the starting point for CWB. It is probably the longest-established software for corpus analysis. It is widely adopted and is used as a back-end engine for other software such as CQPweb and TEITOK (Janssen 2018). It has also served as an inspiration for other software, such as Manatee (Rychlý 2007), which is the back-end for SketchEngine (Kilgarriff et al. 2004).

CWB is also the system upon which BNCweb (Hoffmann & Evert 2006) is built. BNCweb was created to provide a user-friendly querying interface for the heavily annotated BNC 1994 (Hoffmann et al. 2008:25). BNCweb allows users to perform simple queries, like searching for words or word sequences, and returns concordance lines together with normalised and restricted frequencies and the range of occurrences (number of texts with one or more results). Queries can also be easily restricted according to the text metadata, allowing for searches only within a defined subsection. The concordance view can be easily tweaked to have a KWIC or sentence view, to sort the query randomly or in corpus order, to expand the context, or to sort the results (right or left context). Text metadata can also be easily accessed. Previous queries (saved by the user or listed in their history) can be accessed. The default “simple query” syntax makes it easier for users to find words with a particular prefix or suffix; use wildcards; and look for parts-of-speech (POS) and lemmas. The alternative query language is the powerful CQP syntax, which allows fine-grained searches and the retrieval of elaborate structures, such as flexible word sequences or lexico-grammatical patterns potentially making reference to sentence (or any other mark-up) boundaries.

Other functionalities rather than queries are also quickly accessible. Users can view the frequency distribution of the results to a query according to the text metadata. It is also possible to generate collocations with different statistical measures (including MI, MI3, Z-score, T-score, log-likelihood, or the Dice coefficient) (Evert 2005). It is possible to establish markup-based boundaries to the collocation span (across sentence breaks or not); to calculate collocates according to either word form or lemma of the query node's adjacent tokens, select the window span; and set restrictions on the minimum frequency of the collocate and of node/collocate co-occurrence. It is also possible to filter the list for collocates according to their collocations. Subcorpora can be created and compared by generating keyword lists. Users can also categorise a set of concordance lines according to their classification.

Compared to other tools then available, BNCweb offered an intuitive interface that even people without previous CL knowledge can easily use, making it, in the words of Hoffmann and Evert (2006:189), “a user-friendly and feature-rich corpus tool”. Although the unattractive interface might deflect some users, the tool still offers a variety of features that non-expert users can easily learn how to operate. BNCweb, together with WordSmith Tools and AntConc were probably a beginning of what would allow students, seasonal corpus linguists and other adventurous users to use CL tools in order to analyse language, as they still top the list of the most cited corpus tools (see 4.3.2).

3.4.3.2 CQPweb operation and performance

Because CQPweb began as a rewrite from scratch of BNCweb, it inherited the functions of its inspiration. CQPweb was first created as a teaching tool. For this reason, a common scenario for its use is in a classroom with many students performing similar or the same queries on the same corpus, as they work through a

given sequence of same tasks. By virtue of following BNCweb's data management architecture, CQPweb is optimised for that scenario. The system caches data generated by user requests, such as query results and subcorpus frequency lists, so any identical future requests receive a much quicker response. That is, the data processing only needs to be done once, because if the same process is required again, it can be accessed from the cache.

For this and other reasons, when compared to the computer-based tools like AntConc, CQPweb excels in terms of processing speed. As all data processing is done on the server, rather than the users' machine, it can work well, provided internet connection is available. Like any browser-based web application, CQPweb is cross-platform and cross-device. This means users can access a CQPweb server via any operating system (e.g. Windows, Mac, Linux) or even different devices (e.g. phone, tablets, computer) and always obtain the same results. This is different from what happens with AntConc, for instance. Word counts can vary depending on the version of the software and the operating system in use, creating discrepancy in results. This does not occur with CQPweb. Once the corpus is indexed in CWB, the word count will not be affected if the CQPweb version changes. Detailed information on how the indexing process works is given in Christ (1994), Evert and Hardie (2011), and Hardie (2012).

3.4.3.3 Open-source tool

CQPweb is open-source. Being open-source means the software can be continuously edited by other users and consequently having a growing number of features. Being open-source does not mean that changes to this piece of software will be restricted to it. The open-source code can also inspire other pieces of software. For instance, Sketch Engine's (a commercial tool) system, resembles the back-end engine used by CQPweb.

However, for as much as it is ideal to have free sources of knowledge and tools, it might be cheaper for an institution to pay for a more user-friendly tool, than to train staff to use open-source software. One reason for that is that, in most cases, commercial tools are much more user-friendly (Feller et al. 2006).

For instance, an extremely useful option that CQPweb offers is the system administrator account. With this account, the super user can manage corpora and users. That makes CQPweb ideal for sharing corpora online and restrict access in case it is necessary (e.g. due to copyright issues). These adjustments can be made via the browser interface, which makes CQPweb's administration system, to a certain extent, user-friendly. However, it is still a rather difficult task for many. Not to mention the issues that emerge when setting and maintaining the server, which is far beyond the knowledge expected from a linguist. Sketch Engine, on the other hand, offers corpus installation and sharing with a better usability. Hence, if the goal is to have usability not only for the final user, commercial alternatives like Sketch Engine might, ultimately, be a more convenient tool.

3.4.3.4 Online platform: easy sharing and accessing data

Although Sketch Engine allows for easier corpus installation, CQPweb also allows users to upload their own corpus. This has been implemented in a recent version of CQPweb and, as of this writing, is still an experimental feature. It also requires the system administrator to grant permission.

Voyant (Sinclair & Rockwell 2020), a web-based corpus tool, allows user to import their corpora for free without the need to require permission. Although Voyant offers 29 tools and extensive documentation on them, the software has some issues. For instance, uploading corpus data to the server can be quite slow, and the tools available

within Voyant are not necessarily of interest to CL researchers and students. Voyant lacks certain tools commonly used in CL, such as generation of keyword lists. In other cases, a tool may behave like some widely known CL tool, but with a different name. This is the case with the *correlation* function, that measures how significantly two terms in a corpus are related to one another, which of course is all but identical to a collocation analysis.

Online tools like CQPweb play a crucial role for beginners. They not only establish good practice, by providing structured means to explore corpora using well-CL defined techniques; in, but access is easier, as it does not require installation and can be quickly accessed. This is illustrated with the high number of references to and users of the BYU corpora (Davies 2004-).

3.5 Discussion and conclusion

In this chapter, I have discussed a range of different software used to investigate corpora. The main difference among the tools reviewed was on their architecture (i.e. via a web browser, an installation in the user computer, or a scripting environment). A point on which they coincide is that they are all tools that can be used when introducing corpus linguistics to beginners. Quanteda is not user-friendly, as it requires familiarity with programming language. However, it can be easily used as an introductory tool for users interested in learning programming languages to investigate textual data.

Although the goal of making CL tools as user-friendly as possible is valid, it might, sometimes, backfire. For instance, the lengthy corpus importing in #LancsBox and Voyant Tools make the software less attractive to many users. Speed is key, as users want (almost) immediate responses to their requests (see 5.5). Hence, lightweight

tools like AntConc or online tools like BYU corpora are popular among casual users (see 4.4).

Users also want tasks to be done automatically or by default, as is the case with the automatic tagging in #LancsBox, or the setting of a default choice of statistic for collocation generation in all the tools that I have considered in this chapter. As discussed in 3.2.1.4, an ideal tool should allow users to achieve their goals with as few obstacles as possible. Users of CL tools need statistically reliable information on their data, but, in many cases, beginners tend to avoid statistics or not to fully grasp them (see 5.4). One solution to this problem that the tools reviewed here present is to convey statistical information via data visualizations. Although visualizations are supposed to make it easier for all users to interpret the data that they express, this is not always the case (see 5.4). The GraphColl function in #LancsBox, and the plots in Quanteda, can be seen as early steps in the field of data exploration via graphical representation. However, the options available are still not an optimal solution.

Overall, the flexibility of script-based tools negatively impacts usability. However, if a tool is designed to ensure high usability, it is harder to also give users flexibility to customize the software for their specific needs. Hence, an ideal scenario would be to have a tool that has an easy interface and fast access for the ordinary user, but that also allows advanced adjustments of the settings, should a user need it.

CQPweb matches these criteria. Its interface is relatively easy and can be quickly learned by new users, and because it a web-based tool, processing can be quite fast, depending on the server hardware. CQPweb also has the administrator function, that allows flexibility in adjusting the settings. It is also open-source so the code can be edited to accommodate any other possible needs. For these reasons, the new tools

developed in this thesis were implemented within CQPweb, as I will discuss in chapters six and seven.

4 Corpus-based studies: a literature investigation

4.1 Introduction

Developing new tools is not only about studying a new piece of software and imagining new features and functions. It is also about knowing what similar pieces of software are in use, and why and how they are used to accomplish their users' goals. In a commercial environment, this discovery process is often accomplished by means of a marketing survey. In such a survey, potential customers of a product are asked about their favourite products; about their main considerations when choosing a product or service; and so on (Brhel et al. 2015). In an academic context, a survey of users is also possible. For example, Tribble (2006) undertakes such a survey. Or in state-of-the-art surveys, such as Wiechmann and Fuhs (2006) (see 3.2), and Boulton (2012). Boulton deals specifically with Corpus Linguistics (CL) tools used in the language learning environment. With the aim of studying how learners use these tools, he discusses 80 publications from three different journals on data-driven learning from the early 2000s.

Although the aforementioned studies of corpus linguists or language learners as users do provide some useful information, they only address a few programs that were developed about twenty years ago. Any account of software that is more than five years old is almost certainly outdated. Moreover, these studies do not allow us to identify precisely what the most used tools are *at present*, given the immense variety of tools available and the growing number of corpus-based studies being undertaken. One way of listening to a wider public than a small survey group is to investigate how scholars report their corpus-based research.

The aim of this chapter is, then, to identify (i) the corpus linguistic tools most frequently utilised in language research; (ii) why and how these tools are used to address their users' research questions; and (iii) whether and to what extent the ranges of tools used across different subfields of language studies intersect one another. To accomplish this, I carried out a literature investigation encompassing over 5,000 academic articles reporting corpus-based studies (4.2). Section 4.3 describes the process by which I ran queries on two academic databases to retrieve papers reporting on corpus-based research. The resulting compilation of papers is then analysed in section 4.4, in order to arrive at large-scale information on the use of CL software across language studies. Since the compilation constitutes a corpus, and I treat as such in this chapter, my investigation itself is an instance of corpus analysis.

4.2 The databases

4.2.1 Choosing the sources

The articles included in this investigation were retrieved from two different academic databases: the Arts and Humanities Database (AHD)¹⁰ and the Linguistics and Language Behavior Abstracts (LLBA)¹¹. There is not a clear distinction between a database and a search engine in the context of retrieval and index systems for academic publications. For present purposes, an academic database is understood to be any source through which is available a broad and well-documented electronic journal collection.

Although the target audience of this thesis are non-specialist users of corpus data and methods (NSUs) (see 1.1), these two databases encompasses different areas of knowledge and more advanced than the one expected among NSUs. My rationale for a literature investigation probing a diverse set of academic fields is as follows. First, as explained in the introductory chapter of this thesis, the target audience is not wholly restricted to language learning and teaching. This study also considers, as a secondary target audience, which are other language researchers at a beginner level in CL. Hence, it is reasonable also to explore what is used in other fields of language studies.

Second, any preference that might be found for specific tools in certain fields might be arbitrary or no more than a matter of tradition in those fields. It does not necessarily mean that these tools represent the best ones available. Hence, looking into just one

¹⁰ https://about.proquest.com/products-services/Arts_and_Humanities.html

¹¹ <https://proquest.libguides.com/llba>

specific subfield of research would risk failing to represent language studies as a whole due to a bias to that subfield's particular preference. For instance, a tool that *could* be of great use to both lexicographers and language teaching researchers might in practice be restricted to the former group – due purely to the second group's lack of awareness of its existence. Moreover, NSUs are very likely to navigate across fields, as their ultimate aim is to use CL methods in their own endeavours.

Third, software is continuously and quickly changing. New tools can emerge in more technologically advanced research groups before spreading to other researchers and sub-fields. It is therefore worthwhile to allow new trends in different fields of language research to emerge from the investigation of the literature. New and emerging tools would be excluded from the results if a broad range of subfields were not included.

In light of these points, the two databases (AHD and LLBA) used in this investigation were chosen because (i) they are constantly updated, so that recently published articles and the recent advances they report will not be out of scope; (ii) the publications that they index have a high impact, and thus are more likely to represent consensus or common methods within their respective subfields; (iii) and they cover a wide range of research (sub-)fields (see above). Specific reasons for use of these two databases in particular are given below.

4.2.2 LLBA

LLBA features abstracts and indexes from 3,584 different publications, as of the time of the data retrieval¹², and thus meets the criterion of wide coverage (see 4.3.2). This level of diversity can help prevent the results from being skewed towards the preferences and practices of individual subfields or, indeed, journals.

LLBA's diversity concerns not only the variety of journals, but also the country of those journals' origin. The publications encompassed by LLBA come from 98 different countries, most being from North America (45%) or Western Europe (40%).

Another reason for opting for LLBA as a source is its approach to linguistics and language studies. Multiple aspects of language study, such as phonetics, morphology, semantics and syntax, are covered by the database, as well as a wide variety of linguistic fields, such as descriptive, comparative and historical linguistics.

The field of language teaching and learning is also covered. 242 of the journals in LLBA deal specifically with language and education, which represents 9% of the English-medium journals in the database. This includes a number of publications with very high impact, such as *Language Learning* and *Studies in Second Language Acquisition*. Most publications are highly ranked in indexes of scientific research impact such as the Journal Citation Reports¹³ and SCImago Journal Rank¹⁴. This is relevant here not only because it helps focus my analysis on present practice in high-

¹² 3rd February 2018

¹³ <https://clarivate.com/webofsciencegroup/solutions/journal-citation-reports/>

¹⁴ <https://www.scimagojr.com/>

quality research, but also because of the large audience reached by such publications, by definition. Thus, more people are made aware of the software that is used in a paper published in a high impact journal than would be made aware of the software used in an equivalent paper in a lower impact journal. Also, high quality publications can be an indication of good performance of the software, as software quality has an enormous impact on research.

4.2.3 AHD

While the LLBA was selected because of its extensive coverage of the field of linguistics, the interdisciplinary scope of AHD was a key criterion in opting for this latter database. The topics covered in its approximately 400 peer-reviewed titles (437 at the time of collection) vary greatly and are spread across the following main subjects, summarised from the AHD website's documentation pages:

- Art, design, crafts and photography
- Archaeology, anthropology and classical studies
- Architecture, interior design and urban planning
- History, philosophy, geography and religion
- Modern languages and literatures
- Music, theatre, film and cultural studies

Although many of these subjects are only loosely associated with language, the growth of interdisciplinary studies makes these publications of interest for purposes of this investigation. Studies using large language datasets are no longer restricted to CL.

Taking into consideration how researchers outside CL undertake computer-based textual analysis might well point the way to new paths for CL.

Although LLBA and AHD complement each other, there is an overlap of journals; thus, many articles are listed in both indexes. To address this issue, after I compiled my dataset via the procedures to be detailed in 4.3.1, I eliminated any duplicate articles from the dataset.

4.3 Procedures

4.3.1 Article retrieval and processing

The first step was to establish how to retrieve the relevant articles from each database. To do so, I *queried* the database. In this scenario, querying a database means using a set of conditions to filter the entire database and retrieve just a selected part of it. Performing such a query requires a choice to be made of which words to include in the search. Many different terms can be used to describe corpus-based research. Therefore, no single search term can retrieve all journal articles using CL methodologies. To achieve a broad representation of the fields or sub-fields covered by the database at hand, I used key words listed in articles that I was certain to feature corpus-related research.

The query “*corpus tool*” OR “*corpus software*” OR “*corpus method*” OR “*corpus-based*” was used with both databases, with additional restrictions placed on the search so that only articles in scholarly journals published in English and no more than three years prior to the point of data collection (2015 to 2017) would be retrieved. The search retrieved 1,228 articles from the ADH and 1,234 articles from the LLBA.

All articles were downloaded as plain text. In cases where Lancaster University's library did not have access to the full text of the article, the abstract alone was retrieved at this stage. The data was formatted into XML, with each file's header containing: the text identifier; the article title; the journal name; the publication's subject area(s), such as literature or education; the country of the journal; the year the article was published; and the database from which it was retrieved. More extensive metadata was preserved externally. This includes the digital object identifier (DOI), volume and issue, and author affiliation of each of the articles.

After collecting and preparing this data, a search for duplicate texts was performed. Among the original 2,462 texts, 145 were duplicates. One copy of each duplicate pair was removed from the collection, and in the remaining copy, the metadata record of the article's source was updated to note its presence both databases. After the duplicates were removed, the corpus consisted of 2,317 texts, totalling 4,875,535 words.

4.3.2 Investigation Methods

4.3.2.1 Tools and methods

Because of the substantial size of this dataset, I used CL methods for its analysis. I analysed the data using the open source software environment and programming language R¹⁵ and *Quanteda* (Benoit et al. 2018), a text analysis package (see 3.4.2). I opted to use a scripting environment so that the procedures could be as reproducible as possible. Looking ahead, it seems like that it might be desirable to repeat the present

¹⁵ <https://www.r-project.org/>

analysis at a later date for a subsequent timespan. Exact comparability of data and methods would be required for such a future analysis; doing the current analysis with scripts assures that.

This investigation was divided into three steps: (i) *identifying* software mentioned in the articles; (ii) investigating *in which contexts* these tools are being used; and (iii) understanding *how* these tools *are applied* in language research.

4.3.2.2 Identifying the tools

4.3.2.2.1 Pre-existing lists

Many sources attempt to keep track of the growing range of CL software tools. For this study, I used a combination of the constantly updated CL software lists from The Linguist List¹⁶, Martin Weisser's website¹⁷ and Corpus-Analysis' research centre¹⁸. However, considering the speed at which new pieces of software are developed, as well as the diversity of the fields that now make use of computer-assisted text analysis, these lists are likely to leave at least some tools out at any given point. For instance, Poliqarp¹⁹, a corpus processing program, is not included in any of the three lists mentioned above – but it *is* mentioned in research articles within the corpus. Moreover, many of the pieces of software mentioned in lists such as these turn out, on further investigation, to have fallen out of use. In some cases, listed hyperlinks are

¹⁶<http://linguistlist.org/sp/SearchWRListing-action.cfm?subclassid=7223&SearchType=LF&WRTypeID=2>

¹⁷ http://martinweisser.org/corpora_site/CBLLinks.html

¹⁸ <https://www.corpus-analysis.com/>

¹⁹ <http://poliqarp.sourceforge.net/>

broken and do not lead to an existing tool, as in the case of WConcord 3.0, listed by the corpus-analysis website. This might be a consequence of the software being discontinued by the developer; or simply losing popularity among researchers.

Hence, the first step was to identify whether the tools culled from the aforementioned lists were named in the corpus. Although not being mentioned in any of the more than 2,000 articles in the dataset does not guarantee that a tool is no longer in use at all, it *is* a strong indication that the software is not currently popular among researchers. Each name of a tool from the three lists mentioned above was queried in the corpus. In cases of a single piece of software being referred to in different ways, the alternative names were also queried. For instance, the online interface to the Corpus of Contemporary American English (Davies 2008-) is found in the dataset under different names including COCA and corpus.byu.edu.

Of the 277 pieces of software present across the three lists, 49 were mentioned by name in the dataset. Most of the named tools which were not found in the dataset were no longer available for download (e.g. Concordance Software²⁰); had not received any recent updates (e.g. ParaConc²¹); or simply could not be accessed due to disappearance of the online presence linked by the list in question (e.g. Multilingual Corpus Toolkit²²). A further investigation showed that 16 out of the 49 names found were false positives. For instance, *Amalgam* and *Flair* only occurred with their traditional word meanings, rather than as the tool names. Hence, from the original 277

²⁰ <http://www.concordancesoftware.co.uk/>

²¹ <http://paraconc.com/>

²² <https://sites.google.com/site/scottpiaosite>

tools mentioned in the existing lists, only 32 (table 4.1) were mentioned at least once in the corpus.

ANNIS	CQPweb	Mallet	Tred
AntConc	Dart	Maltoptimizer	UAM
BYU	DocuScope	Maltparser	VocabProfiler
CasualConc	Elan	Monocon	Voyant Tools
CLAWS	FrameNet	ngrams	VU Corpus
COCA	Gephi	Pie	Wmatrix
Coh-metrix	GraphColl	Praat	WordSmith
Compleat	LDA	Sketch Engine	Wordstat

Table 4.1: tools from existing lists found in the corpus

Many of the tools in the lists are not corpus analysis systems, but rather are tools for other purposes (such as automated text annotation) or are language knowledge resources for use in the development and operation of such tools. An example of the latter is Framenet²³, a dictionary database in which the words are tagged for semantic roles. However, because such tools and resources are often used in association with corpus tools in the strict sense that is relevant to my concerns, I opted to retain on the list of search terms any piece of software related to corpus investigation in any way.

²³ <https://framenet.icsi.berkeley.edu/fndrupal/>

4.3.2.2.2 Collocations

In order to identify pieces of software other than those on the list above, I looked at collocates of with terms possibly related to the use of software. The search terms used as node were: *software*; *tool(s)*; *program*; *corpus*; *corpora*; *corpus-based*; *method*. A list of the top 200 collocations was generated for each. The cut-off of 200 was determined from a first trial search, which suggested that relevant results would appear only above this threshold. The collocate lists were merged into one, with any collocates on more than one list highlighted, as I expected that they were more likely to be associated with a tool name. A concordance of each collocate expected to be the name of a tool was examined to verify that this was indeed the case. 62 additional names of tools were identified in this way.

Table 4.2 displays the CL tools and table 4.3 shows the CL-related software. The tools were manually separated into two tables for clarity. Also for clarity, CL tools and their absolute frequency are shown in figure 4.1. The word *algorithms*, although it does not refer to a specific tool, was included in the final list. This is because the word was frequently used when the authors created their own scripts, as in the example below:

Once we had gathered this data into a plain text flat file, we used Python code – more specifically the algorithms contained in the Python NetworkX library – to analyze the network. (Text 1287: Ahnert & Ahnert 2015)

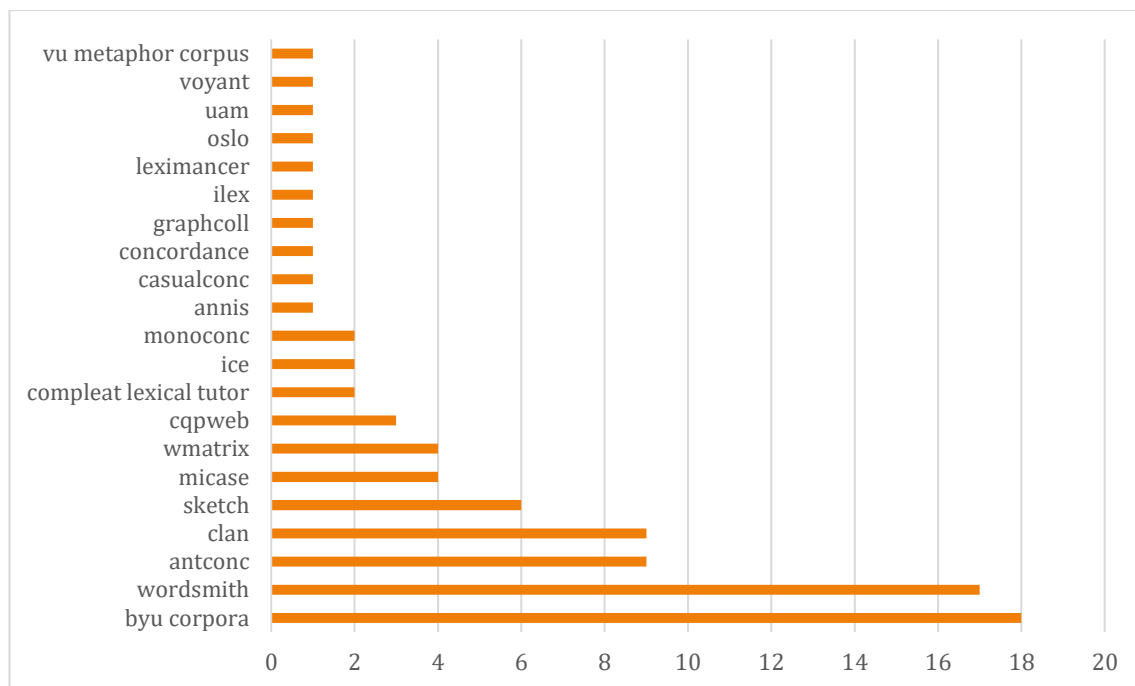


Figure 4.1: overall frequency of CL software

	web	comp.	script	OS	free.	comm.	conc.	other
ANNIS	yes	no	no	yes	no	no	yes	no
AntConc	no	yes	no	no	yes	no	yes	no
BYU Corpora	yes	no	no	no	yes	no	yes	no
CasualConc	no	yes	no	no	yes	no	yes	no
CLAN	no	yes	no	yes	no	no	yes	no
concordance	no	yes	no	no	no	yes	yes	no
Corpus Tools	no	yes	no	no	yes	no	no	yes
CQPweb	yes	no	no	yes	no	no	yes	no
GraphColl	no	yes	no	no	yes	no	yes	no
ICE	yes	no	no	no	yes	no	yes	no
ICLE	no	yes	no	no	no	yes	yes	no
iLex	no	yes	no	no	yes	no	no	yes
Lexical Tutor	yes	no	no	no	yes	no	yes	no
Leximancer	yes	no	no	no	no	yes	yes	no
MICASE	yes	no	no	no	yes	no	yes	no
MICUSP	yes	no	no	no	yes	no	yes	no
MonoConc	no	yes	no	no	no	yes	yes	no
OSLO	yes	no	no	no	yes	no	yes	no
Poliqarp	yes	no	no	yes	no	no	yes	no
Sketch Engine	yes	no	no	no	no	yes	yes	no
UAM	no	yes	no	no	yes	no	no	yes
Voyant	yes	no	no	yes	no	no	yes	no
VU	yes	no	no	no	yes	no	yes	no
Wmatrix	yes	no	no	no	no	yes	no	yes
WordSmith	no	yes	no	no	no	yes	yes	no

Table 4.2: CL tools found in the corpus and their categories

	web	comp.	script	OS	free.	comm.	conc.	other
algorithms	no	no	yes	yes	no	no	yes	no
ALICE	no	yes	no	no	yes	no	no	yes
cancode	no	yes	no	no	no	yes	yes	no
celex	no	yes	no	no	no	yes	no	yes
CHILDES	yes	yes	no	no	yes	no	no	yes
claws	yes	no	yes	no	no	yes	no	yes
COBUILD	no	yes	no	no	no	yes	no	yes
coh-metrix	yes	yes	no	no	no	yes	no	yes
dart	no	yes	no	no	yes	no	no	yes
docuscope	no	yes	no	no	yes	no	no	yes
ELAN	no	yes	no	no	yes	no	no	yes
FrameNet	yes	no	yes	no	yes	no	no	yes
GDEX	yes	no	no	no	no	yes	no	yes
gephi	no	yes	no	yes	no	no	no	yes
GraphPad	no	yes	no	no	no	yes	no	yes
LDA	no	yes	no	no	yes	no	no	yes
LENA	no	yes	no	no	no	yes	no	yes
mallet	no	no	yes	yes	no	no	no	yes
maltoptimizer	no	no	yes	yes	no	no	no	yes
maltparser	no	no	yes	yes	no	no	no	yes
ngrams	yes	no	no	no	yes	no	no	yes
praat	no	yes	no	yes	no	no	no	yes
QDA	no	yes	no	no	no	yes	no	yes
R	no	no	yes	yes	no	no	no	yes
Rbrul	no	no	yes	yes	no	no	no	yes
SALT	no	yes	no	no	no	yes	no	yes
Tlex	no	yes	no	no	no	yes	no	yes
tred	no	yes	no	yes	no	no	no	yes
VocabProfiler	yes	no	no	no	yes	no	yes	no
WEKA	no	yes	yes	yes	no	no	no	yes
WordGen	no	yes	no	no	yes	no	no	yes
wordnet	yes	no	yes	yes	no	no	no	yes
wordstat	no	yes	no	no	no	yes	no	yes

Table 4.3: CL-related software found in the corpus

4.3.2.2.3 Investigating the context and the type

The next steps consisted in verifying if their usage varied across research fields; and if certain tools' characteristics prevailed. To identify these characteristics, the tools were classified according to the following criteria (created for the purpose of this analysis): system-based; pricing; and main function. *System-based* refers to how the tool is accessed: via web browsers (web-based); via a locally-installed application (computer-based); or via scripts of some programming language (script-based). As some programs are accessible in more than one way, this characteristic can have more than one value. *Pricing* captures whether the piece of software is paid-for

(commercial), free to use (freeware) or free to use and edit (open source). *Functionality* captures whether the tool works as a concordancer, even if other functions are present, or if it targets another function or functions, such as corpus annotation or alignment.

It is worth noting that, in many cases, a particular corpus and the tool used to access it had the same name (e.g. MICASE, MICUSP). In all such cases, the characteristics of the tool rather than the corpus are considered.

The article metadata recorded the subfield of research of each corpus text. This information was used to identify the (sub)field of the article in which each mention of a tool appeared. These were classified as one of the following nine subfields: *Anthropology, Sociology and Philosophy; Arts; Computer Applications; Education; Humanities; Linguistics; Literature; Psychology and Psychiatry; and Social Sciences*. These classifications came from the original article subject descriptors in the databases; there were in total 77 different descriptors. As such a fine-grained classification would not yield meaningful results for the relatively small numbers in question, I opted to group them more broadly. For instance, articles with the subject field as any of “Computers--Microcomputers, Linguistics, Computers--Computer Assisted Instruction, Computers--Personal Computers” or “Computers--Internet, Linguistics” were all assigned to the category “*Computer Applications*”. Table 4.4 shows the absolute frequency across fields of software used for concordances and other types corpus exploration. Figures 4.2 to 4.4 show the distribution in percentage of types of tools (including those which are not CL tools in the strict sense) across fields.

	ASP	Arts	Comp.	Edu.	Hum.	Ling.	Lit.	Pysch.	SocSci
annis	0	0	1	0	0	0	0	0	0
antconc	1	0	0	1	0	5	2	0	0
casualconc	0	0	0	1	0	0	0	0	0
clan	2	0	0	1	0	1	0	5	0
lexical tutor	0	0	0	1	0	0	1	0	0
concordance	0	0	0	0	0	0	1	0	0
cqpweb	1	0	0	0	0	1	1	0	0
byu corpora	1	0	1	4	1	9	0	2	0
graphcoll	0	0	0	0	0	1	0	0	0
ice	0	0	0	1	0	1	0	0	0
ilex	1	0	0	0	0	0	0	0	0
leximancer	0	0	0	0	0	1	0	0	0
micase	0	0	1	3	0	0	0	0	0
monoconc	1	0	0	0	0	1	0	0	0
oslo	0	0	0	0	0	1	0	0	0
sketch	1	0	0	1	0	3	1	0	0
uam	1	0	0	0	0	0	0	0	0
voyant	0	0	0	0	0	0	1	0	0
vu	0	0	1	0	0	0	0	0	0
wmatrix	1	0	0	0	1	2	0	0	0
wordsmith	1	0	1	2	2	9	1	1	0

Table 4.4: absolute frequency of CL software mentioned across fields

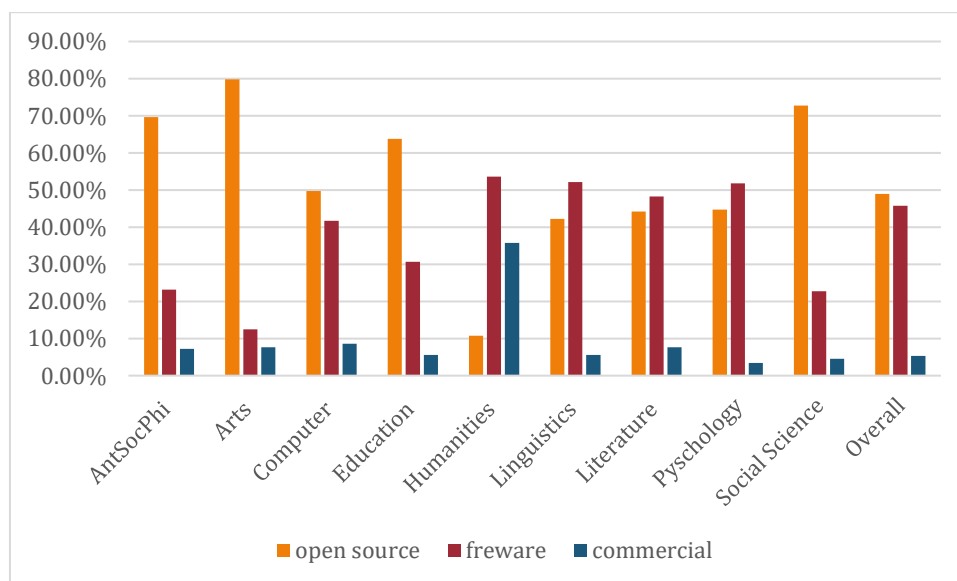


Figure 4.2: percentage of software type (pricing) across fields

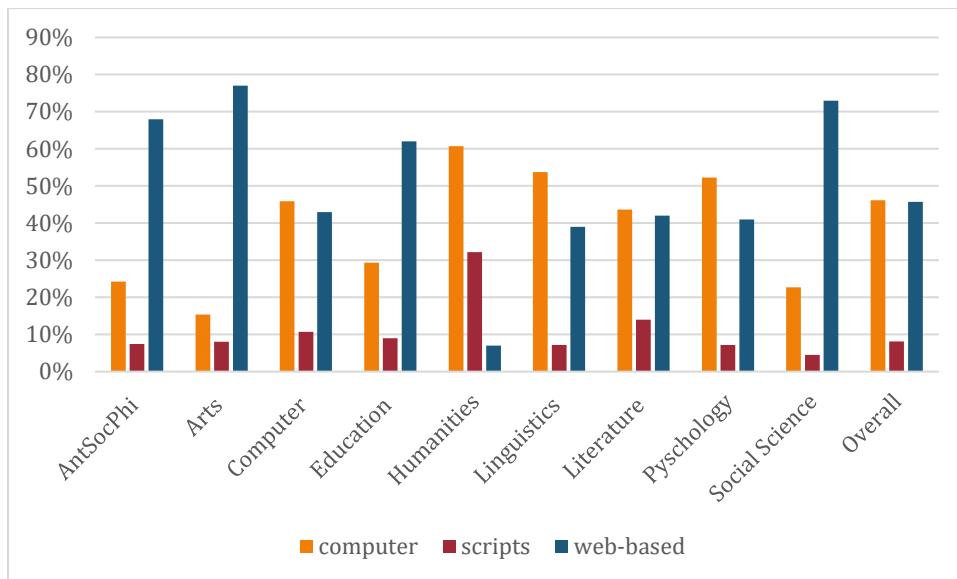


Figure 4.3: percentage of software type (environment) across fields

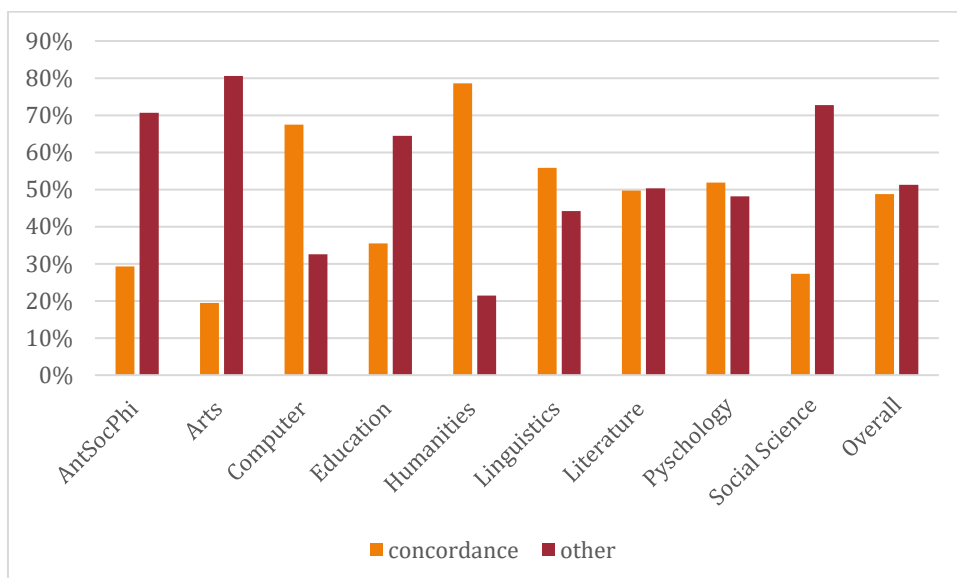


Figure 4.4: percentage of software type (functionality) across fields

4.3.2.3 A closer look at the tools: understanding the usage via concordance lines

The frequency data of mentions tool mentioned are estimates, due to noise in the underlying data. It is not possible to get precise frequencies (which could be higher) without extensive manual filtering, for a number of reasons. Sometimes the same tool is referred to by more than one term (e.g., the *IMS Corpus Workbench* is also known as *CWB*). Some tool names are ambiguous with existing English words (e.g. CLAN,

CLAWS and SALT). Encoding issues can obscure mentions that ought to have been retrieved, the text of the corpus having been converted from PDF. PDF generation (or, alternatively, automatic text extraction from PDF) may change the underlying characters for presentation reasons, changing the string representing a tool name; a search for the term *MonoConc* fails to find the following example, in which some other character has replaced the second 'o' in MonoConc::

[...](2013), include the ' AntConc', the ' Word Smith Tools' and' MonÂ°
Conc Pro'. (Text 1681: Jaeger 2015)

Apart from preventing issues like the one above, a closer look at the concordance lines for all the tools mentioned in the previous section, helped me have a better understanding of how the tools were being used and in which specific area of research.

4.4 Analysis

The tables in the previous section were used only as a reference to explore in more depth the right articles and concordance lines. The figures are treated as indicative, not as an accurate report on how frequent the tools are used across fields of research. The goal with relying on corpus methods was to be able to investigate a high number of publications in an optimal time span with enough evidence to support the claims.

4.4.1 Difficulties in finding mentions of the tools

The methods described in the previous section were useful in finding mentions of tools in the articles. However, in many articles, the authors did not state what software they used for their textual exploration. For example, several articles in the area of language acquisition describe thoroughly their method, including extensive

description of the recording and transcription system, but not of the means by which they analysed the corpus, as in the example below.

much as possible quiet room). Recordings were made directly on a laptop using Audacity, v. 2.0.4 as the recording software, set at 44,100 sampling rate, and a Blue Yeti USB microphone set at cardioid direction. (Text 1264: Baltazani & Kainada 2015)

When this phenomenon is observed, it is likely that the data has been analysed either with statistical tools and scripts, or by hand and eye – that is, by the analyst going through the entire data. The two following examples illustrates cases in which the corpus analysis was undertaken without the assistance of any tool.

the output was manually searched for target structures. Both restrictive and non-restrictive RCs were included in our analysis, and no distinction was (Text 1661: Kirjavainen et al. 2017)

To get an initial sense of the response quality, we inspected the data manually for possible fake responses by searching if any participant had given the same responses (Text 1593: Gladkova et al. 2016)

When an article does explicitly name the tool, there are still some issues. In many cases, a tool is cited via footnote, rather than the standard academic citation style, with author and year. This is particularly the case with web-based tools, such as the BYU Corpora (Davies, 2002-). Although most software authors provide the proper citation at their program's website, in many cases the only reference provided in the article using the program is the web address of that site (in a footnote or directly in the text).

However, many of the online platforms available today, such as the BYU site (<http://corpus.byu.edu/corpora.asp>), MICASE (<http://micase.elicorpora.info/>) Compleat Lexical Tutor (<http://www.lextutor.ca/>), or the Sketch Engine (<http://www.sketchengine.co.uk/>) (Text 1407: Gilmore 2015)

When URLs are given, we often observe that the link to the tool does not lead – or rather no longer leads – to an existing page. Some such broken links were for software that was mentioned in a paper only for historical purposes, and not because the research reported by the paper actually used that software. This is true, the case, for example, for *Drexel*, *Concordance Generator* and *Discon* in the example below:

were very slow. Examples include the 'Drexel Concordance Programme'; the 'Concordance Generator' and 'Discon'. The second generation constitutes corpus tools that were introduced between 1980 and 1990. Like the first generation (Text 1681: Mazibuko & Ndebele 2017)

But there are, equally, other cases where the program with the broken-link reference was used in the research being reported.

In yet other cases, the name of a tool was given, but no web address was provided. In such cases, I used Google and other web-search engines to attempt to find out more – but I would usually not be successful (this was the case for DepCluster, among others). The lack of any presence online for a piece of software might reflect a situation where that software was probably designed to be used only within the authors' research centre rather than being made publicly available.

4.4.2 Tool cost and availability

Most of the tools whose links that were no longer accessible were mentioned by articles published in 2015 (the first year covered by the corpus). More recently developed tools are mainly open-source (e.g. Quantedata) or freeware (e.g. GraphColl/#LancsBox), while older ones are more likely to be paid-for (WordSmith Tools). There is a tendency towards making tools available, whether via an openly accessible server instance for web-based access, or of the source code via some code repository. Many were found in Github and other open code repositories, including for instance Voyant Tools (Sinclair & Rockwell 2020). This phenomenon of sharing software scripts online might be a reflection of the increasing access to computational resources; the emphasis that funding bodies have placed on software development and enhancement; the increasing concern of the academic community with research reproducibility; and the need for or interest in sharing corpora online. ShinyConc,²⁴ is an example open-source package that helps users to create customized web-based concordancers.

Of the (still high number of) paid-for tools mentioned in post-2015 articles, most are not CL tools in the strictest sense. Rather, they are mainly additional applications used for statistics, for instance (e.g. GraphPad) – not for searching or processing of the actual corpus data. Some exceptions are Sketch Engine and WordSmith Tools, which are paid-for CL tools with a high number of mentions.

²⁴ <http://shinyconc.de/>

4.4.3 More than concordance lines

As mentioned in the literature review chapter, originally the main, or sometimes only, function of a corpus tool was to undertake a corpus query and then display the results as a concordance. However, more recent tools offer new functions, the need for which is evident in many in the articles in the corpus.

4.4.3.1 Query syntax, spelling variation and text formatting

One concern that was recurrent in the corpus was how to address issues of searching for patterns in the corpus and retrieving the expected results. Tools with either sophisticated or simplified query language are often mentioned. The Sketch Engine query language, the Corpus Query Language (CQL), is an example of sophisticated query system mentioned in the databases. Advanced query languages such as the CQL and the Corpus Query Processor (CQP) (see 3.4.3) are exceptional means of refining searches. They are efficient in getting as many as possible desirable occurrences without also obtaining a high number of false positive results (improving precision and recall, in the terminology of information retrieval. These systems can also come in handy when a user wants to search for a word that has more than one possible spelling. However, a corpus might also be pre-processed to standardize spelling variation, using a tool such as VariAnt or Vard. These are not concordancers. However, the existence of such tools, and of articles discussing the problem of spelling variation, indicates that it would be useful for users of the type represented by the articles' authors if software for corpus analysis were able to account for different spellings. Another issue that also affects the behaviour of corpus tools is the different encoding (see 3.2.1). Some studies report the need to prepare the data with text encoding formatting tools like SALT and SarAnt, a simplified regular expression system that allows users to search and replace (sequences of) characters.

4.4.3.2 Visualization: data summary and multimodal data

Another trend that I observed is the emergence of tools that provide visualization of the quantitative data, such as Voyant Tools, GraphColl (#LancsBox), Leximancer and Casualconc. These recent tools are, however, only named in the corpus within papers by their creators describing the release of the tools. Hence, it was not possible to identify how these new tools are being used.

I did not find any evidence in the corpus for discussion of new tools with support for the visualization of multimodal data. Rather, it seems, audio and video data is mainly approached using Elan and Praat. This suggests an increasing amount of research that requires tools to deal with videos and audio, rather than only textual data. One example of that is the launching paper of iLex, a tool for sign language.

4.4.3.3 Metadata and annotation

Many articles refer to the creation of subcorpora and retrieval of text metadata. This is especially evident in articles in sociolinguistics or research with spoken data. Annotation tools were also mentioned frequently. Tools like SALTO, Spre, tagant (automatic) and WorldBuilder System (online collaboration) demonstrates the need for corpus tools that handle text annotation and metadata well.

4.4.4 Complementing CL methods

Other tools that are not strictly linguistic software are reported in the corpus as being used in combination with text analysis. For statistics and visualization, the articles in the corpus made mention of using tools such as Goldvarb, Rbrul and GraphPad. In the last year of the corpus I observe a rise in the frequency of mentions of geo-location, especially driven by articles where geographical software is used to map and display

linguistic variation. Tools of this kind that are mentioned in the corpus include BatchGeo, WebLicht, Wordstat.

Possibly driven by the increase in media studies, new tools for topic modelling and web crawling were found. Tools used for crawling the web are Spiderling; BootCat; and FireAnt. For topic modelling the tools found were Mallet and the Stanford Topic Modeling Toolbox.

4.5 Discussion

This chapter has identified 21 pieces of CL software in the strict sense (figure 4.1), of which BYU corpora, WordSmith Tools, and AntConc were the most frequently mentioned programs in a corpus of recent academic publications. The numbers were not impressive, but this is mainly due to the fact that many authors do not refer to the CL software used in their research. Despite this limitation, the study gave an overview of CL tools usage across different fields. Linguistics and Education were the field in which highest number of mentions for CL tools, BYU corpora being the most used. This preference might be due to the easy online access to a range of corpora (see 5.5). WordSmith Tools is mentioned in almost all fields, indicating its versatility. Different fields are now converging in that all exhibit a strong preference for tools that are available at no cost, that does not require programming language knowledge and that deal well with metadata and annotation.

5 Target audience: contextual design and usability

5.1 Introduction

In this chapter I will present an investigation of how users interact with Corpus Linguistics (CL) software. Section 5.2 presents an overview discussing the advantages of observing (as well as its means) users of CL tools. Section two describes the method I used for the present investigation. In section four, I present alternative observation methods that I used to complement the investigation. I discuss the findings in section five. The final section summarises the chapter.

5.2 Overview

In this chapter I will perform a closer qualitative analysis of users of CL software. Talking with and observing users in their own environment can bring several benefits for software designers, as will be shown in 5.3. It can reveal information that users might not be actively aware of, or might not consider relevant, although the developer would. Observation can generate insights on matters such as users' reaction.

In this chapter, I utilise a blend of different approaches to user-experience and human-computer interaction with the aim of better understanding users of CL software and identifying their main needs. In these approaches, the principle factors taken into consideration are users' attitudes and reactions towards the software, such as their satisfaction and their assessment of its learnability. I designed a three-step approach, based on Hartson & Pyla (2012), with which the steps focus on, respectively: *contextual inquiry*; *contextual analysis*; and *design-informing model*. These concepts are all part of the *Contextual Design* paradigm of software development, which I will now briefly introduce, before addressing the three steps in detail in 5.3.2 to 5.3.4. I chose Hartson and Pyla's model for its vast application and relevance in the user-experience studies (e.g. Zahidi et al. 2014; Franklin 2013).

5.3 The Contextual Design Approach

5.3.1 Contextual design

The present investigation utilises the contextual design approach to software design and development, which is

a structured, well-defined user-centered design process that provides methods to collect data about users in the field, interpret and consolidate that data in a structured way, use the data to create and prototype product and service concepts, and iteratively test and refine those concepts with users (Holtzblatt & Beyer 2014:137)

An issue considered when designing the method for this present study, was the need to choose between observation on the one hand, and interviewing or surveying on the other. Both kinds of procedure have both advantages and downsides. While observation has benefits such as witnessing user habits, it does not capture issues that

do not emerge at the moment of observation. Moreover, it can also influence participants' behaviour due to the *observer effect* (Hartson & Pyla 2012). As for interviews and surveying, it is worth noting that users' behaviour can dramatically differ from how they describe their work (Simonsen & Kensing 1997). However, interview and surveys can reveal users' inner responses otherwise not easily spotted.

5.3.2 Contextual inquiry

There are several different approaches adopted at the beginning of user experience (UX) research, such as focus group discussions and usability testing. Focus group discussions present structured interviews for a set of people simultaneously. Because this setting saves time, it is ideal when quick responses are needed. However, what people say and do often differ. For this reason, usability testing comes in handy. Instead of asking users what they want, the researcher observes them while completing a given task using a specific tool (Kuniavsky 2003). Within the contextual design approach, a similar approach to usability testing is a *contextual inquiry*, which is

an early system or product UX lifecycle activity to gather detailed descriptions of customer or user work practice for the purpose of understanding work activities and underlying rationale. The goal of contextual inquiry is to improve work practice and construct and/or improve system designs to support it. Contextual inquiry includes both interviews of customers and users and observations of work practice occurring in its real-world context. (Hartson & Pyla 2012:89)

Within contextual inquiry, different approaches can be adopted. Hartson and Pyla (2012) highlight the difference between *data-driven* and *model-driven* approaches.

The former is indicated when bias in the data collection needs to be avoided. In this case, the inquiry is conducted without predefined categories, and the data that is gathered is itself used to guide further analysis. In the model-driven approach, the processes of data collection and the analytic procedures and categories to be used in its interpretation are designed on the basis of the researcher's knowledge and experience (that is, their existing *model* of the context). This mode of research risks missing certain findings due to the bias necessarily introduced by the predefined analytic categories. But while risking bias in this way, the model-driven approach improves the efficiency of the overall inquiry.

For this study, I adopted a data-driven approach to contextual inquiry. The advantage of using data-driven contextual inquiry is that no predefined framework interferes with direct engagement with the participants' behaviour and responses, which can thus be analysed in a 'bottom-up' manner that does not demand a comprehensive prior understanding of the possible observations in the context at hand.

5.3.2.1 Participants

In this study, it was essential to identify the main features across different tools. Due to the quantitative character of the initial informal studies, this part of the process did not require a high number of participants. There is not much consensus on the appropriate number of participants in a contextual design. While Nielsen (2000) states that five is a good number, Spool and Schroeder (2001) claim that five is not sufficient. Other authorities note that it is more valuable for a Contextual Design study to encompass a broad range of tasks and user backgrounds than to maximize the number of participants (Lindgaard & Chattratichart 2007). On that basis, in this study, I placed greater priority on having participants from a variety of backgrounds over having a very large quantity of participants.

As explained in 3.5, the ultimate goal of this thesis is the development of new tools for CQPweb, a web-based program. Users of CQPweb might have only basic CL knowledge and be able to use the most straightforward affordances of a public CQPweb system via the web, or they might have more expertise and be able to build their own corpora or run CQPweb on their own computer. For this reason, the choice of initial participants was driven by the different work roles that a CQPweb user can adopt.

Computer experience (with emphasis on CL software)	<p>novice: may know application domain but not specifics of the application</p> <p>intermittent user: uses several systems from time to time; knows application domain but not details of different applications</p> <p>experienced user: “power” user, uses application frequently and knows both application and task domain very well (Hartson & Pyla 2012:192)</p>
Research Field	<p>language teaching</p> <p>discourse analysis</p> <p>language description</p> <p>...</p>
Career stage	<p>undergraduate student</p> <p>graduate student</p> <p>researcher</p> <p>lecturer</p>

Table 5.1: user classes

The user group of central concern in this thesis is non-specialist users of corpus data and methods (NSUs; see 1.1). However, including users of different backgrounds in this chapter’s investigation may well offer insights that would not be achieved if *only* raw beginners – who do not yet know what they do not know – were included. Table 5.1 outlines the full set of parameters taken into consideration when recruiting the

participants, most of which parameters relate to different areas of prior specialist knowledge. Table 5.2 displays information on the selected participants and their classification according to the categories presented in table 5.1.

Code	Occupation	CL experience	Software used...	for...
DA	lecturer	intermittent	CQPweb	discourse analysis
DB	researcher	intermittent	Sketch Engine	language description
PA	graduate student	novice	AntConc	discourse analysis
TA	lecturer	experienced	CQPweb	teaching
TB	lecturer	novice	BYU	teaching

Table 5.2: profiles of the five participants in the contextual inquiry

5.3.2.2 Procedures

According to the rationale discussed above, I adopted a data-driven approach without predefined survey/interview questions. I asked the participants to perform a task using their preferred corpus, using the *think aloud* technique. In this technique, the users perform a task and speak aloud about the steps they take and their reasons for that, simultaneously with actually doing the tasks.

Although my main focus is not interface design, I also took users' reaction to the interface into consideration due to the importance of aesthetics in enhancing cognition (Kirk 2012). Aesthetic impressions are highly associated with user acceptance, satisfaction and quality perception (Hassenzahl 2004, Lindgaard & Dudek 2003). When using a corpus tool, the search process itself is not the primary goal of the user, but one of the means used to achieve it. For this reason, the distractions and interferences during this process should be minimised.

The interviews were conducted using the software TeamViewer²⁵. This is an application for recording a shared computer screen and associated audio during a computer to computer call. The intent was to make the interview as non-invasive as possible. Since participants were requested to share their views freely on their preferred CL software, non-invasiveness is important. The participants' PC microphone audio and screen were recorded for further analysis.

Before the beginning of the interview, the intent and format of the experiment is explained. The intent is described to the participant as getting insights on how to improve CL software and identify possible new tools to be developed. The instructions given are simple. The participants are instructed to choose any activity they usually do when using CL methods and to demonstrate and narrate the process. Once the participant was aware of those two points, the experiment begins. The users were expected to work through their activity and share their impressions on corpus tools without any further prompt. However, in case the task did not flow naturally, one or more of the questions in table 5.3 were used as prompts (depending on the stage of the experiment).

Once the interview was over, the recording was transcribed. Five different participants were interviewed. All five participants were anonymised (by using identity codes instead of names); the following metadata, as provided by the participants, were recorded: current career level occupation, experience with corpus linguistics tools, software used and research area (see table 5.2).

²⁵ <https://www.teamviewer.com>

Which corpus tool do you most often use?
Can you please tell me and show me how you do your work? <ul style="list-style-type: none"> - What actions do you take? - Which corpora do you use? - Can you please demonstrate what you do and narrate it with stories of what works, what doesn't work, how things can go wrong, and so on?
How many hours a day do you use the tool?
In which device/operating system/web browser do you usually use corpus tools?
Is your data stored locally or in the cloud?
Can you walk me through a couple of experiences you have had using this tool?
What do you like most about this tool?
What improvements, if any, would you like to see in it?

Table 5.3: possible questions for use in the contextual inquiry

5.3.3 Contextual analysis and user needs and requirements extraction

After interviewing and observing the users, the second step was to conduct a contextual analysis in order to interpret the observation data. The transcripts and notes of the interviews were analysed. Elements in the data that reveal some need to improve or create a function were identified and transformed into a *requirement statement* (RS). I did this by rewording the requirement expressed in the particular segment of the data in the format of a suggestion. An RS is a self-standing and concise sentence, stating a concept, fact, rationale or idea. Each RS is identified with a

combination of a source ID (letters) followed by a numeric code. Therefore, for the following participant comment,

“I love CQPweb and I think it is incredibly useful. If I could stick my own corpora in there and work there that would be fantastic, being possible to upload your own data. So, I end up using things like Wordsmith Tools because I can put my own corpus in it.”

we have the following RS

“Users should be able to load their own corpus [TA09]”

As the RSs are generated, an affinity diagram (AD) is created. An AD is a tool to visually organise ideas, putting the RSs that address similar topics together (figure 5.1). Each RS goes into the AD in a specific format (figure 5.2) to facilitate categorisation.

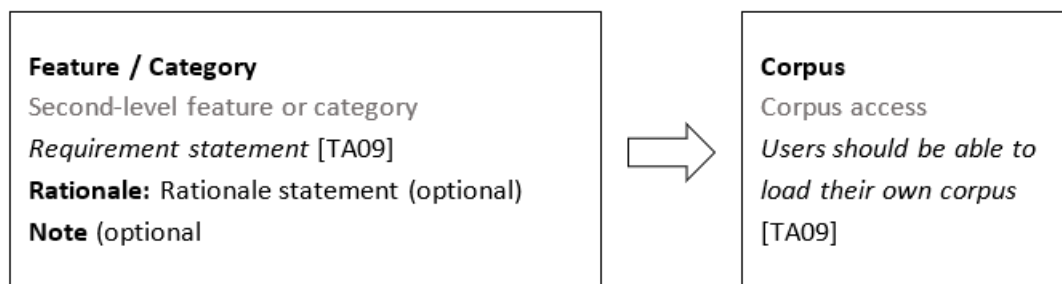


Figure 5.1: requirement statement format

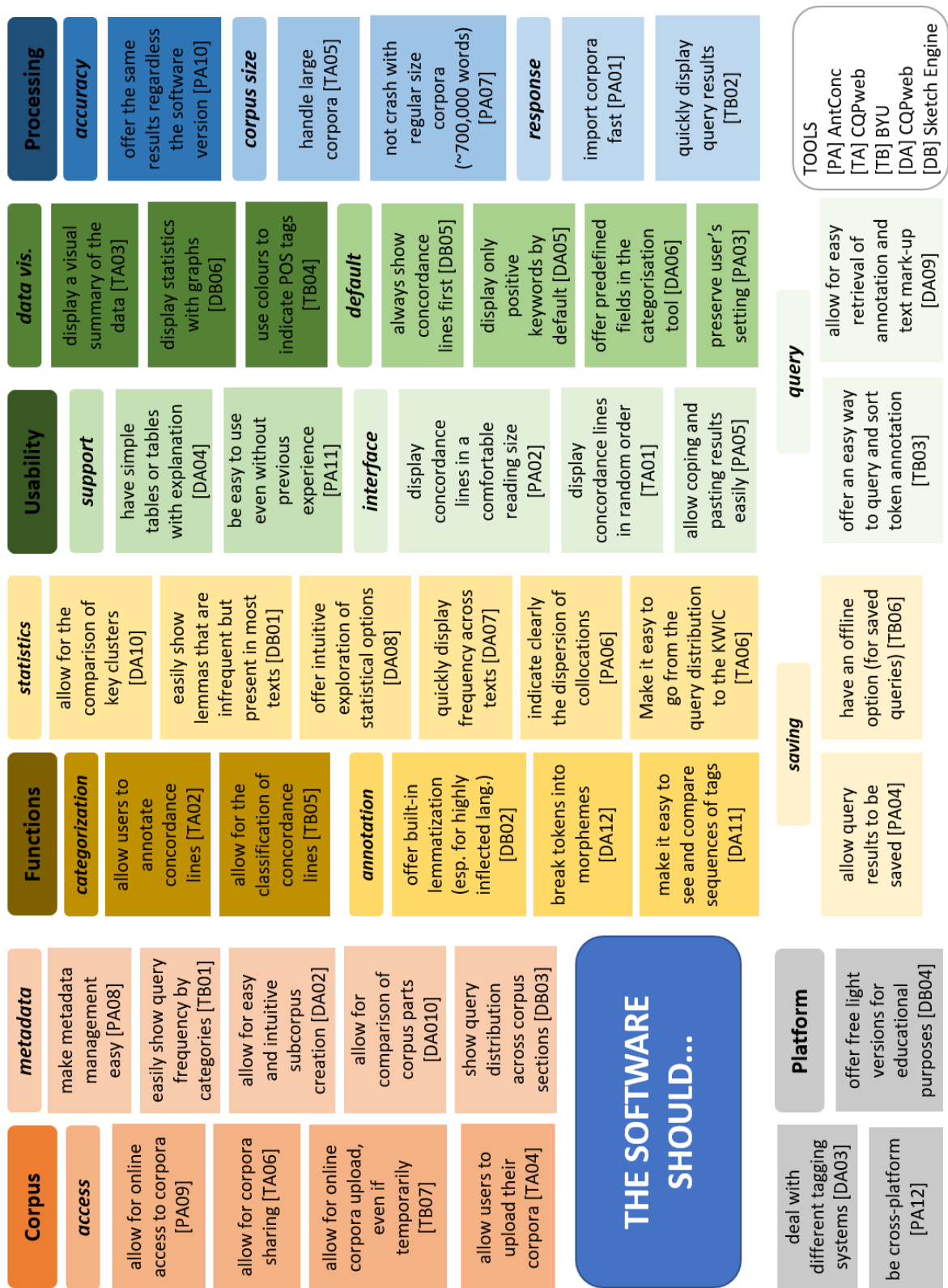


Figure 5.2: affinity diagram for the interviews/observations

5.3.4 Design-informing models

As it is not possible to cover all the requirements collected on the AD, at this point of the research, the use of *personas* was adopted. This technique, also known as *design-informing models*, consists of creating a hypothetical user with characteristic that are representative of the target user. Devising such a model user to inform the design is mainly useful to avoid designers and developers engaging in one of two counterproductive behaviours: attempting to cover all users' interests or creating a tool for themselves (Hartson & Pyla 2012). The underpinning principle is that it is better to focus on making a smaller percentage of the user population extremely satisfied (*primary personas*) without making the remainder of the user population unhappy (*selected personas*) (Cooper 2004).

Following the rationale above, a few selected personas were created by analysing the AD, and one primary persona was chosen out of this selection. Both types of personas are selected to be used as a reference when developing new tools (figure 5.3). These personas are hypothetical but specific users. The primary personas are the NSUs and the secondary are advanced users of CL. The intent with this method is not to gather an accurate description of all the interviewed and observed participants but to depict a user that is representative of what NSUs will become after a few interactions with CL tools.

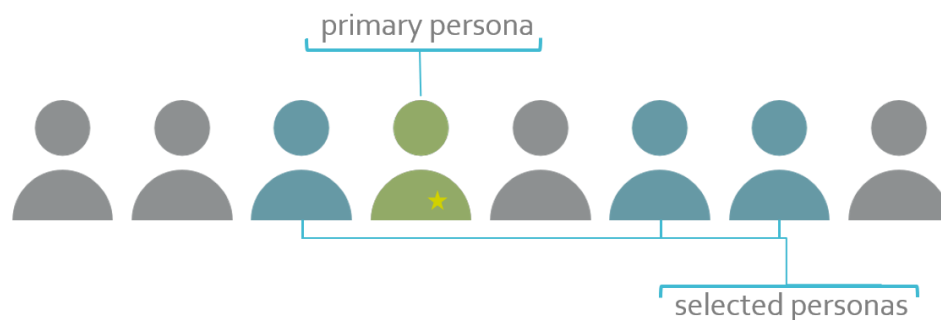


Figure 5.3: selected and primary personas

5.3.5 Tool development: iteration with users

Once the RSs to be addressed are established, the tools need to be designed. The process of developing (or enhancing) software “involves empirical definition, specification of levels to be achieved, appropriate methods, early delivery of a functional system” (Harrison & Pyla 2012:49) and also the need (and willingness) to change the system. Usually, corpus analysis tools are developed by single researchers without a team to back them up, as it is the case with WordSmith Tools (Scott 2020), AntConc (Anthony 2019) and CQPweb (Hardie 2012). As software development is time-consuming, and these creators are only part-time developers, it is unsurprisingly that such projects usually have little or no room to consider user-experience. Hence, these projects tend mainly reflect the creator’s goals or preferences.

For this reason, the first step is to design some prototypes and show them to possible users (Hearst 2009; Harrison & Pyla 2012; Cooper 2004). These users should be a close representation of the selected and primary personas described above and not necessarily the participants of the contextual inquiry.

Although possible users are aware of the intent of the tasks, they are not given detailed instructions on what the tool is supposed to do. The intent is that the design is intuitive enough for the user to have a general idea of the tool’s purpose. Possible users struggling to understand the tool is an indication that the prototype should be discarded and another one should be designed.

Since software development must always go through cycles of trials and errors, the main concern when designing the method for the present analysis was to keep all the procedures well-documented, making iterations possible (Hearst 2009). During this iteration of design solutions, user suggestions and responses are taken into

consideration. This step will be further discussed in chapters six and seven, as they discuss the development of the two new tools.

5.4 Other types of user observations

Although Contextual Design was the starting point approach used to understand the users' needs, it was not the only method used. Informal observation of groups of users was undertaken in different scenarios, as described below.

5.4.1 Workshops, lectures and seminars

Alongside to the formal expertise, I have also held or helped with CL workshops, modules and summer schools in Chile, Brazil, Turkey and the United Kingdom during my PhD. These events helped me achieve an overview of users from different nationalities, education level, and area of knowledge.

Although these users come from various contexts, they share many characteristics. For instance, a vast majority of users have Windows as their operating system. The number of Mac users were significant, while Linux users were almost non-existent. Many users have also shown interest in using CL software on their tablets rather than on a computer. In fact, in one event held in a computer lab, many users opted to continue a task on CQPweb on their phones, as the institutional internet was not reliable. One frequent complaint, however, was the need to register for an account. A common request was to be able to access the corpora without the need for a login.

These users also showed a greater interest in simplicity than in advanced functionality. For instance, when using the GraphColl function in #LancsBox, most users opted to use only the table with collocates rather than using the graph. They claimed that the

table was more straightforward and intuitive than the visualisation, even if the visualization was what attracted them to use the tool.

In the context of university seminars, I also observed that when students had a clear and real example of a corpus query, they were more likely to succeed in their assignments. Providing concrete examples in the classroom has proven effective with students. Hence, adding help pages or tooltips to CL software could be convenient.

5.4.2 Talks

Attending lectures given by researchers who rely on corpus methods was also a source of information. For instance, in a talk on text mining of political speeches, Blaxill (2016) explained that his methodology was changed in the process of his research. He had initially chosen to perform his research using AntConc, for its being user-friendly and free of charge. However, due to the limitations of the application, the text investigation tool was then changed to Python, with the help of paid external staff to develop the scripts. In this case, the research was not drastically affected by the limitation of the first choice of tool. Situations like that might suggest that the findings in 4.3, demonstrating a considerably high number of studies relying on programming language, might not be an actual representation of users' abilities. In most research contexts, it is more likely that the research is driven by the tools instead, due to time, computational knowledge and funding restrictions.

5.4.3 Web analytics

Web analytics consists in collecting and reporting internet data and usage. This procedure is done in parallel with the other experiments and throughout the whole process. The intent here is to, by observing the most frequent queries, understand what users do or do not do with the tool. For instance, studies on search interface have

shown that users struggle with Boolean format when performing their queries (Hearst 2009:108; Dinet et al. 2004; Hertzum & Frokjaer 1996).

For this investigation, I used the function *query statistics* for the CQPweb at Lancaster²⁶. This function, which is only available for administrator users, reveals the usage statistics for particular query strings, as shown in figure four. The table with the 40 most frequent queries reveals that most searches are simple queries for words and do not rely on advanced searches with the Corpus Query Processor (CQP). However, a more in-depth investigation should be taken, such as asking users directly, since it might indicate that users are not aware of CQP queries, but alternatively that they are not interested in this kind of query.

Query	Freq.	Query	Freq.
1	20,668	21 [word="bloody"%c]	1,209
2 [word="the"%c]	3,808	22 [word="can"%c]	1,190
3 [word="not"%c]	2,712	23 [word="like"%c & pos="RR"]	1,182
4 [word="fuck"%c]	2,344	24 [word="perfectly"%c]	1,130
5 [word="lovely"%c]	2,224	25 [word="research"%c]	1,126
6 [word="must"%c]	2,216	26 [word="man"%c]	1,114
7 [word="however"%c]	1,912	27 [word="love"%c]	1,111
8 [word=".*n't"%c]	1,829	28 [taglemma="(question)_SUBST"%c]	1,088
9 [word="sorry"%c]	1,805	29 [word="said"%c]	1,066
10 [pos="NNB"]	1,710	30 [word="beautiful"%c]	1,049
11 [word="like"%c]	1,649	31 [pos="GE"]	1,034
12 [word="sick"%c]	1,569	32 [word="refugee(s)?"%c]	999
13 [word="shall"%c]	1,497	33 [word="woman"%c]	965
14 [pos="N.*"]	1,433	34 [word="please"%c]	953
15 [word="utterly"%c]	1,403	35 [word="you"%c]	937
16 [word="ill"%c]	1,352	36 [word="says"%c]	932
17 [word="I"%c]	1,286	37 [word="people"%c]	926
18 [hw="fuck"%c]	1,282	38 [word="red"%c]	916
19 [word="alien(s)?"%c]	1,277	39 [word="something"%c]	899
20 [word=".*ly"%c]	1,248	40 [word="bare"%c]	884

Table 5.4: Top 40 queries at CQPweb Lancaster

²⁶ <https://cqpweb.lancs.ac.uk/>

5.5 User needs and requirements

From the observations carried using both methods, many user needs and requirements were found, as described in figure 5.2. This section discusses some of the general observations.

The first generation of CL software (McEnery & Hardie 2012) consisted basically of a tool with one functionality, mainly generating concordance lines. The scenario today is different. Users now want to have a combination of tools in a single package. For instance, the RS below

The software should allow users to classify the concordance lines [TB05]

suggests that users want to do all the process of corpus exploration and analysis in a single environment. This is particularly true when it comes to statistics. Many of the requirements were related to the need for built-in user-friendly statistical tools.

Although the participants have shown the desire to have tools that do more than concordance lines, in many cases, users use only a small part of the functions available. In most cases, this is because they are unaware of the existence of the functions. Another reason is that they find them too complex to use. For instance, the RS

*Distribution should be straightforward and easily connected to KWIC
[TA06]*

comes from the user statement

“We don’t bother using this functionality [distribution] ‘cause it’s just not very straightforward and what we do is concordance lines analysis, grammatical analysis, looking for semantic prosody”.

Hence, even if the required tools exist, if people cannot use them, it is as if the tools did not exist at all.

Not seeing a tool or not being able to use it is mainly because the software lacks usability. Many of the RS related to usability. In fact, 14 out of 44 were explicit requirements related to the interface and making procedures easier. For example, the RS

It should be easier to retrieve annotation and text mark-up [DA09]

comes from a user explaining why he only performs simple queries and manually filter the results according to the desired part-of-speech.

5.5.1 Some caveats

Users tend not to know what they want, and their research is led by the tools that already exist. The methods and approaches described in this chapter are suitable means of minimising this effect and helping identify user needs and requirements. However, we should not use those methods solely for two main reasons.

First, some needs are not tangible or easily detected. For instance, the participants in this study were able to demonstrate their needs for tools and methods that they are already familiar with or aware of. They could not point to a scenario of which they are not aware.

Secondly, this type of research can sometimes deviate from the primary goal, as it gives more space to reveal issues with usability rather than functions. Tools should be made user-friendly, so it is accessible to more people. However, they should be, ultimately, useful.

5.6 Summary

This chapter has covered the steps to perform a contextual design approach (5.3) and has also discussed other informal observations used in this study (5.4). In section four I presented the findings of the observations in the form of an affinity diagram.

Based on the analysis presented here and in the previous chapter, two new tools were developed for CQPweb. As explained in chapter three, CQPweb was chosen mainly based on its being open-source, web-based and well-established. The investigations carried in the current chapter also contributed to this choice, as CQPweb features many of the requirements discussed here. Some of these requirements are the possibility of sharing corpora online and the easy to deal with text metadata.

The first tool, which will be described in the next chapter, addresses the need for easy statistical analyses. Its development also considers the users' need for usability via a simple interface. The second tool developed deals with parallel corpora and is addressed in chapter seven.

6 Advanced Dispersion

Multiple studies have been conducted on the importance of the use of measures of dispersion and distribution in Corpus Linguistics (CL). However, there is still a paucity of studies, especially in applied linguistics, that make use of advanced and reliable measures of dispersion. In chapter four, I found that very few studies using corpus methods report dispersion, in line with the findings of Gries (2008, 2013). One reason for this may be the lack of convenient and user-friendly functions to calculate dispersion measures in most widely used CL software.

This chapter will: discuss measures of dispersion commonly used in CL (6.1); present reasons for graphically visualizing dispersion (6.2); discuss factors to be considered in the design of a visualization (6.3); discuss a range of different prototypes for such a visualization (6.4); and present the implementation of a new visualization system for dispersion (6.5).

6.1 Dispersion: definition, measures, applications

Corpus linguistic methods rely heavily on frequencies and distributions, no matter how advanced the users are. However, there is a considerable difference between the

way CL statistic experts and non-specialist users of corpus data and methods (NSUs) interpret these frequencies and distributions.

Frequency, in a simplistic way, refers to how often a single word or tag or a combination of units, or anything that can be counted, are present in a corpus or corpus section. It can be presented as an absolute value or normalized to a frequency relative to the corpus size. This is a basic, yet crucial, piece of information for corpus analysis. However, when a non-proficient user of CL methods relies solely on frequencies of, say, a word for the analysis of a measure, the results can be deceiving. Frequency, relative or absolute, does not take into account information such as the relationship of the word under study with others and how it is distributed in the corpus.

In the evolution of CL, different techniques have emerged to furnish the information that frequency on its own cannot provide. A concrete and simple (yet widely used) is the type/token ration (TTR). As the name indicates, *type/token ratio* gives the number of *unique* elements (types) per the total number of elements (tokens) in a corpus. These elements are most often words. This metric is often used as an indicator of lexical diversity when comparing two corpora of comparable sizes (Baker et al. 2006:162). It can also inform the interpretation of frequency, as TTR is affected by the size of the corpora. CL methods mainly address *divergence*, the difference between two corpora or subcorpora, through the calculation of key words. Keywords in CL are items that occur in a corpus or corpus section more often than would be expected and are often presented in a list format. To address how element(s) in a corpus are associated to each other, different *correlation* measures are adopted. A common technique used to express this relationship is through collocation, “the phenomenon

surrounding the fact that certain words are more likely to occur in combination with other words in certain contexts” (Baker et al. 2006:36). In most CL software, users can choose how the collocations are generated; the results include a statistical score for each collocate. However, for the average user, the main information absorbed is the list with the top collocates.

These techniques are frequently observed in studies which use CL techniques. Even if the quantitative mechanism behind the calculation of keywords and collocations are not very clear for non-advanced users, such users can still make sense out of the data these techniques provide, as table of collocates or key words constitute tangible linguistic output. However, there are other important measures to be considered in CL that are not so easy to picture and are in need of more attention. Dispersion in a corpus is one of them. Dispersion may be defined as “(t)he degree to which occurrences of a word are distributed through a corpus evenly or unevenly/clumsily” (Gries, forthcoming). When not completely neglected, dispersion is often treated by linguists in an oversimplified way. Gries refers to words, but dispersion can be calculated for word types, lemmas, phrases, annotations of any sort, or the results of any query in a corpus. Henceforth, the term *distribution* is used here to refer to distribution of any kind of item resulted from a corpus query, albeit using word-type distribution as the paradigmatic example.

There is no consensus on the difference in terminology for word dispersion and word distribution. In some cases, *dispersion* refers to how the results of a query are spread across a whole corpus or text and *distribution* when referring to the disposition across corpus categories, i.e. groups of texts that share some characteristics. For the purpose of this paper, *dispersion* and *distribution* will be used interchangeably to refer to the

degree a word is spread out across corpus units, as the pieces of software discussed in the next subsection section use both terminologies. If a word occurs much more often in one text of a corpus than in the other texts, it can be said to be unevenly dispersed. Conversely, an evenly dispersed word is expected to have a relatively constant presence across all corpus texts (Gries 2008).

There is a distinction to be made between quantitative and qualitative approaches to dispersion. The quantitative view characterises how word occurrences vary across defined corpus sections and positions. The qualitative approach instead prioritizes construing dispersion through the layout spread of instances within corpus units, which are, in many cases, represented by each text.

While the qualitative approach is the one usually observed when non-advanced corpus linguists incorporate dispersion into analyses, an advanced and detailed view of dispersion does require the quantification of dispersion. This can be achieved through the use of one of a number of statistical measures, which will indicate the degree to which a word or phrase appears to be well-dispersed. Different dispersion measures have been used and reported in CL studies. Among those frequently used are *range*, Rosengren's *S* (Rosengren 1971); Carroll's *D2* (Carroll 1970); and Juilland's *D* (Juilland et al. 1970).

Range is calculated by simply counting how many texts in the corpus the searched word or phrase occurs in. Range is, by far, the dispersion measure most frequently reported in my literature investigation. For this calculation, the size of the texts as well as the frequency of the word are disregarded. This simple and straightforward calculation might be the reason why range is present, even if not always named as *range*, in a wide variety of tools, such as AntConc (Anthony 2019), WordSmith Tools

(Scott 2020) and CQPweb (Hardie 2012). The ubiquity of range across different CL tools is also likely to explain why it is, to date, the most used dispersion measure.

However, range is not a robust measure as it may not perform well in certain circumstances, yielding false conclusions (Brezina 2018:48). For instance, the word “six” occupies position 240 in a word frequency list generated from the Spoken British National Corpus 2014 (Love et al. 2017), with an absolute frequency of 4,665 and a relative frequency of 408 words per million (wpm). If we only consider range, we can say that *six* is fairly well distributed across the corpus, as it occurs in 904 different texts out of 1,251 texts in the British National Corpus (BNC) 2014. However, further investigation shows that the frequency of *six* in each text varies greatly: the lowest relative frequency observed is 41 wpm, and the highest is 8,883 wpm (figure 6.1). In this scenario, using only range to address dispersion would not suffice.

Because this measure might lead to inconsistent observations, approaches have emerged to address issues caused by uneven distribution of words across texts. For instance, if a topic-specific word occurs in a text, it is likely that its frequency will be much higher in this text than in other parts of the corpus. This probabilistic phenomenon is characterised by Kilgariff (1997) as the “whelk problem”, in reference to dispersion of the lemma *whelk* in the BNC 1994. Although it is an infrequent word in everyday English, if a corpus features a text about whelks, the mollusc, the overall frequency of “whelk” will be deceptively “high”, in terms of not clearly reflecting its generally lower frequency across all other texts. To address this problem, Kilgariff (1997) suggests limiting the size of the sample retrieved from each of the sources that populate the corpus. If using a corpus already compiled, a common approach is to

consider only words or phrases that occur at least once in a minimum number of texts (Kilgariff 1997). But because the necessary minimum would vary according to the corpus size, it is difficult to define a threshold.

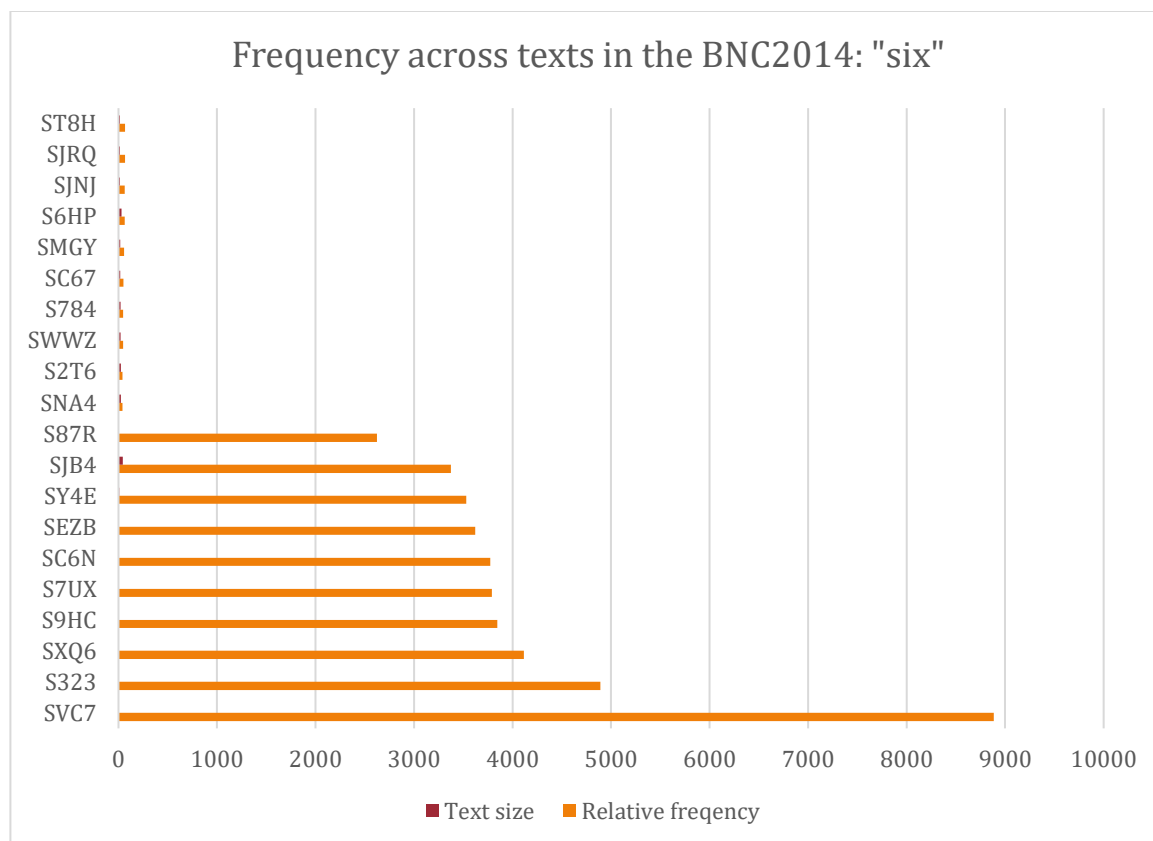


Figure 6.1: relative frequency for ‘six’ for individual texts in the BNC 2014

Another point to be considered is how the corpus is compiled. To calculate range, it is necessary that the corpus be divided into parts, and often the parts corresponds to the texts. However, this is not always the case. Many corpora are given just as a single unit, meaning there are no text or unit boundaries. Another issue is that, even if a corpus is indeed divided into parts, these parts can vary greatly in size. An approach to reduce the impact of wide variation in size of parts and to tackle the issue of the lack of any divisions would be to compute dispersion by considering the corpus as a single string and calculating the distance between successive occurrences of the element in question (Savický & Hlaváčová 2002; Washtell 2007). Unlike range, this measure is

not affected by the text boundaries, but only by the frequency and locations of the word in the corpus. This can be helpful, for example, when comparing word dispersions in corpora with very distinct structures. The high amount of processor time and computer memory for this calculation is a limitation of this approach (Gries 2008). Thus, the usability of this method is questionable. Moreover, to my knowledge, there is almost no research on its application in CL.

A more commonly used measure is Juilland's D (Lyne 1985). This is calculated as follows: (1) calculate the standard deviation(s) for the frequencies of a given word across corpus parts; (2) divide it by the mean frequency of the word in the entire corpus; (3) divide this result by the square root of the difference between the sizes of the corpus parts and 1, and then subtract the result from 1, as shown below.

$$s = \sqrt{\frac{\sum_{i=1}^n n(x_i - \bar{x})^2}{n}} \quad \blacktriangleright \quad V = \frac{s}{\bar{x}} \quad \blacktriangleright \quad D = 1 - \frac{V}{\sqrt{(n-1)}}$$

When Juilland's D is used as a dispersion index, a word that is perfectly distributed in the corpus (meaning it has the same relative frequency in each and all corpus parts) receives a score of 1. Conversely, a word that only occurs in one corpus part is given a score of 0. This measure is commonly used for the compilation of frequency dictionaries and word lists, to avoid unevenly dispersed words being highly ranked in a way that is not optimal for this application. For instance, Davies & Gardner (2010) rely on a combination (in this case multiplication) of each word's frequency and Juilland's D dispersion index. Their frequency dictionary only includes words with a score for the combination of Juilland's D and word frequency above 0.94, to guarantee that the word is relatively ubiquitous (Davies & Gardner 2010:5).

Although Juilland's *D* is a commonly used dispersion measure, it has suffered criticisms. A main critique (Gries 2008) is that its value is affected by the number of corpus parts. The more parts a corpus has, the closer to 1 the index score for a given corpus part will be, all else being equal. This behaviour might wrongly suggest that a word is evenly distributed when this is not the case (Biber et al. 2016, Burch et al. 2016).

An alternative measure of dispersion, the Deviation of Proportion (DP), has been proposed by Gries (2008) and noted by other researchers (Biber et al. 2016). It is calculated as follows: (1) compute the difference between the observed and expected frequency of the word for each corpus part as a percentage; (2) sum the absolute values of these differences; (3) divide the result by 2.

$$DP: 0.5 \times \sum_{i=1}^n \left| \frac{v_i}{f} - s_i \right|$$

A normalised version of DP (DP_{norm}) is calculated by dividing DP by the difference between 1 and the size of the smallest part in the corpus (Lijffijt & Gries 2012).

$$DP_{norm}: \frac{DP}{1 - \min(s)}$$

For both measures, lower values indicate more even distribution.

To my knowledge, this measure is only calculated by one of the CL software packages discussed in chapter three of my thesis, #LancsBox (Brezina et al. 2015). However, some authors recommended it, for its simplicity of calculation; the ease of understanding of the results; the consistency of results even across corpora with unequally-sized parts; the spread of scores throughout the range from 0 to 1, unlike

Juilland's D scores, which tend to be concentrated at the higher end of the range (Gries 2008, 2013; Biber et al. 2016). One criticism of this measure is that in its calculation, the frequency of a word in each corpus part is not taken individually but rather as a sum of the values for all the texts (Burch et al. 2016). An alternative to DP which addresses this is D_A (formula below). This measure is obtained by calculating the average of the differences of the distances between all pairwise sequential occurrences of the word in question in the corpus. As Burch et al. state, D_A is thus derived from of detailed information of the frequency in each text, whereas D_p relies on this information presented in batches. Although theoretically promising (Burch et al. 2016), this measure has the drawbacks of lengthy processing time and a lack of research on to date its applications to CL.

$$D_A = 1 - \frac{1}{k(k-1)/2} \frac{\sum_{i=1}^{k-1} \sum_{j=i+1}^k |Y_i - Y_j|}{2\bar{Y}}$$

Another approach to addressing uneven dispersion is to use an *adjusted frequency*. As the name suggests, this involves adjusting the absolute frequency in such a way as to minimize the impact of uneven dispersion. Adjusted frequency can be calculated on the basis of distance or of corpus parts (Gries 2008). To calculate the adjusted frequency based on distance (AFD), the distances between successive occurrences of the word are regarded. A lower average distance between occurrences means that the frequency is adjusted to fewer words by a greater degree.

Savicky and Hlavacova (2002) study three different measures for adjusted frequency based on distance: Average Reduced Frequency (ARF), Average Waiting Time (AWT) and Average Logarithmic Distance (ALD). They conclude that the ARF is the most consistent of these measures, across a range of different kinds of corpus. This

calculation is ideal when dealing with corpora that are not pre-divided into parts, as the distance between tokens rather than frequency within corpus parts is the basis of the measure. However, as previously mentioned, any calculation which relies on distance between tokens demands intensive computer processing as each such distance must be extracted. This occurs when dealing with a non-indexed corpus, but as discussed in chapter three of my thesis, this process is more efficient when dealing with indexed corpus.

A more efficient approach, in terms of computer processing, is to calculate the adjusted frequency based on corpus parts (AFP). Some common AFP measures are based on a combination of word frequency and dispersion measures (Gries 2008). This includes, for example, Rosengren (1971) Adjusted Frequency (R_{AF}). R_{AF} is the product of Rosengren's S with the quotient of the word frequency and the number of corpus parts.

However, as frequency and dispersion are different concepts, combining them into a single value means losing certain amount of information. For instance, when using Juilland's D as a base for adjusted frequency, this is done with the product of the absolute frequency and the dispersion value. So, if an adjusted frequency is reported as 18, it is not apparent whether the real frequency is 18 with Juilland's D equal to one or the real frequency is 180 with Juilland's D equal to 0.1 (Gries forthcoming). Yet, it is worth pointing that no one is proposing the use of adjusted frequency for every purpose, rather that for some specific reasons, such as when deciding to include in a frequency dictionary (Brezina 2018). AFs are designed for specific purposes such as developing learner word lists. However, for linguistic investigation it is interesting to have frequencies and dispersion scores as separate measures.

The measures discussed so far are just some of the many ways of calculating dispersion. Gries (2008) compares the behaviour of 24 different dispersion measures and concludes, unsurprisingly, that “different measures of dispersion will yield very different (ranges of) values when applied to actual data” (Gries 2008:9). As a way of reducing this large array of possibilities, Gries (2010) groups measures that behave similarly, and advises that if the user is not confident which measure to adopt, one measure from each group should be calculated, so as to have a comprehensive view of the data. However, Gries (2010)’s advice cannot help people who are not well-versed in statistics. In fact, a pattern that I identified during the observation of participants (chapter five) was that they are not even aware of dispersion. For those who are so, they do not fully grasp the meaning of the measures.

6.2 Graphical visualization of dispersion

As discussed in the previous section, there is a vast number of ways to measure dispersion, whose suitability varies according to the data being analysed. For NSUs these dispersion measures present only an ever-growing bundle of opaque numbers which often imply confusing results. As discussed in 6.1, the concept behind the calculation of dispersion measures are not easily grasped by NSUs. For instance, consider this extract from an article included in the Literature Investigation (chapter four):

The software WordSmith Tools was used for the linguistic-textual process.

Through the Juilland dispersion coefficient and use coefficient, the most frequent phraseological units were identified in the academic texts. (Silva et al. 2017:345)

The authors do not go into any further detail and give the impression that Juilland's coefficient is related to the frequency of the units in question rather than to the spread.

From the observations explicated in the previous paragraph, we can infer that users struggle to understand dispersion through the summary of statistics. If they could see how a given query is dispersed across the corpus, instead of concentrating on the calculation of dispersion itself, this process can be easier. The purpose of developing a visualization is to help this audience be aware of the importance of word dispersion in a corpus, understand its functioning and apply dispersion measures in their analysis. Plotting, i.e. illustrating by means of a graph, the dispersion is, thus, a means of aiding the user work around the difficulty of intuitively grasping what is meant by apparently arbitrary scores on arbitrary scales.

As Gries (2010) observes, a considerable number of researchers who use CL methods lack two important methodological skills, statistics and programming (Gries 2010, 2018; Paquot & Plonsky 2017). Inadequate applications of CL methodologies and their shortcomings have been reported for many years now. Baayen (2001) indicated a high number of studies overlying on frequency and neglecting information on dispersion. Other publications also point to the lack of exploration of heterogeneity within the corpus (Kilgarriff 2001) and of variation by text or speaker (Brezina & Meyerhoff 2014).

For this, Gries (2010) argues that a great number of researchers mostly rely on one particular CL application software. Thence, their research is limited by what the software is able to do (Gries 2010; Gries 2015: 93). In line with what he says, in most of the cases observed in my literature investigation (chapter four), dispersion is

reported – if at all – through the range measure. This is probably due to the simplicity of its calculation or to its prevalence in several CL software applications.

When it comes to visualizing dispersion, some pieces of software already show some graphic representation. Five of them are discussed in this section.

6.2.1 AntConc: Concordance Plot

AntConc includes a tool called *Concordance Plot*, which allows users to visualize the positions at which tokens of a given word occurs. Each token is represented in the position it occurs in a text by a vertical line forming a *barcode plot* for each text (figure 6.2). Clicking on any of the lines takes the user to the textual context in which the clicked word is. The name of the text is given above its barcode plot; the absolute frequency of the word is given on the right-hand side of the plot. Files that do not contain the searched word are not included in the visualization. When a word occurs in bursts, the vertical bars are packed together closely; the tool offers an option to zoom in and out, so as to have a better view of these areas. One common use of this tool is to verify whether the high frequency of a word in a corpus is due to a topic-related text, skewing the results. Thus, the interpretation is based on its bursty visual appearance.

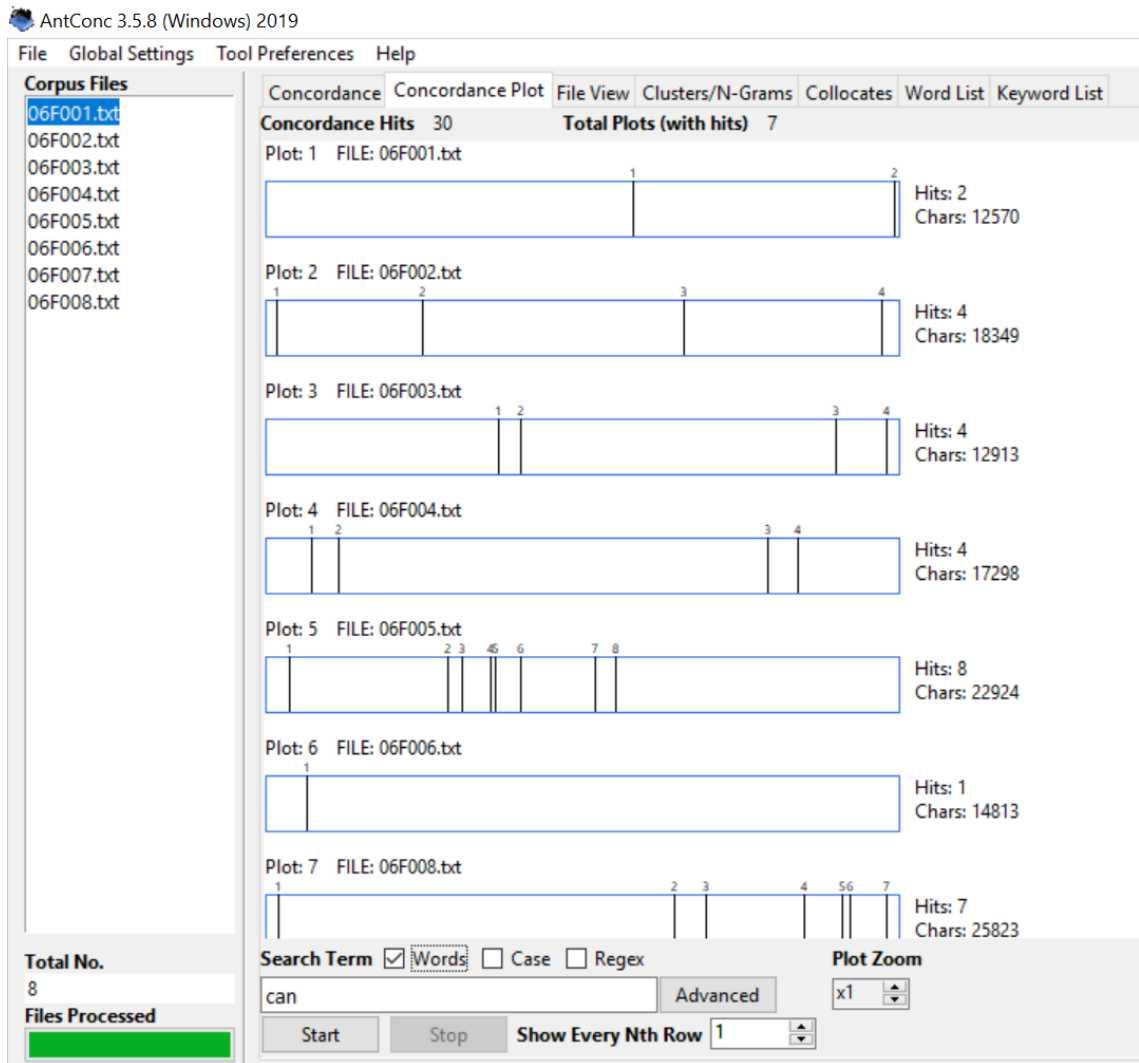


Figure 6.2: screenshot of a concordance plot from AntConc

AntConc refers to how a word is laid out in a corpus. This is a different approach from the dispersion measures discussed in the previous section. The dispersion measures aim at summing up in one number how spread-out or concentrated the tokens of word are. The AntConc visualization, on the other hand, simply displays the actual, concrete locations of the tokens so the user can carry a qualitative analysis of the spread.

Although the Concordance Plot has the benefits of quickly and easily revealing how a word is dispersed in a file, this tool does still have some shortcomings. For instance, AntConc does not provide information on the size of each section of the corpus, i.e. each text. Instead, it misleadingly represents each corpus file (text) as if all were of the

same size, even though, in many cases, they are not. Also, seeing the dispersion in each individual file can be useful, especially when a corpus is compiled within only one text file. However, users might want to see dispersion across the whole corpus before analysing each individual file one by one. One limitation of AntCon is, thus, not to provide users with a visual layout for the whole corpus.

6.2.2 WordSmith Tools: Text Plot

Text Plot is a tool in WordSmith Tools (Scott 2020) that, like *Concordance Plot* in AntConc, displays each occurrence in a text as a vertical line, and each file, i.e. text, as a barcode plot. However, *Text Plot* also offers some functions that are not available in AntConc. Besides the absolute frequency of the searched word for each text, *Text Plot* also provides, on the left-hand side of the barcode plot, the relative frequency of this word, and a measure indicating how dispersed the word is in the whole corpus. It also displays a barcode plot for the entire corpus, allowing the user to see the dispersion across all files at the same time (figure 6.3²⁷).

Double-clicking on the plot opens a list with all the numbered tokens and the word's position is given (figure 6.4). As in AntConc, all texts are graphically represented with a horizontal bar of the same size, even if they differ in length. However, Text Plot offers a *Uniform view* option. This setting causes each text's length to be represented by an initial and a final blue bar (figure 6.5). Because Text Plot also presents the

²⁷ WordSmith screenshots were retrieved from https://lexically.net/downloads/version7/HTML/dispersion_basics.html

dispersion measure, this tool provides the user with more than only a qualitative view of dispersion. The quantification of dispersion is described as follows:

It splits the corpus up into a number of divisions (default = 8) and for every word, computes how the word spreads out in the whole set of texts (Scott 2020).

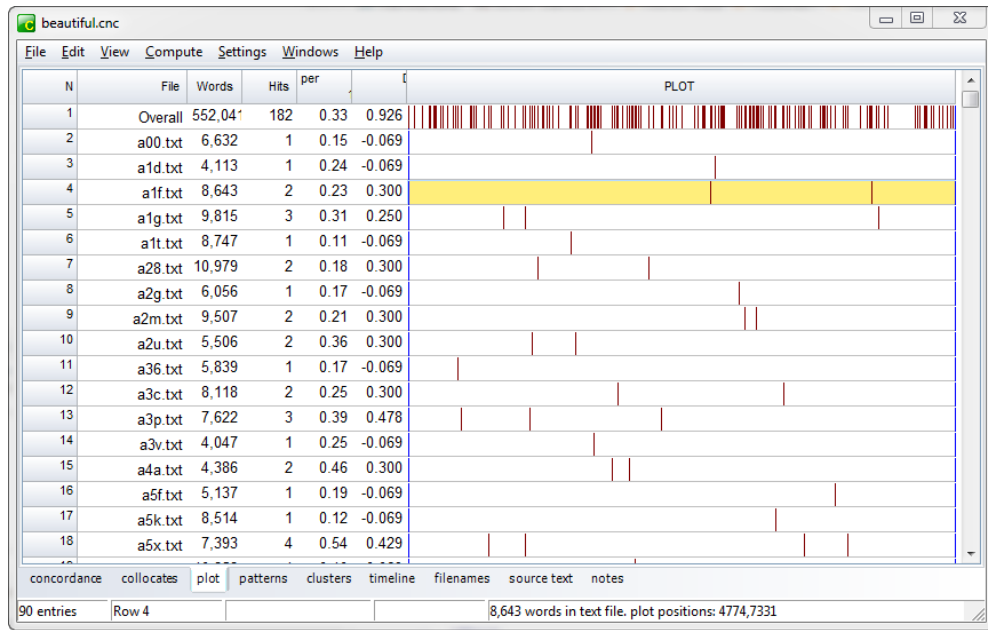


Figure 6.3: screenshot of Text Plot, in WordSmith Tools

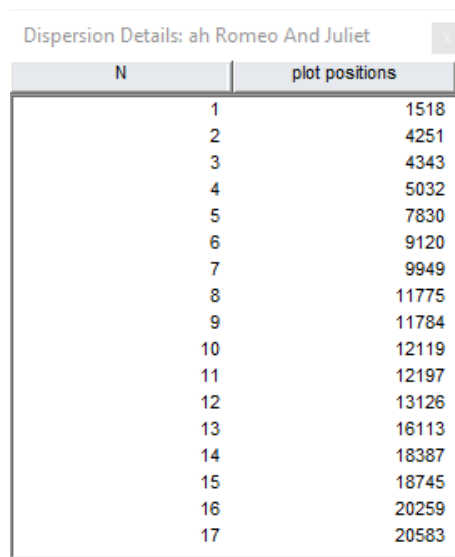


Figure 6.4: screenshot of Word Positions, in WordSmith Tools

This dispersion measure is similar to Juilland’s D. The only difference is that it divides the quotient of the standard deviation and the mean of frequencies over corpus parts by the square root of corpus parts, and not corpus parts minus 1, as observed in Juilland’s D. One drawback with this measure is that the number of divisions, eight, is arbitrarily chosen. However, the users can alter this number in the settings and set their own division based on the text files.

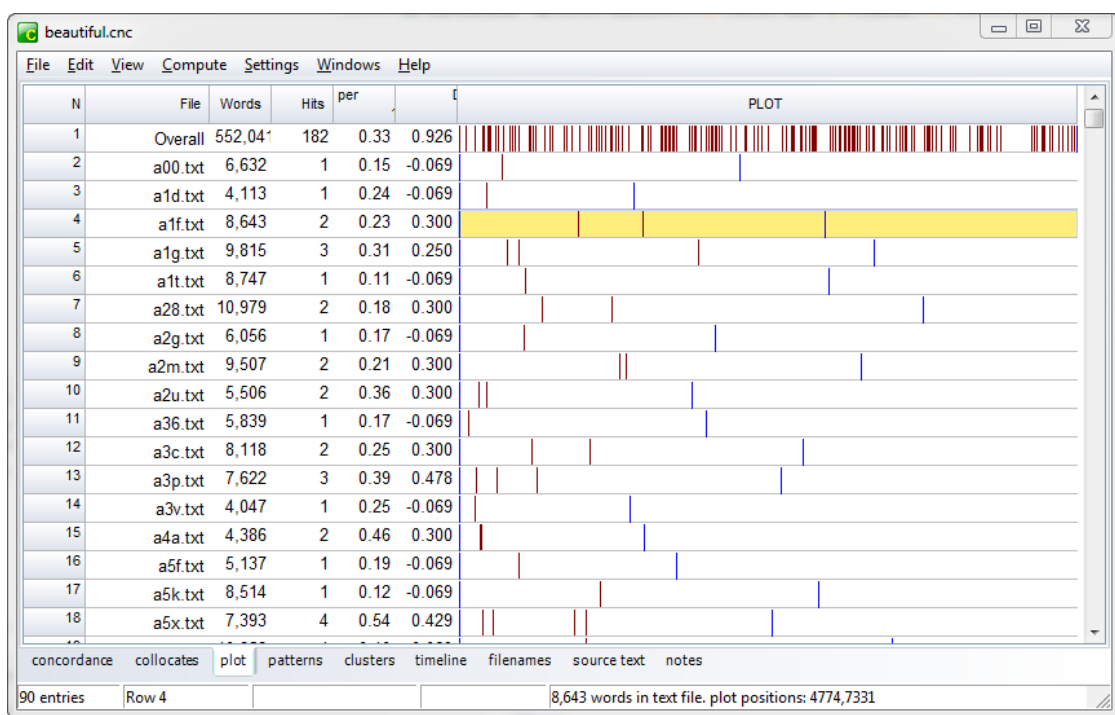


Figure 6.5: screenshot of Uniform View, in WordSmith Tools

6.2.3 Quanteda: Textplot x-ray

Another tool that displays dispersion graphically is *Textplot x-ray*, available within a software library called Quanteda (Benoit et al. 2018). This tool shares the basic principles of WordSmith and AntConc’s display. Each occurrence is shown as a vertical line and each text is represented as a rectangle. As in WordSmith’s *Text Plot*, the user can choose whether the rectangles for each text are presented at the same size (relative scale), or at sizes that differ according to the text length (absolute scale). This

choice can be useful, for example, in situations when the user wants to inspect relative burstiness (figure 6.6²⁸). Another advantage of Quanteda over the previous tools is that it also allows the user to modify the plot. For instance, the user can perform the analysis on query terms and plot more than one set of results at the same time (figure 6.7). One drawback of Quanteda is that it is an R package, rather than a full application, and using it requires some R programming knowledge, which, by definition, is not a requirement in my target user group.

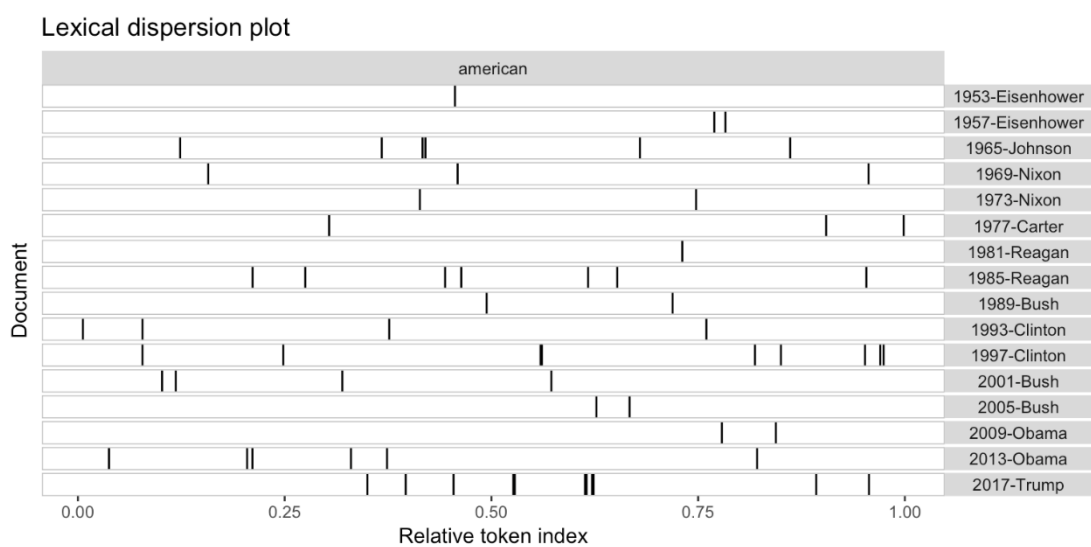


Figure 6.6: Text Plot x-ray, in Quanteda (relative scale)

²⁸ The Quanteda screenshots in figure 6 and 7 were retrieved from <https://quanteda.io/articles/pkgdown/examples/plotting.html>

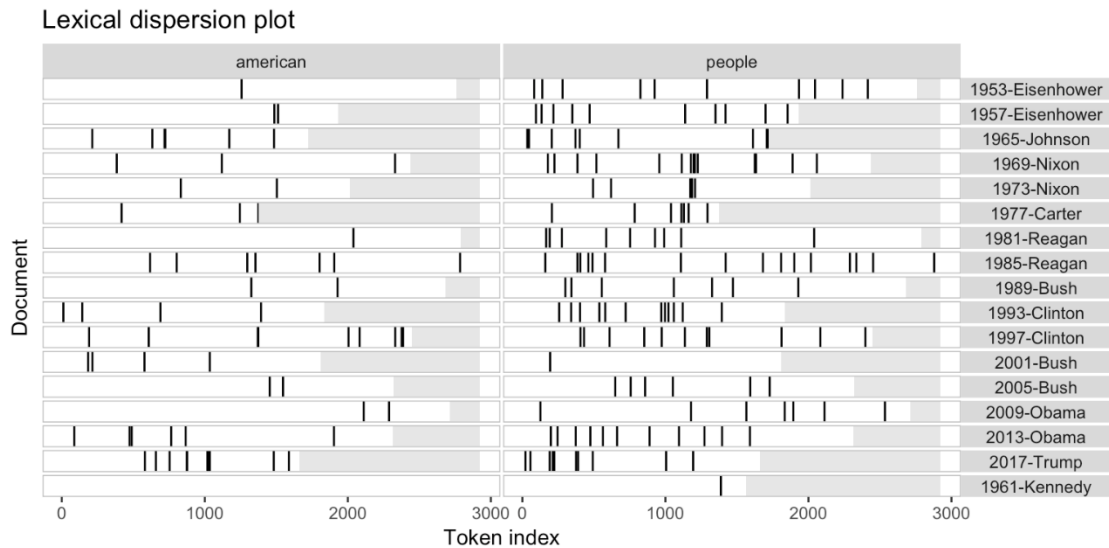


Figure 6.7: Text Plot x-ray for two words, in Quanteda (absolute scale)

6.2.4 CLiC: concordance plot

The three pieces of software mentioned above are desktop-oriented. This means that the user needs to install the software on their computer to use it. As discussed in chapters two, web-based corpus tools are becoming increasingly more popular among users. To my knowledge, two such applications display dispersion graphically: CLiC (Mahlberg et al. 2016) and Voyant Tools (Sinclair & Rockwell 2020).

CLiC is a web application developed to support narrative fiction analysis (Mahlberg et al. 2016). As of this writing, CLiC offers four built-in corpora and several individual books but does not allow the user to upload their own corpora. One of the tools available in CLiC is the *Distribution plot*, which, in a similar manner to AntConc, WordSmith Tools and Quanteda, renders the dispersion in the format of a barcode plot (figure 6.8). As in AntConc, the rectangle has a fixed size, regardless of the text size; likewise, no dispersion measure is calculated. When the vertical bar representing an occurrence is hovered over, the user can see a fragment of the context in which the

token occurs. If the vertical bar is clicked, another window opens, displaying the entire text.

One feature unique to CLiC is the visualization of sections within each barcode plot. Each book of the corpus is represented by a barcode plot and the start of each chapter within each book is indicated by a small triangle below the plot. This allows the users to see how the search term is dispersed in the corpus and its subsections (books and chapters) at the same time. However, this functionality can only be implemented due to the characteristics of the corpora available in CLiC.

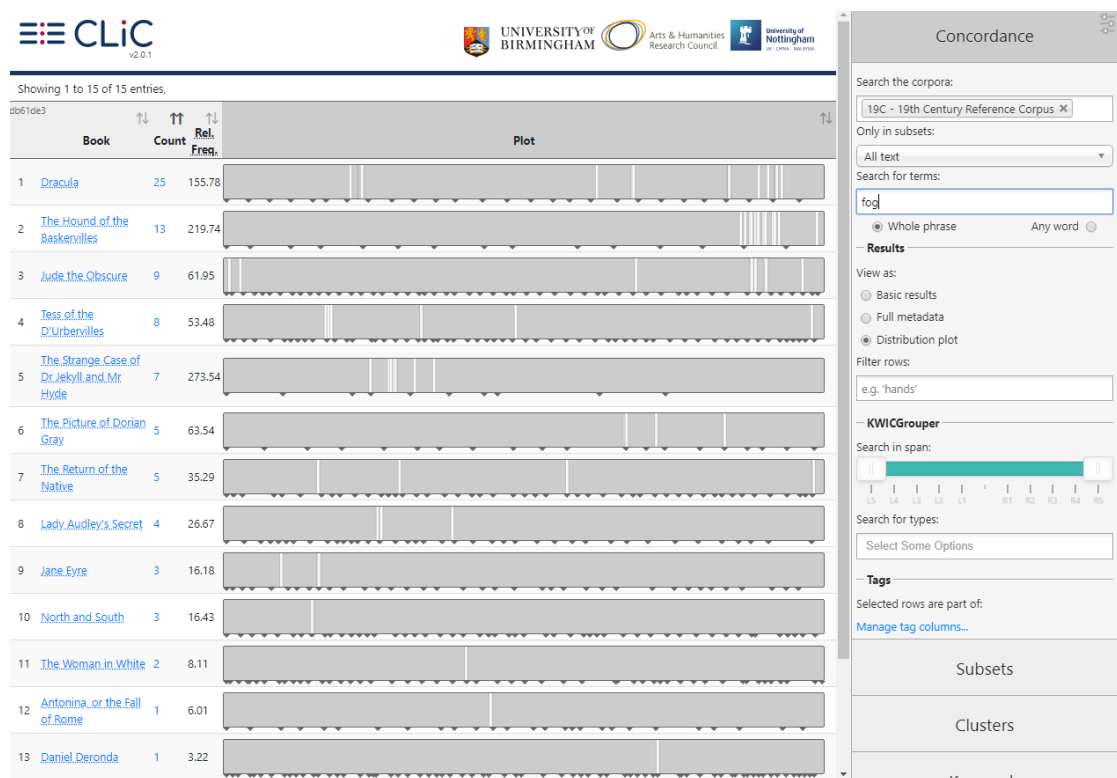


Figure 6.8: screenshot of Concordance Plot in CLiC

6.2.5 Voyant Tools: corpus terms and trends

Like CLiC, Voyant Tools is a web-based environment for computer-assisted linguistic analysis. It comes with two built-in corpora, but users can also upload their own

corpora. Among the 24 tools that Voyant currently offers, two address distribution or dispersion in a corpus: *Corpus Terms* and *Trends*.

Corpus Terms shows a table with the frequencies (counts) and distribution (trend) of each word (term) across the corpus (Figure 6.9). The trend is represented on a sparkline graphic. A *sparkline* is a small line chart that visualizes tendencies and variations in a very succinct way. Here, the sparkline indicates the relative frequency of the word in each text in the corpus. The peaks and valleys of the relative frequencies are highlighted with a small orange dot. When the user's mouse hover over the line, the tool shows the name of the text and the relative frequency of the word in that text. By default, Voyant presents a list of the most frequent words in the corpus, excluding *stopwords*. Stopwords are words users might want to disregard in their analysis. Excluding stopwords is a common practice in Natural Language Processing (NLP), but rarely seen in CL. Normally, stopwords lists consist of function words such as prepositions and determiners, such a list is the default for Voyant. However, Voyant allows the user to use their own stopwords lists.

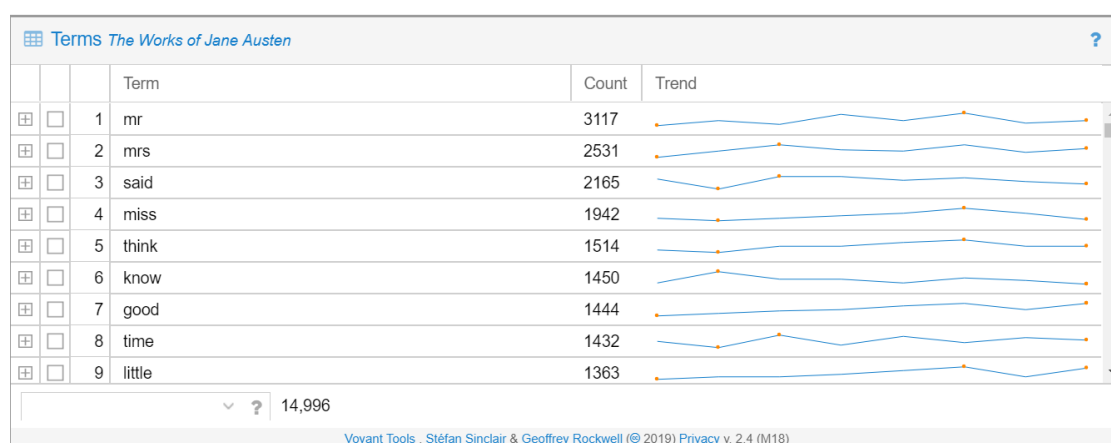


Figure 6.9: screenshot of Corpus Terms, in Voyant Tools

While *Corpus Terms* displays the relative frequencies for each word in a line, *Trends* shows these relative frequencies as a scatter plot (figure 6.10). The texts are

represented on the x-axis and the relative frequency is the y-axis. While in *Corpus Terms* the data for different words is placed one next to another (juxtaposition), in *Trends* the selected terms are plotted overlying each other. The superposition of different terms allows the users to have a clear view of the entire context – the corpus – without having to rely on their memory, as it is the case when they focus on specific parts of the corpus. This aspect will be further discussed in section 6.3.2. Each term displayed in *Trends* is represented by a line of a different colour; a legend is given above the plot. By hovering over the dots, the user can see the name of the text and the relative frequency. Clicking a dot opens a small window containing the corresponding corpus text appears, scrolled to the text section where the term first occurs. If the user double-clicks the dot, the display shifts to plot the distribution of the term(s) within the text. In this case, the x-axis lays out not the corpus texts, but the segments of the selected texts.

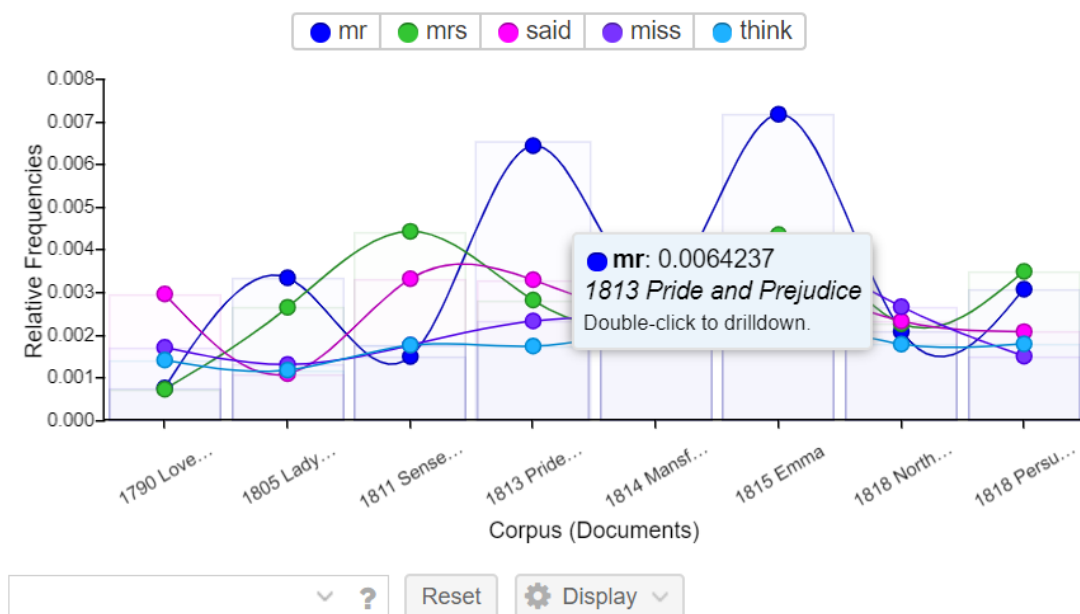


Figure 6.10: screenshot of Trends, in Voyant Tools

The two Voyant visualizations from take a step further than the previous visualizations here described. They actually graphically show the users how the terms

are distributed in the corpus. Yet, this web application was designed to address digital humanities (Sinclair & Rockwell 2020) and not specifically corpus linguists.

I have now considered the different ways in which dispersion (and distribution) is calculated and applied by some available software. The next section discusses the implications of these observations for the development of a new system for visualizing dispersion.

6.3 Criteria for a new visualization

Data visualization provides new ways to explore the data, as discussed in detail in chapter six of the full thesis. A primary consideration when developing this tool was on the need to create an environment which would support and empower the users, rather than put them off. If a user has a hunch and wants to further explore this hunch, the procedure to do so should be simple and straightforward and should minimize the time required for the operation. When establishing the criteria to develop this new visualization for dispersion, I considered three main issues: usability, functionality and implementation. These will now be discussed in detail.

6.3.1 Usability

Usability is a term that is associated with several concepts. It can be related to memorability, efficiency, satisfaction, and/or ease of learning, among others (Dubey et al. 2003). However, there is not consensus on the definition of this term, whether by scholars or by standardization bodies such as the International Organization for Standardization (ISO) (Abran et al. 2003). Although the term does not have a precise definition, a broad definition is given by ISO 9241-11, which defines *usability* as

the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. (ISO 9241-11)

Moreover, to reach a high level of usability, a product needs not only to enable the final goal, i.e. what the user wants to do with it, to be achieved, but in a way that affords the user a pleasant experience (Campbell et al. 2003).

A usable software product has the three main characteristics: learnability, efficiency and satisfaction (Shneiderman et al. 2017). Having a high *learnability* means that the user becomes competent on the first contact(s) with the software. For example, an absolute novice user of a well-designed concordancer should be able to quickly generate concordance lines for a query. *Efficiency* here means that the software should allow the users to easily achieve their goals by minimizing the effort the users have to put on it. So, for instance, actions that users are expected to perform frequently should be conveniently accessed. Software that produces *satisfaction* is that which gives the user an engaging and appropriate experience. A satisfying product attracts users, who will then feel positively towards the prospect of further user of the software (Norman 2004).

To develop a tool that displays these three characteristics, we need to have view of the users' goals and context of use. When it comes to CL users' expertise, there are normally two different perspectives. On one side, we have researchers who emphasize the importance of corpus linguists learning how to code and to create their own program for their research. The other perspective, much more often the reality (Paquot & Plonsky 2017), linguists struggle even with the use of ready-made software, whose features and functions have been created by a specialist developer.

From this perspective, it is crucial that users know and understand how the data is being processed and analysed. For instance, Gries (2010, 2013) argues that linguists should be familiar with programming languages such as Python and R, and that they should use these skills to customize their research procedures according to the needs of the research at hand. This has been his and other scholars' view for over twenty years (e.g. Biber et al. 1998; Baayen 2008). In the last fifteen years, there has been a noticeable increase in the number of language enthusiasts interested in promoting programming skills, as materials on programming for linguistic analysis are becoming increasingly more numerous and popular (e.g. Hammand 2002; Baayen 2008; Gries 2009; Levshina 2015). This growing number might also suggest that the available software does not entirely meet certain users' needs, who might want advanced applications.

But while some users might need or want new advanced tools for their research, many others have different needs. Tribble (2012, 2015) has conducted surveys among researchers and educators on their use and perception of CL tools in language teaching and learning. Two of his aims were to understand why people use, or do not use, CL techniques, and to understand what makes the interaction between user and CL tool successful or not. Tribble (2015) observed that the main reasons given by users, mainly teachers and applied linguists, for not using corpora were lack of knowledge, and enough time to learn the new skill. Among the reasons given to Tribble for the selection of one piece of software over another were preferences for the application that is user-friendly and cost-free. From the literature investigation (chapter four), it seems that this second group is larger than the group of researchers creating custom software for linguistic analysis. The explicit use of custom programming for textual

analysis appears in only a few of the articles found in that literature survey, most of which were in the field of psycholinguistics.

Based on these two different scenarios and on the findings in 5.5, I defined two main points to be guide the development of the visualization. (1) The tool should efficiently afford a certain set of research methods and techniques. If users want to explore further, the tool should provide them with advanced features. Hence, easily retrieving different dispersion measures is crucial. (2) To address the second group's needs, the tool should allow the user to quickly learn how to operate it.

6.3.2 Functionality

I considered three main user needs derived from the user investigation (see 5.5) when developing the functions to be included in the visualization. They were: to report the results; to compare and contrast different queries; and to work with subcorpora or specific corpus parts. When considering the features to add to the tool I also relied on the results from chapter four to assess whether users would be likely to use particular features in their research.

To address the need of reporting back the results, i.e. to be able to export results for inclusion in papers, essays, etc., I sought to include a function to download an instance of the visualization generated. This image can easily be incorporated in articles, presentations and so forth. To avoid loss of important information, the downloaded visualization should also incorporate a record of the steps performed to generate it.

A common approach in CL is to contrast results as means of analysis. Comparing how different words are dispersed in a corpus can give the user an insightful view of the corpus. Thus, another functionality for the visualization is that it must allow the

plotting of more than one search term at the same time. WordSmith Tools also allows the plotting of more than one query at the same time. It does this by adding each new barcode plot below those already present. This juxtaposition, placing objects separately in space, relies heavily on the user's memory (Gleicher et al. 2011). Although juxtaposition does allow relative comparison, when the dispersions are placed in the same space, users can have a better sense of the dispersion. This is because the comparison occurs within the same eye span. Thus, my visualization will allow comparison via overlay (superposition) of the multiple terms the user chooses to search for.

Using dispersion measures, there are no threshold values for whether a word is or is not well-dispersed. It is hard to state what magnitude of, say, DPnorm score gives a word the status of being evenly or unevenly distributed (Gries 2008). Plotting dispersion data for multiple queries on the same graph allows users to use the results from different words and phrases as reference parts. Therefore, my dispersion display will allow users to plot the dispersion of additional queries onto the graph generated for their initial query. Finally, because users might want to investigate dispersion in only a restricted part of the corpus, a restricted query for these additional plots should also be possible.

6.3.3 Implementation

As mentioned in 6.1, the new visualization tools I developed and implemented in CQPweb. One of the main reasons for this choice is that CQPweb is web-based. Among the several benefits of web-based software (see 3.4.3), I would highlight the following: easy sharing of data; quick access from any computer or mobile devices; no need for installation by the end user; and fast processing of the data. This last

characteristic is crucial. To make the tool efficient to the user slow processing must be avoided. Because in CQPweb the analysis runs on the server side, researchers who do not have a powerful computer are still able to work with large corpora (1 to 2 billion words) at an acceptable speed. Fast data retrieval means the users do not have to wait for the results, keeping them engaged with the process.

Another reason for choosing CQPweb is that it is open-source. This same criterium was sought when choosing the means of how to implement the visualization. For this, I chose the Data-Driven Documents JavaScript library (D3.js)²⁹ to create the visualizations. This library allows interactive and dynamic visualizations to be implemented in web browsers. D3.js has the benefit of being very fast, even when dealing with very large datasets. D3.js is also a powerful and flexible tool that allows for the implementation an immense variety of visualizations.

6.4 Prototypes

Before implementing the final version of the visualization, a series of prototypes were developed and presented to other linguists. The constant discussion with and feedback from my colleagues were essential to the design of the final version. This section deals with the main motivations and inspirations for each prototype; the thought processes which served as the basis for the design; results in terms of which elements worked, and which did not; and an account of how each prototype led to the next.

²⁹ <https://d3js.org/>

6.4.1 Mock corpus

The prototype visualizations were designed mock data and the programming language R. The mock data is the book *The Cat in the Hat* (Seuss 1957), consisting of 3,448 words across 38 chapters. Each chapter was treated as a different text. Using a real text meant I could test the feasibility of the visualization with corpus-like data. By using sample data, I could visualize, for instance, the contrast of considerably evenly dispersed tokens such as *the* with *sit*, which was not equally distributed in the text.

If I had designed the prototypes without any data, I could have risked failing to take into account important aspects of language behaviour. I opted to use a small book whose content I am familiar with, so I knew what to expect from the data.

6.4.2 Prototype one: parallel coordinates

The main idea behind this visualization was to help users understand how dispersion measures work. One of the difficulties that users might experience when dispersion is concerned is distinguishing or understanding the meaning of each of the several dispersion measures that can be used. The aim of the first prototype was to assist the user to visualize the different measures all at once, and to choose, based on this visualization, the measure best suited to their analysis.

With this aim, I chose a type of visualization known as *parallel coordinates*. This visualization is ideal for exploring multivariate data with many entries (Few 2009). One application for it is to identify clusters of observations with similar behaviour. Moreover, it allows the comparison of values in different units, such as the different dispersion measures. Lines are usually used in graphs to represent change, as in, for example, a time-series line graph. But in a parallel coordinate graph, lines are used to connect different numerical values for the same observed item. In my first prototype,

each line represents a word and each intersection a different dispersion measure (Figure 6.11). As the high number of lines makes it hard to actually see anything, the graph also permits user interaction to select relevant lines. For example, if a user wishes to see only words whose DP_{norm} is in a certain range, they can select this range in the graph and only those lines which passes through that range of DP_{norm} will be highlighted.

After showing it to colleagues, however, I understood that the difficulty in using dispersion measures was not in comparing and selecting the measure. From discussion and feedback, it instead became clear that most users struggle with the key concept of dispersion, even before moving to the task of choosing an appropriate measure. The users were not familiar with the different dispersion measures; hence they did not feel at ease when trying to interpret the graph. Even though there was a column for word frequency, the first impression of some of the users was that the rises and declines in the lines were showing variations in frequency across the texts of the corpora.

Although this graph could be used in a study aimed at comparing the different dispersion measures, I do not think it suits the purpose of this research. It did not seem to make analysis of dispersion any easier for novices and intermediate users of CL methods. But the prototyping process generated useful feedback for subsequent prototypes. Testing this first version showed me that users enjoy interacting with the graphs and have the concept of frequency at the forefront of their minds.

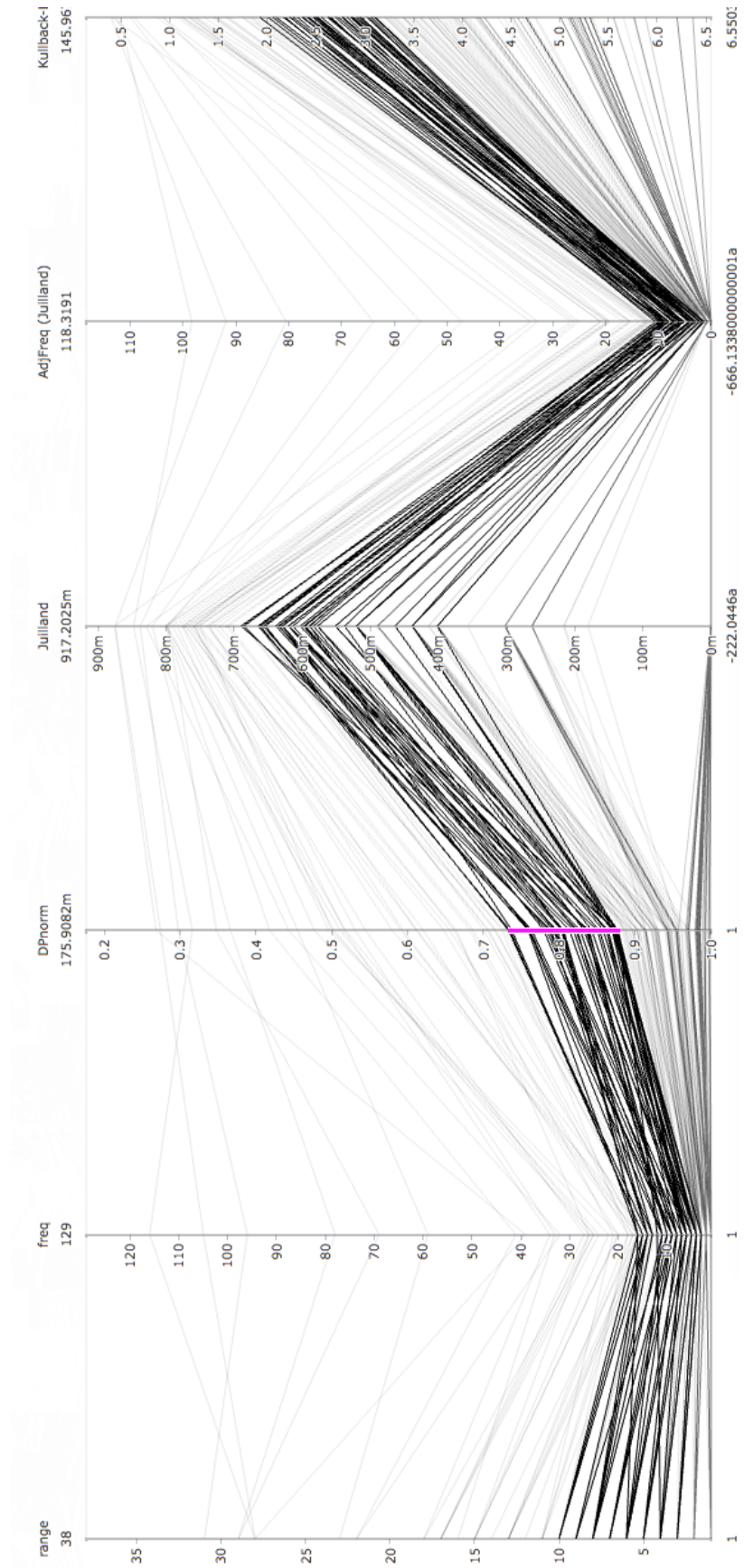


Figure 6.11: prototype one - parallel coordinates

6.4.3 Prototype two: histogram and scatterplot

Prototype two focus on helping the user grasp the concept of dispersion. It graphically displays the relative frequency of the word or the query item in each text of the corpus. Prototype two consists of two graphs, a histogram on the top and a scatterplot below it. As sample data, I used the British English 2006 (BE06) corpus (Baker 2009) instead of *The Cat in the Hat*, since BE06 is divided into categories (fiction, prose, etc.), which the prototype makes use of. Figure 6.12 shows that over 400 texts out of 500 do not contain the word ‘happy’. This prototype adds the standard CQPweb page header giving information about the query from which the dispersion data was generated.

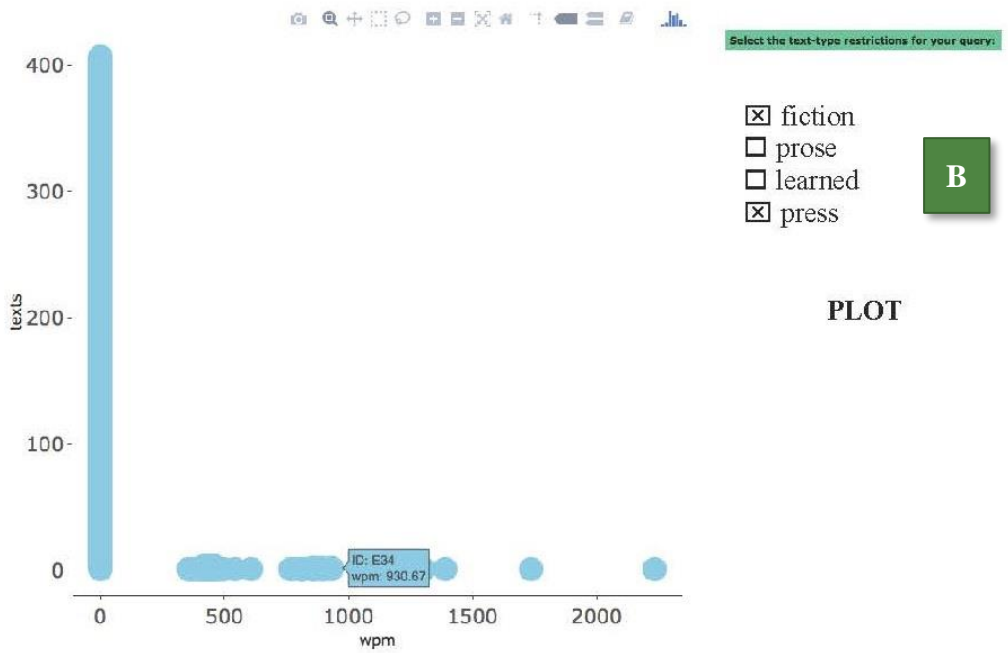
The histogram shows the distribution of texts according to the relative frequency of the item searched for (figure 6.12A). The bars for the histogram are made of dots, each representing a text. By hovering over a dot, the user can see a pop-up with the text ID and the exact relative frequency in the text. The histogram is also interactive. If the corpus is divided into categories, users can click to select on the categories they want to explore (figure 6.12B). Only dispersion data from texts in the selected categories will be displayed.

The scatterplot in the lower part of the display shows the relation between word frequency on the x-axis and range (on the y-axis), i.e. in how many texts the word occurs. Another dispersion measure (DP_{norm}) is encoded by colour intensity. The darker the colour of the label, the more evenly dispersed the word is. On the left-hand side, users can select words to be plotted onto this second graph (figure 6.12C). The words on the box are drawn from the user’s query history, so queries already

performed can be plotted together and then compared. CQPweb keeps a complete query history for each user, and this information is easily accessed.

Although prototype two was much better received than prototype one, this visualization did not fulfil the main goal of helping users understand dispersion in so as to be able to use it in their analyses. The users did not easily understand how the histogram conveyed the relative frequency. Because many texts did not contain any instance of the search term, the bar representing frequency 0 was the highest one. At first glance, this gave the users the false impression that the word was frequent in the corpus. Although the scatter plot seemed to be easier to grasp, the users did not seem inclined to use the tool in their research. According to their feedback, a table giving the dispersion measures numerically would be as useful as seeing the DP_{norm} via colour intensity in the scatter plot. Aspects of this prototype highlighted as positives were the possibility of accessing the query history, the capacity to restrict the query, and the ability to hover over dots to get more information on the texts.

Your query "happy" returned 127 matches in 93 different texts (in 1,147,097 words [500 texts]; frequency: 110.71 instances per million words) (1.932 seconds - retrieved from cache)



A

B

PLOT

Query history

- happy
- sad
- excited
- fine
- angry

C

PLOT

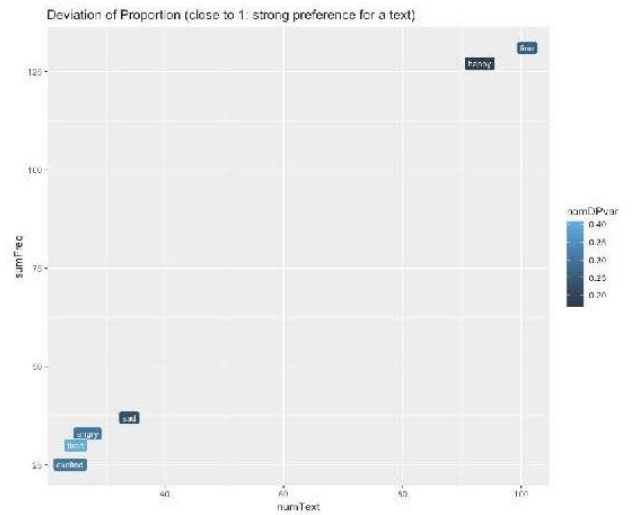


Figure 6.12: prototype two - histogram and scatterplot

6.4.4 Prototype three: time-series style

The idea of displaying the relative frequency of each text was preserved in prototype 3 (Figure 6.13). However, because using a histogram had been found to be potentially misleading, this format was not preserved. In this new prototype, the texts are plotted on the x-axis and their relative frequencies on the y-axis. Thus, all the textual frequencies are visualized independently. Since a corpus can easily have many more than 1,000 texts, the visualization allows the user to scan through the texts and to zoom in and out of specific regions interactively (figure 6.13A). The graph can show both an overview and also a detailed view of certain parts of the corpus. As in the previous prototype, a dispersion measure is displayed via a colour scale – the darker the colour of the, the more evenly dispersed the word is (figure 6.13B). In this prototype, the users can also choose which measure is displayed, DP_{norm} or Juilland's D (figure 6.13C). The controls for selecting corpus parts and allowing multiple plots were also preserved.

I observed my colleagues to be more engaged with this prototype. They observed it looked aesthetically better than the first two prototypes. One reason for that could be that, unlike the previous prototypes, which were made in R, this one was now made using the D3.js library (see section 6.3.3). One issue with this visualization, however, is that the lines linking the texts gave the false impression that the texts formed a continuum, which they do not. Another point made in feedback was that, although the variation in relative frequency gave the users an idea of how evenly dispersed each query is, this visualization did not allow the user to see the dispersion of instances within each text.

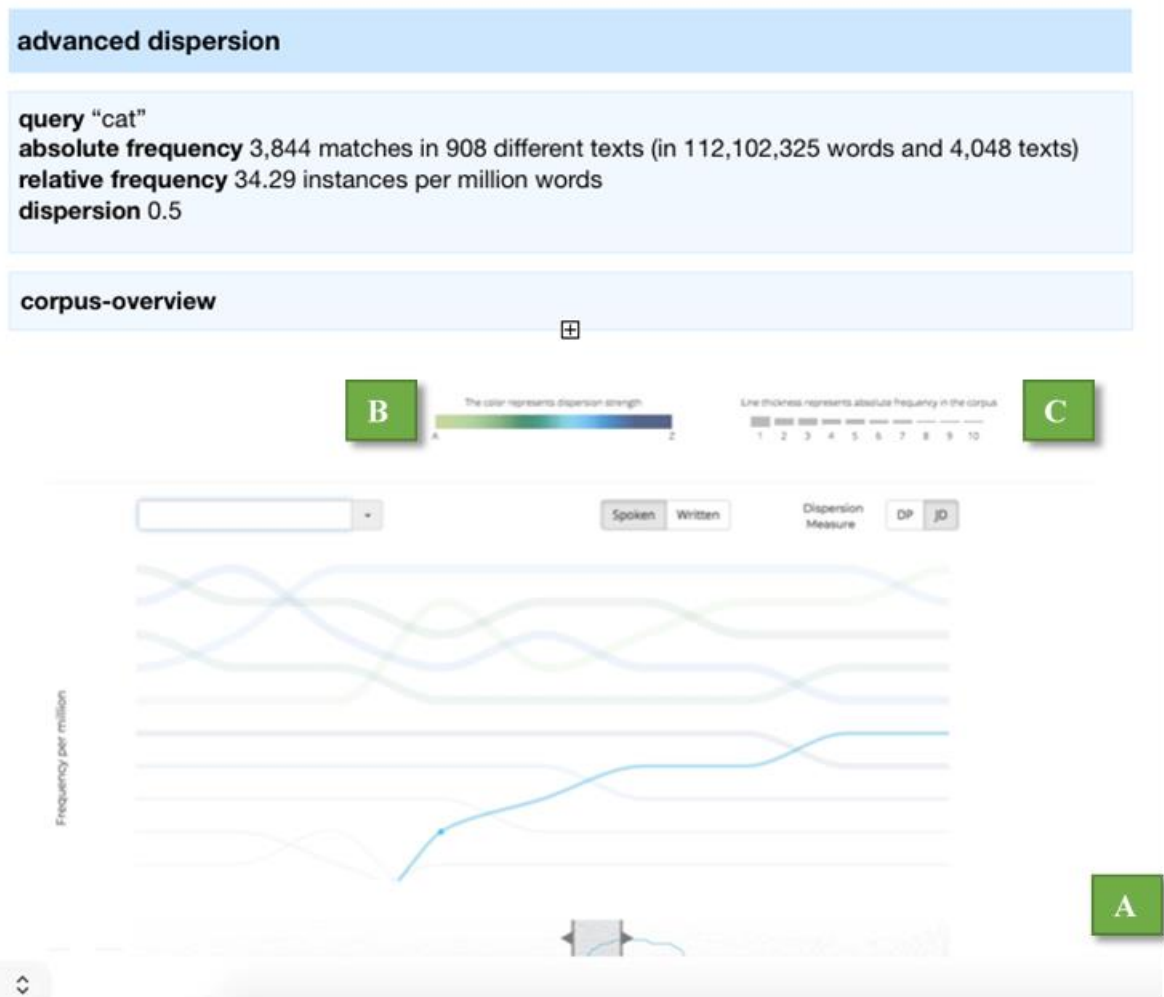


Figure 6.13: Prototype three - time series style

6.4.5 Prototype four: scatter-plot and barcode style

Prototype 4 (figure 6.14) preserves much of prototype 3. The multiple-query option based on user history, the relative frequency per text and the graphical display of the dispersion index are retained, but with some alterations. The multiple-query option row also offers the user the ability to plot the word types in the corpus with lowest and highest dispersion index, to serve as path to comparison (figure 6.14A). Some users suggested after seeing prototypes that having a point of reference would help in understanding how evenly dispersed a word is. As in the previous version the relative frequency is presented on a scatter plot with a scan bar is plotted on the side so that users can scan through all the texts (figure 6.14B). The user can click and drag the

graph to zoom in on specific parts of the corpus and zoom out for a general view. The dispersion measures in this version (Juilland's D and DP_{norm}) is represented by the size of the dot, instead of being coded through colour. The new functionality in this prototype is that when the users click on a dot, they add to the interface a display of the dispersion of the query term within the corresponding text.

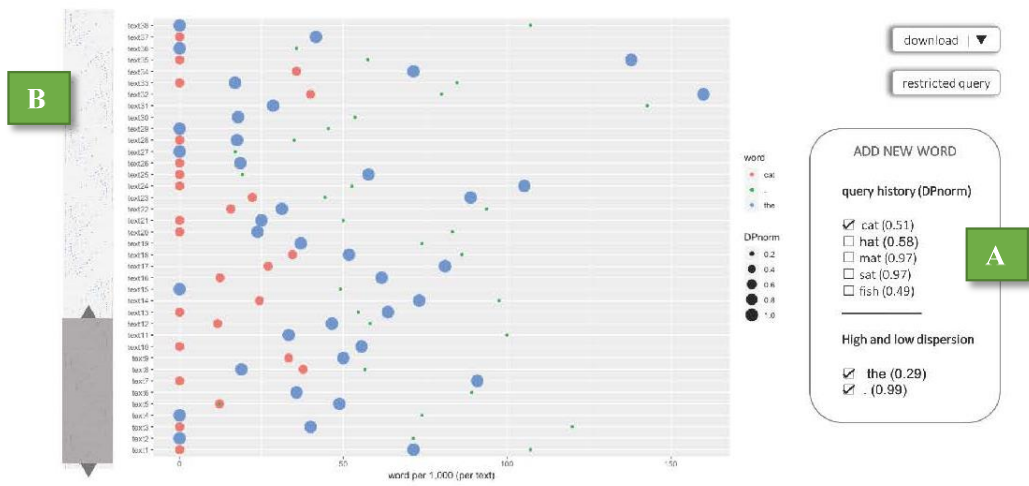
This is the first prototype that actually allows the users to visualise the dispersion directly, instead of the statistics that summarize it. Previous prototypes had hit problems with the complexity of the dispersion measures and how difficult it had proved to be for the users to grasp their meaning. But this visualization seemed to have supported the user to an understanding of the dispersion. The strategy of this prototype, which succeed on this front, was to break the complex formulae into simple objects that the users were already familiar with: relative frequency, corpus texts and position of tokens within the corpus. On the basis of these understandable elements, the users can then compare them and make their own interpretations.

As new dots are clicked, new barcode plots are displayed. As in the overview display, the text can also be scanned, zooming in and out, on different parts of the text. As in the tools discussed in 6.2, the horizontal rectangles represent the texts and the vertical lines represent the distinct occurrences of the search term in the text. The horizontal bar size reflects the number of tokens in the text, and the lines for each query are displayed in a different colour. Users easily understood the function of this visualization and expressed a high degree of interest in using it for their research. As this was the prototype with highest level of satisfaction, it served as the model for the actual implementation, as it will be discussed in the next section.

advanced dispersion

query "cat"
absolute frequency 26 matches in 17 different texts (2,062 in words and 38 texts)
relative frequency 12.61 instances per thousand words

corpus-overview



single-text view

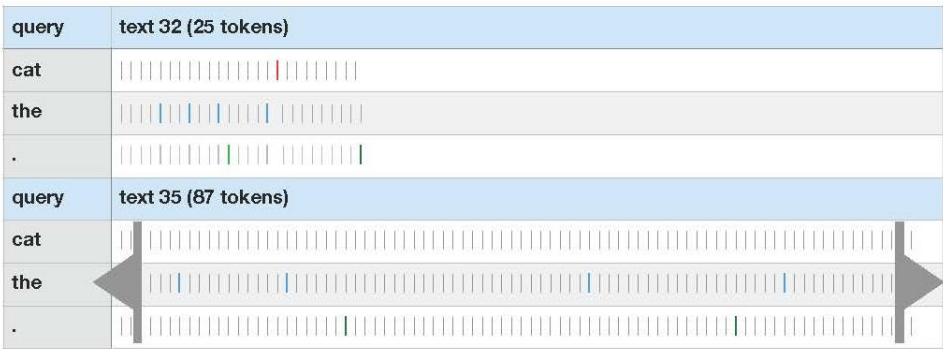


Figure 6.14: prototype four - scatter plot and barcode style

6.5 Implementation of the visualization

This section presents the visualization that I ultimately implemented after creating the prototypes and discussing them with a small number of users.

As with many of the functions in CQPweb (Hardie 2012), to use the advanced dispersion tool, the user needs first to perform a query to produce a set of concordance lines. From there, the user can then select “Dispersion” from the dropdown menu and go to the new page. From the same menu, the user can access pages to other functions such as *collocations*; *distribution*, showing how the query is distributed across the corpus section; and *frequency breakdown*, revealing the percentage of each different form of the query across the entire corpus (figure 6.15).

Your query "is" returned 2,691 matches in 66 different texts (in 334,998 words [66 texts]; frequency: 8,032.88 instances per million words) [0.056 seconds - retrieved from cache]

Navigation: |< << >> >| Show Page: 1 Line View Show in random order Go!

No	Text	Solution 1 to 50	Page 1 / 54
1	According to Jim 1_04	that 's a hundred and fifty bucks . <DANA> Jim , this is your tenth annive	
2	According to Jim 1_04	should n't have . <JIM> Damn right . The only guy touchin' me is gon na be me . A	
3	According to Jim 1_04	<DANA> Perfect . <CHERYL> Excellent . <JIM> Well you know that serenity spa sure is nice but it 's not a	
4	According to Jim 1_04	honey I just ca n't wait . Every year your anniversary gift is better than the yea	
5	According to Jim 1_04	<JIM> Well this year 's gift , honey , this year 's gift is , wow ! <CHERY	
6	According to Jim 1_04	a slightly higher retail value than last year . <CHERYL> Oh , he the best ? <DA	
7	According to Jim 1_04	<CHERYL> What 's with her ? <JIM> Who cares . Mmm , this is great . Is this a cr	
8	According to Jim 1_04	with her ? <JIM> Who cares . Mmm , this is great . Is this a crayon ? <A	
9	According to Jim 1_04	you 've been married . <JIM> How much ? <DANA> See like , this is from when you pr	
10	According to Jim 1_04	like , this is from when you proposed , and this one is from that Stones concert . <JIM> How much Dana ? <DANA>	
11	According to Jim 1_04	'm already gettin' the milk for free . <DANA> Hey , that cow is my sister . <JIM> Dana you broke the rule . The rule is	
12	According to Jim 1_04	is my sister . <JIM> Dana you broke the rule . The rule is fifteen dollars a year . This , this bracelet is like ...	
13	According to Jim 1_04	The rule is fifteen dollars a year . This , this bracelet is like ... <DANA> Jim . <JIM> Hold on . It 's like a thirty-three	

Figure 6.15: screenshot of dispersion - step 1

Upon selecting *Dispersion* and hitting the button *Go!* the user is taken to the dispersion page (figure 6.15). As standard in CQPweb, a heading row with the information on the query is given at the top of the page. The visualization is composed of two parts: dispersion overview and single-text dispersion. Each of the two parts is explained in detail below.

6.5.1 Dispersion Overview: text frequency

As its name suggests, this part of the display provides the user with an overview of how the search term is dispersed across the corpus (figure 6.16). This overall visualization can serve as a tool to identify discrepancies in relative frequency. For

example, a word with a much higher frequency for one or a few texts, relative to the rest of the corpus, is not evenly dispersed. Hence, this burstiness is very readily perceptible. The relative frequency of the query result in each individual text is given on the y-axis and the text names are plotted on the x-axis. As is the norm in CQPweb (Hardie 2012) and other pieces of software such as Sketch Engine (Kilgarriff 2004), which arises from longstanding practice in CL, the relative frequency is given as a number of instances per 1,000,000 tokens.

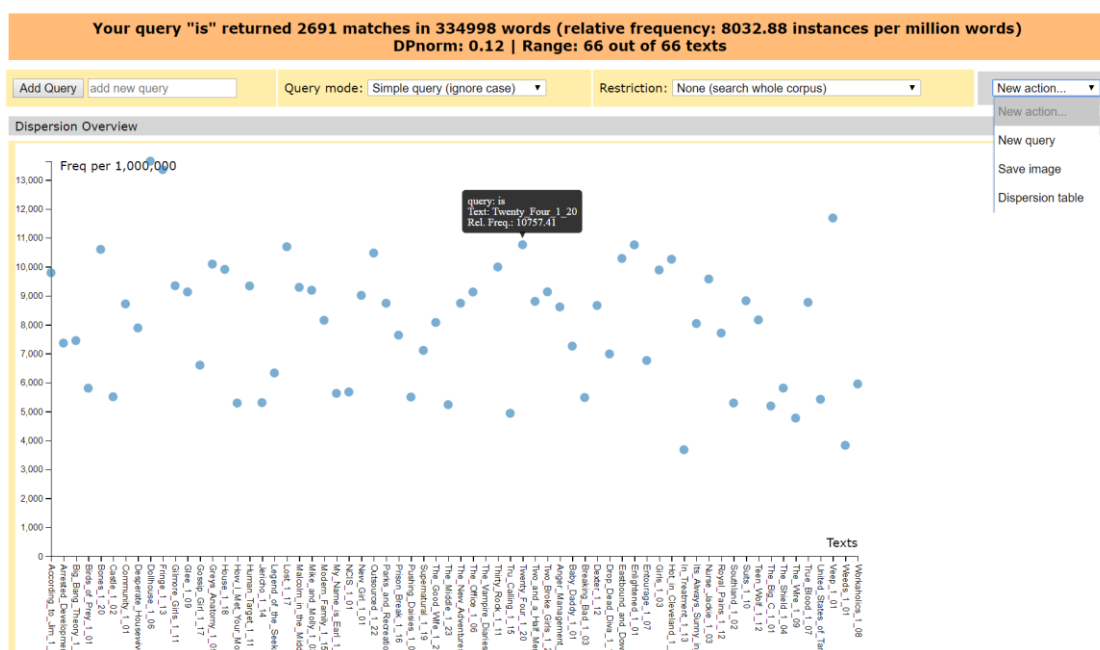


Figure 6.16: screenshot of dispersion overview

New query data can be added via horizontal bar above the plot. The user can perform simple queries or use CQP syntax (cf. Evert & Hardie 2011). The queries can also be performed with restrictions, i.e. on different corpus parts. As new queries are performed, new dots are added to the plot (Figure 6.17). Each new query’s data is represented by a different colour which added to the legend that appears on the top right-hand side of the graph. The DP_{norm} for each query is provided next to the query so as to allow a quick view. Hovering over an item in the legend, pops up a box containing the scores for Juilland’s D and range.

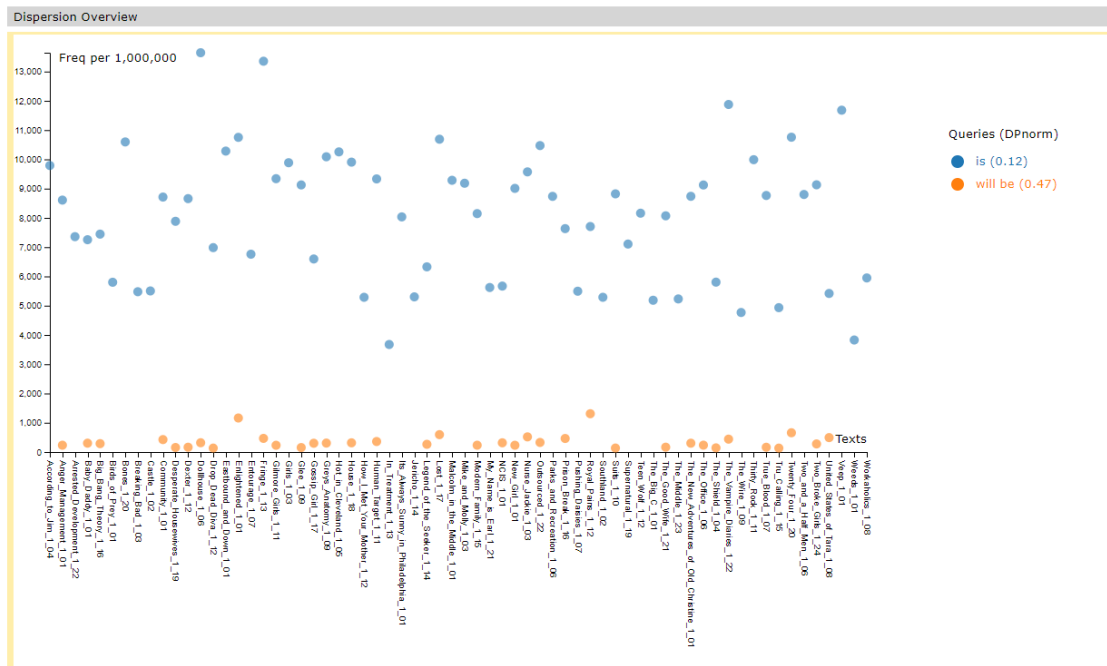


Figure 6.17: dispersion - tool multiple queries

6.5.2 New query and new action menu

The user can also click on the drop-down menu on the right-hand side and download a file with different dispersion measures for the searched terms or queries. The reason for this additional method of retrieving the dispersion measures is to guarantee replicability and to keep the visualization clean. The *Save image* option downloads the image generated for the dispersion overview (Figure 6.16). Finally, the user can click on *new query* to go back to the corpus main page, i.e., the initial query user interface.

The number of parameters that can be implemented were kept to a minimum so as to make the interface as simple as possible. Instead of producing a table with statistical measures, the users can now see the variation for two important variables when calculating dispersion: frequency per text and range (by showing how many texts have a respective dot on the graph). When hovering over each of these circles, a box showing the text ID code and the relative frequency appears. Clicking a circle causes a single-text view to be plotted below the corpus overview.

6.5.3 Single-text dispersion

In the single-text dispersion display, each individual result for the selected query is plotted as a circle on the x-axis, at the point where it appears in the selected text (figure 6.18). The width of the bar represents the sequence of token positions. The reason for choosing a circle instead of the bars (figure 6.19) observed in other tools (see section 6.2) is because the overlapping semi-transparent circles makes it easier for the user to identify where the query hits are piling up. The interface allows the user to scan through different parts of the texts and zoom in and zoom out of these selected parts, moving between a general view and a focus on specific parts of the text. New layers of circles can be added as the user clicks on other words plotted on the corpus-overview, allowing use among different texts to be compared. This zoomable display supports qualitative analysis, as it is the direct data, not a statistical summary as relative frequency and range are.

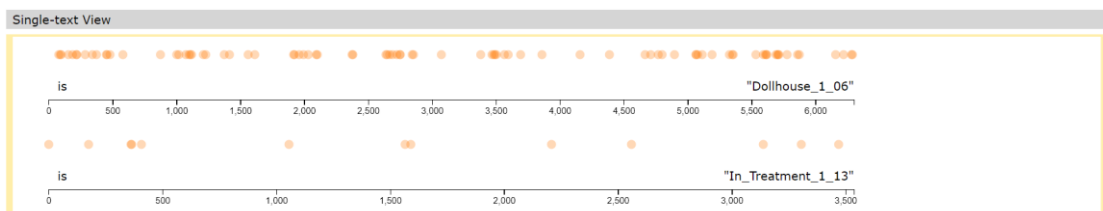


Figure 6.18: dispersion - text view function

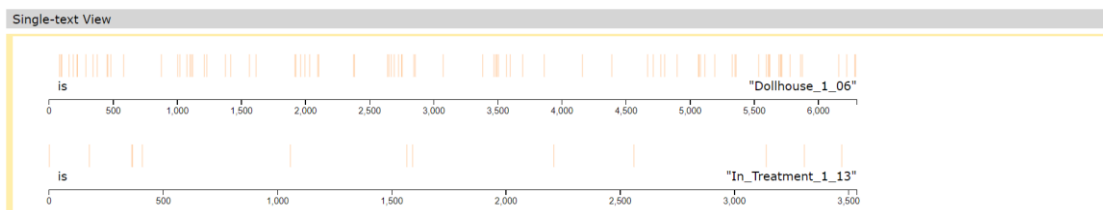


Figure 6.19: dispersion - text view with bars

6.6 Conclusion: observations and next steps

I do not claim that this visualization displays the full behaviour dispersion and dispersion measures. Factors such as the size of each text and the number of times the searched term occurs in each text (also allowing for comparison among them) have been made easily accessible through a few clicks. I believe this will make researchers using the software more likely to consider dispersion in their analyses. From that, I expect, for example, that researchers that solely rely on frequency to perform their analysis will, after using the dispersion tool, consider that other factors that impact the results. Easy availability of dispersion should make it easier for researchers to take it into account. Moreover, a striking visualization of what a big effect dispersion can have – by really making concrete the differences – can bring to the user the importance of considering dispersion as well as frequency.

Another expected result is that, by exploring the data through the restricted search and immediately seeing the consequences of selecting different corpus parts, users will be more conscious when creating their own corpora. After understanding that retrieving specific parts of the corpus might be crucial to certain analysis, they might start better documenting and annotating their corpora, instead of analysing any corpus as whole chunk of data, ignoring that, in some cases, there are some hierarchical characteristics.

7 Parlink: a tool for parallel corpora

7.1 Introduction

As discussed in chapter three, there is still an enormous gap between the potential of Corpus Linguistics (CL) methods and the methods adopted by researchers and students. There is still a great need to enhance education on linguistic data analysis practice. One way to bridge this gap is to provide underused (or new) sophisticated data analysis methods via user-friendly interfaces. Delivering data and tools in such a way has the benefits of attracting more users and guiding them through accurate analysis.

This chapter will present a new visualization to perform parallel corpora studies and discuss the steps taken to achieve it. The first section deals with a literature review of the state-of-the-art parallel corpora analysis. The third section discusses why this is a promising and in need of improvements area of CL. Section four deals with the development and the logic behind the newly developed tool. Section five presents some possible further implementations to the tool.

7.2 Parallel Corpus Linguistics: a literature review

The use of Parallel Corpora is one application of CL that has become more prominent in the last decades (Guinovart 2019:355). This use has proven effective in a plethora of fields, such as translator training; lexicography; language teaching and learning; contrastive studies; computer-aided translation and machine translation (Aijmer & Altenberg 1996:12; McEnery et al. 2006:46; Doval & Sánchez 2019:10; Guinovart 2019:355).

Despite the increase in popularity, parallel corpus studies still use only on a small portion of their potential. The reasons for this limitation are many. For instance, in the field of language teaching, there is a belief that the existing parallel corpora offer a variety of language that can be very difficult for the students to find meaningful patterns (Doval & Sánchez 2019:10). Another issue is the lack or difficulty in access to tools to process parallel corpora other than concordancers (Rabadán 2019).

7.2.1 Key Terms

It is debatable whether Parallel Corpus Linguistics is a field of research within CL or only a methodology (Borin 2002:1). For simplicity, here, Parallel CL is a methodology which deals with the process of comparing and contrasting two corpora. For that, we need a *source corpus* (SC), which is the one to be queried, and its equivalent corpus, the *target corpus* (TC). For this work methodology is more critical than corpora content, hereafter SC refers to the queried corpus and TC to the parallel corpus.

The degree to which a corpus is parallel or not varies. For this reason, different authors offer different types of classification. A frequent and straightforward classification is to distinguish between *parallel corpora* and *comparable corpora*.

Two corpora can be described as parallel if they contain source texts and the equivalent translations in another language. Comparable corpora, on the other hand, are texts in different languages sampled with the use of the same techniques (McEnery et al. 2006:46). For obvious reasons, comparable texts are much easier to be found and collected than parallel ones. Hence, many more comparable corpora are available – especially for low-resource languages (Gamallo 2019:251). However, extracting meaningful information from those corpora is much more challenging and complex when compared to an analysis of parallel corpora.

To easily explore two parallel corpora at the same time, they need to be aligned, with comparable *aligned zones* or regions (AZ). These zones can be segments such as words, sentences or paragraphs (Hewavitharana & Vogel 2013). In linguistic studies, the most common AZ type are *sentence alignment*, while in Natural Language Processing (NLP) *word alignment* is more frequently used (Tillmann & Hewavitharana 2013). Word alignment helps with the identification of words and multiwords and their respective translations. When word alignment occurs, it usually follows the identification of aligned sentences and their identification process is usually more demanding (Gamallo 2019).

Alignment can be done automatically, manually or with a combination of both. Automatic aligners have an average accuracy rate of approximately 97% (Zariņa et al. 2015), varying according to the nature of the data and to the language pair (relatedness and script used). For being time-efficient, they are ideal for large data. However, when seeking total or nearly total accuracy, a common practice is to automatically align the corpora and then manually edit this alignment to ensure accuracy (Guinovart 2019). Most of the alignment tools available offer a sentence alignment system, as they are

the best-established level used for parallel corpora (Tiedemann 2011:37, Volk et al. 2014). Some frequently used alignment tools are the Giza++ (Och–Ney 2003); Vanilla aligner (Gale & Church 1991); and LF-Aligner (Doval et al. 2019:108).

Within the linguistic community, LF Aligner has one of the highest accuracies (Doval et al. 2019:109). LF Aligner is a Graphical User Interface (GUI) of the Hunalign Sentence Aligner (Varga et al. 2007). Hunalign tokenises and segments texts into sentences for two comparable texts. It features a hybrid process in which both length-based and lexical matching approaches are used (Tóth et al. 2008; Varga et al. 2007). If a dictionary is provided, Hunalign will use this information combined with the Gale-Church alignment algorithm. This method parts of the assumption that sentences in parallel corpora should have approximately equal lengths (Gale & Church 1993). If there is no external source, Hunalign proceeds as follows. First, the Gale-Church method is applied to produce a sentence alignment. From this first alignment, a bilingual lexicon is created. The final step is then to rely on both the lexicon recently created and the sentence length method to then create a second and more accurate alignment (Varga 2012:92-119). The positive reception of Hunalign’s GUI, LF Aligner, might be due to its simplicity in use and to the fact it does not require any external resource.

For obvious reasons, a total correspondence of AZs in parallel corpora is not always achievable, as languages express different contents in different ways. Sentences in the TC might be reordered, split or merged. Tools such as the Hunalign can deal reasonably well with split or merged sentences. However, automatic aligners cannot deal well with crossing alignments, i.e. when AZs in the TC are in a different order from the SC. A common way to address these limitations is to encode the texts in

Translation Markup Language (TML), the standard format used for aligned texts (Guinovart 2019). TML is a simple markup language that supports the identification of zones that are in a different order or inexistent in one of the parallel corpora.

7.2.2 Current Methods

As aforementioned, parallel corpora have multiple purposes; being translation-related work the most common of their application. Typical applications of CL applications to translation are translator training (Zanettin 1998; Doval et al. 2019:104); research on translation universals or features translations (Baker 1993, 1995; Laviosa 1997; Olohan & Baker 2000; Olohan 2004); and linguistically oriented translation research (Rabadán 2005). Effectively, parallel corpora help with terminology extraction (Alcina 2011); identification of language meaning and use according to the context (Heid 2008); gain in foreign language expertise (Doval 2018:182; Bernardini & Ferraresi 2013); among others.

Techniques for corpora investigation and analysis might vary according to the applications and the resources available. The use of data other than the corpora themselves is a widespread practice in parallel corpora studies (Zanettin 2012). In many cases, these external resources are lists of bilingual expressions compiled based on bilingual dictionaries (Gamallo 2019). For matters of clarity, the next section discusses the applications in two groups: (i) CL for translation practice and (ii) for translation theory.

7.2.2.1 Translation Practice

Corpora effectively contribute, even if indirectly, to the translator toolkit (Beeby et al. 2009). They are used, for instance, to inform the creation of dictionaries and termbases. Termbases, also known as glossaries, are databases with words and their

equivalent translations in the target language. Although dictionaries and termbases provided a significant amount of abridged information, they rarely offer contextualized examples of use (Bernardini & Ferraresi 2013:35).

A more resourceful tool in this aspect is Translation Memories (TM). TM systems provide an archive with units, usually a sentence, previously translated by a human. TMs help translators by reducing translation time and assuring consistency across the texts (Reinke 2018). Despite being a handy tool, TMs have shown a negative impact on the translator's strategies, as it provides only previous translated combinations, increasing the choice for less appropriate translations (Bernardini & Ferraresi 2013:2).

A reduced number of translators also rely on corpora to create their own terminology lists. They use corpora as a reference to identify the best translation for terms often used in their field of expertise. Hence, as in TMs, terminology lists include terms in a source language and the preferred translations (Guinovart 2019).

7.2.2.2 Translation Theory

Corpus-Based Translation Studies (CBTS) are often, but not necessarily, applied in quantitative explorations of literary translations. A common goal in this field is to verify if the translation of a piece of work is consistent (Patton & Can 2012:227). Many studies use comparative stylometric analysis by measuring features such as vocabulary richness to compare different translations of the same source text (Rybicki 2012:231). Conventional methods used in CBTS are type-token ratio (TTR); word and sentence (mean) lengths; frequency lists and keyness (e.g. Patton & Can 2012).

Tests such as student T and Chi-squared are sometimes performed to verify if the difference is significant (Ji & Oakes 2012). The chi-squared test is commonly applied "to find the most typical vocabulary of one corpus as opposed to another" (Oakes

2012:127-128), which also has its limitations, such as producing an excess of significant results (Bestgen 2017). One way of addressing those limitations is to add any effect size measure such as the Yule's Q measure (Oakes 2012:130); however, this makes the process even less straightforward for ordinary users.

Another technique often used for lexical investigation is the *distributional similarity*. This technique draws on the hypothesis that a pair of words found in similar linguistic context are likely to be semantically related (Gamallo 2019: 254). This method aims at identifying if a word₂ in the TC has a similar distribution to a word₁ in the SC when comparing equivalent AZs. This identification is achieved through the contrast of the distribution of the words in a bilingual list (Gamallo 2019:256).

Distributional similarity or distributional information (Garcia et al. 2019:268) is often used to identify semantically similar words and their respective translations. The identification is also often expanded to the collocation with these words and their variation according to the context (e.g. Smadja 1992; McKeown et al.; Kupiec 1993; Lewandowska-Tomaszczyk 2012). Examples of applications of this technique are enhancement of bilingual lists and thesaurus and identification of word meanings according to the context of occurrence (Gamallo 2019). A drawback of this method is the need to rely on external bilingual resources besides the corpora being used. Bilingual lists are not available for all combination of language variations. On the rare occasions they are available, they are not fully representative (Gamallo & Garcia 2012). Moreover, the plurality of meanings for a word or phrase makes it impossible to construct a bilingual list that suits different corpora.

Other more complex, and for this reason less used methods are logistic regression (e.g. Gries & Wulff 2012); discriminant analysis (e.g. Patton & Can 2012:219); and cluster

analysis (e.g. Ji 2012). They are used to make classifications and recognize patterns. However, those are not easy methods to apply. They “involve a transformation of the data” and do “not come easy to beginners” (Gries & Wulff 2012:39).

Most of the methods used in translation theory studies are applied to specific corpora and provide detailed comparisons of linguistic features via the application of tests like Pearson’s *r*, Spearman’s rank correlation, Wilcoxon’s signed-rank, Mann Whitney U test (Ji & Oakes 2012). Although efficient for their specific goals, they are (hard to apply) methods rather than tools to aid the work of novice researcher or translator.

7.2.3 Prominent tools and data

Because in many cases it is difficult to separate data from software, this section will cover some notable projects and software dealing with parallel data, discussing the tools, rather than the data, in more details. The tools and data discussed in subsections 7.2.3.1 and 7.2.3.2 are used mainly in translation theory, while subsections 7.2.3.3 and 7.2.3.4 discuss tools used in translation practice.

7.2.3.1 Open Corpus Workbench

A frequently used piece of software for corpus analysis is the IMS Open Corpus Workbench (CWB). CWB is a powerful open-source toolkit used to query and manage large corpora and their linguistic annotations (Evert & Hardie 2011). It also allows for the alignment and query of comparable corpora. CWB is used as a data access layer (back-end) for many tools used for parallel corpora. For instance, The *Corpus Valencià de Literatura Traduïda* (COVALT)³⁰ project uses CQPweb (Hardie

³⁰ <http://cwbcovalt.xtrad.uji.es/cqpweb/>

2012) to query over four million words of multilingual parallel corpora in English, French, German and Catalan (Molés-Cases & Oster 2019). CQPweb allows the users to perform a query in the SC and see the resulting concordance lines of the aligned target corpus or corpora (figure 7.1).

The screenshot shows the CQPweb interface with a search query for the word "appreciate". The results are displayed in a table with columns for "No" and "Text". The text shows concordance lines between an English corpus and an Italian corpus. The word "appreciate" is highlighted in blue in both languages. The interface includes navigation buttons, a search bar, and a "Show Page" dropdown.

No	Text
1	for the competent authorities to be informed about the operations . I appreciate that/IN it is not easy to lay down practicable criteria for the Capisco che non è semplice definire criteri stabili per il concetto di « trasporto particolarmente pericoloso » .
2	Channel Tunnel but also - as I am sure the rapporteur will appreciate - other fixed links such as those between Denmark and Sweden . Non solo , quindi , la galleria della Manica , ma anche - e sono certo che il relatore lo apprezzerà - altri collegamenti fissi come quelli fra la Danimarca e la Svezia .
3	transport of dangerous goods have developed separately and diversely . We naturally appreciate that/IN harmonization cannot be achieved overnight , We therefore provided for a Comprendiamo , ovviamente , che l'armonizzazione non può essere raggiunta da un giorno all'altro .
4	Madam President . Mrs Oomen-Ruijten , as you were able to appreciate this morning , we are introducing a system which is , let (no alignment found)
5	him to be here for this debate . Of course , I appreciate that/IN he has an able representative in Fritz Brüchert , who is Naturalmente so che può contare su un rappresentante di grande valore come Fritz Brüchert , che è presente , ma speravo davvero nella partecipazione del Commissario , visto che dobbiamo trattare un punto così importante .
6	that/IN the rapporteurs at least can speak beforehand . I would greatly appreciate your confirmation on this point . My fear is that/IN we will Le sarei grata se mi confermasse questi punti .
7	for example by budgetizing the Development Fund . The House would appreciate a response to what we have advocated , so that/IN then we Avremmo gradito ricevere una risposta in Parlamento a queste affermazioni per sapere come comportarci .
8	not quite in the order indicated on the agenda , but we appreciate the reasons for that . It is impossible to speak on them Signora Presidente , onorevoli colleghe e colleghi , sono state presentate otto relazioni sul disarcico , in ordine leggermente diverso da quello previsto nell'ordine del giorno , ma per ragioni

Figure 7.1: concordance lines for parallel corpora using CQPweb

Possibly the most massive multilingual parallel corpora freely available, OPUS³¹ (Tiedemann 2012) also has CWB running on its back end. To date, it has over 250 language dialects from the most various sources. There are 42 corpora covering genres such as newspaper texts, Wikipedia, subtitles and parliament texts. Despite its impressive size and data richness, OPUS has the shortcoming of not having a very user-friendly interface. Still, it features some handy functions. For instance, it is possible to display the concordance lines for more than one aligned corpus at the same time, as shown in figure 7.2. OPUS also has a word alignment database with a built-in query tool (figure 7.3).

³¹ <http://opus.nlpl.eu/>

OPUS - Corpus query (CWB)

corpora

(Books) (CAPES) (DGT) (DOGC) (ECB) (EMEA) (EUbookshop) (EUconst) (Europarl) (Europarl3) (Finlex) (GlobalVoices) (KDE4) (KDEdoc) (MBS) (MPC1) (MultiUN) (News-Commentary1) (OfisPublik) (OpenOffice3) (OpenSubtitles) (**OpenSubtitles2018**) (PHP) (ParaCrawl) (RF) (SETIMES2) (SPC) (SciELO) (TED2013) (TEP) (Tanzil) (Tatoeba) (TedTalks) (UN) (WMT-News) (WikiSource) (Wikipedia) (XhosaNavy) (ada83) (fiskimo) (lurenWic) (wikimedia)

languages

af ar bg bu br bs ca
 cs da de el eo es et
 eu fa fi fr gl he hi hu
 id is it ja kk ko lt lv mk
 ml no pl pt pt_br ro
 ru si sk sl sq sr sv ta
 te th tl tr uk ur vi ze_en
 ze_zh zh_cn zh_tw

CQP query (CWB) show attributes alignments

A CQP query consists of a regular expression over *attribute expressions*.
 Introduction of the query syntax
 Example queries

[word="drive"] word positional annotation

show max 20 hits vertical KWIC horizontal
 (advanced search)

Query string: '[word="drive"] :IT [] :PT []'
 20 hits found

167071551	uld you have had a spare tyre ? Somebody 'd better drive her . Go ahead . I 'll take the truck . Oh , Iz .	E meglio che qualcuno la accompagni .
pt		Algum tem de a levar .
490093662	the wrong side of the street . - What side do you drive on ? - On the right side . - I didn 't know that	- Da che lato guidate voi ?
it		- Em que lado da estrada guiam ?
509264365	in the Thermos . I 'm too young for coffee . Can I drive ? Thea ! Sometimes Dad puts me in his lap and let	Sono ancora piccola per il caffè ' .
it		Posso guiar ?
745462388	y . I 'm going to try slunting it through the warp drive . Won 't that overload the relay ? Not if we bypa	Cercherò di deviarlo tramite il motore di curvatura .
it		Vous tenter encaminhá-lo pelo impulsor warp .
pt		
783342180	e weapons who knows what they 've done to the warp drive . Hail them . No answer . They 're trying to disa	Chissà cos ' avranno fatto alla propulsione a curvatura .
it		Se melhoraram as armas , que terão feio aos motores warp ? Contacte-os .
827608076	little longer . Direct synaptic stimulation might drive out the alien presence . I was right . I heard Tu	Una stimolazione sináptica directa potrebbe respingere l ' alieno .
it		Uma estimulação sináptica directa poderia expulsar a presença alienígena .
pt		
940982252	If you start in on the war metaphors . I 'm gonna drive this car into a telephone pole . I am as worried	Se comincis con le metafore de guerra , vado con l ' auto
it		
pt		Se começas com as metáforas bélicas , vou contra um poste .

Figure 7.2: multiple languages concordance lines (OPUS)

af / amh / ara / aze / baq / ben / bos / bre / bul / cat / cze / dan / dut / ell / eng / epo / est / fin / fre / geo / ger / gle / glg / heb / hin / hrv / hun / hye / ice / ind / ita / jpn / kaz / kor / lav / lit / mac / mal / mlt / msa / nor / per / pol / por / rum / rus / scc / sin / slo / slv / spa / sqi / swe / tam / tel / tgl / tha / tur / ukr / urd / vie / zho

OPUS: Search Word Alignment Database for eng

afr ara baq ben bos bre bul cat
 cze dan dut ell epo est fin fre
 geo ger gle glg heb hin hrv hun
 hye ice ind ita jpn kaz kor lav
 lit mac mal mlt msa nor per pol
 por rum rus scc sin slo slv spa
 sqi swe tam tel tgl tha tur ukr
 urd vie zho

- results from automatic word alignment
- wildcard symbols '%' and '.' allowed
- click on translations to query these words with their alignments
- click on frequencies to get concordance lines from the corpus (max 100)
- the concordancer does not use word alignment

honey from all Books DGT Europarl News-Commentary11 OpenSubtitles2018 SETIMES2 TED2013 Wikipedia

#	prob	dut	>>
1733	(0.42)	honing	👍👎
18	(0.28)	honing	👍👎
7446	(0.24)	schat	👍👎
4666	(0.20)	lieverd	👍👎
51	(0.17)	Honey	👍👎

#	prob	fre	>>
4197	(0.44)	miel	👍👎
19648	(0.30)	chérie	👍👎
10598	(0.22)	chéri	👍👎
51	(0.13)	Honey	👍👎
2	(0.06)	honey	👍👎

#	prob	ger	>>
1970	(0.48)	Honig	👍👎
12237	(0.26)	Schatz	👍👎
3156	(0.15)	Schätzchen	👍👎
51	(0.15)	Honey	👍👎
2882	(0.11)	Liebling	👍👎

#	prob	ita	>>
3305	(0.41)	miele	👍👎
32651	(0.32)	tesoro	👍👎
51	(0.13)	Honey	👍👎
28	(0.06)	Miele	👍👎
14	(0.06)	mellifere	👍👎

#	prob	por	>>
3620	(0.47)	mel	👍👎
21372	(0.25)	querida	👍👎
10398	(0.20)	querido	👍👎
51	(0.13)	Honey	👍👎
44	(0.10)	Mel	👍👎

#	prob	spa	>>
6695	(0.48)	miel	👍👎
48743	(0.30)	carriño	👍👎
2	(0.15)	mieles	👍👎
71	(0.13)	honey	👍👎
2	(0.07)	carriña	👍👎

Figure 7.3: a screenshot of OPUS word alignment database tool

ACTRES Parallel Corpus (P-ACTRES 2.0)³² is another project that also relies on CWB. It is an English-Spanish corpus with over 4 million words (Sanjurjo-González

³² <http://contraste-test.unileon.es/demos/demos/p-actres2/demo.html>

& Izquierdo 2019). Like CQPweb and OPUS, the ACTRESS software allows the user to perform online queries in a corpus and see the results with the aligned corpus. ACTRESS software has the additional feature of query composer. It allows the user to easily construct advanced queries for both source and target corpora (figures 7.4 and 7.5).

Figure 7.4: ACTRES query composer

Num	Original (English)	Translation (Spanish)
1	This energy of motion was perceived as a measure of vis viva, or living force, and regarded as a sensible measure of the vigour of the events taking place in a collection of particles. (EAP1E.s143)	Esta energía del movimiento se percibía como una medida de vis viva, o fuerza viva, y se consideró una sensata medida del vigor de los sucesos que tienen lugar en una serie de partículas. (EAP1S.s166)
2	A lot of stored energy (a heavy mass moving rapidly) can in principle do a lot of work-raise a heavy weight through a great height. (EAP1E.s173)	En principio, mucha energía almacenada (como una masa pesada que se mueve deprisa) puede realizar mucho trabajo, como levantar una pesa pesada hasta una gran altura. (EAP1S.s198)
3	A lot of stored energy (a heavy mass moving rapidly) can in principle do a lot of work-raise a heavy weight through a great height. (EAP1E.s173)	En principio, mucha energía almacenada (como una masa pesada que se mueve deprisa) puede realizar mucho trabajo, como levantar una pesa pesada hasta una gran altura. (EAP1S.s198)
4	A lot of stored energy (a heavy mass moving rapidly) can in principle do a lot of work-raise a heavy weight through a great height . (EAP1E.s173)	En principio, mucha energía almacenada (como una masa pesada que se mueve deprisa) puede realizar mucho trabajo, como levantar una pesa pesada hasta una gran altura. (EAP1S.s198)
5	An object that possesses only a little bit of energy (a light mass moving slowly) can do only a small amount of work-raise a light	Un objeto que solo posea una pizca de energía (una masa ligera que se mueve despacio) solo puede realizar una pequeña cantidad

Figure 7.5: ACTRES query results

Similar to CWB, Manatee (Rychlý 2007) is another back-end application used for corpora studies. It is used as starting point for Sketch Engine (Kilgarriff et al. 2014) and, in combination with the front-end application Bonito, composes the open-source version NoSketch Engine (Rychlý 2007). An example of an application of parallel corpora in NoSketchEngine is the EPTIC³³, an intermodal corpus of European Parliament speeches (Ferraresi & Bernadini 2019). This corpus features multimodal components, which the user can easily access by clicking on the hyperlink provided in the concordance lines (figure 7.6). As with the previous tools, NoSketch Engine and Sketch Engine provide powerful searchers and the view of the aligned texts.

The screenshot shows the NoSketchEngine interface. At the top, there is a search bar with the query 'can' and a search button. Below the search bar, the results are displayed in a table with columns for ID, English text, and Italian text. The table shows several concordance lines, such as:

ID	English text	Italian text
#732	appropriate guarantees. We hope its modalities can be announced as soon as... possible, once	#656 garantito, speriamo appunto che le modalità possano essere annunciate... non appena il
#1114	opportunities and freedom, and I hope all can refrain from using violence and that there	#1049 libertà e la democrazia e speriamo che si possano presto tenere delle elezioni libere
#2657	proper labelling of hazardous substances can help those with l- these lung conditions	#2268 buone condizioni e buone norme di sicurezza possano permettere a questi lavoratori di la
#2788	SMEs to reach a larger marketplace which can only increase their trading potential.	#2371 procedura che costi meno, più pratica, affinché possano aumentare il loro potenziale comm
#2838	over processes and the approval of products can only help break down the walls and barriers	#2407 trasparenza e l'approvazione del prodotto possono solo essere d'aiuto per eliminare le
#3759	President ehm mis- vi- Vice-President whether we can make sure there's an annual debate on the	#3125 alla Pres- alla Vicepresidente se ehm si possa avere un dibattito annuale sul setti
#4256	market until the farms are unblocked. This can only happen after it is ensured through	#3577 origine animale da queste aziende bloccate poteva esser messo sul mercato come proc
#4788	how this contamination happened and how it can be best avoided in the future. The German	#4082 informarci ehm in merito a come potut- si è potuta avere questa contaminazione e con
#5013	very much madam President. First of all, can I say?, I sympathise totally with the position	#4341 abbiamo vissuto una situazione molto simile e posso dirvi che chi ha più sofferto, beh sc
#5088	actually not involved in any way # and I can # you have. Ehm I think the word 'criminal	#4341 abbiamo vissuto una situazione molto simile e posso confermare che nessuno dei prodot
#5656	questions to the Commission. First, whether they can now ascertain and confirm that none of	#4805 la Commissione: innanzitutto, ehm si ehm può anche ridurre ehm la quantità di le
#6501	should be very pleased. Careful management can bring down illegal and sus- unsustainably	#5536 continente. Comunque un'attenta gestione potrà trarre ispirazione dalla Tunisia per
#7930	sense, the people of Sub-Saharan Africa can take inspiration from their brothers and	#6680 quella ideale, che tutti auspicheremo. Ma ehm possono garantire che noi ehm manteniamo
#8822	dialogue with the partner countries, and I can assure you that fighting in the corruption	#7398 quella ideale, che tutti auspicheremo. Ma ehm possono garantire che noi ehm manteniamo
#8845	weaknesses the administrations have, we can n't more really to protect the forests.	#7492 forti, questi paesi, ma si rendono conto che possono cominciare a proteggere le loro for
#9710	some encouraging news, and the only way we can improve this process is working together	#8128 Quindi, se si osservano le tendenze, si può vedere che vi sono delle novità inci
#10081	Commissioner De Gucht, for good reasons, can n't be with us this evening and I want to	#8425 il commissario De Gucht che comunque non può essere qui per validi motivi stasera
#11553	negot...	Possiamo , però, procedere solo se ehm acce

 A metadata panel on the right side of the interface shows statistics for the search results, including 'st.length: medium', 'st.duration: long', 'st.speed: low', and 'st.speedwm: 110,8'. At the bottom right, there is a logo for 'Lexical Computing' and version information '2.33-open-2.129.2-open-3.79'.

Figure 7.6: EPTIC corpora on NoSketchEngine

InterCorp, a parallel corpus of Czech and other 39 aligned languages (Čermák 2019), is another project that also relies on Manatee. Instead of NoSketch Engine, the corpus is available via the web-based concordancing software Kontext (Machálek 2020). Part

³³ https://corpora.dipintra.it/public/epctic.cgi/first_form

of a more extensive project, Kontext was developed to be used for the most various reasons and by different types of users, by offering a simple web user interface (Machálek 2020). Hence, besides being a powerful tool, Kontext is also very user-friendly (figure 7.7).

Kontext also connects and offers access to the Translational Equivalent Database, Treq³⁴ (figure 7.8). Treq is a bilingual dictionary of Czech and the other 39 foreign languages found in the InterCorp. The dictionary was generated automatically from InterCorp. The language pairs were automatically aligned word by word and language pairs that occurred very often were considered a possible translation. Treq is connected to the corpus so users can click on a dictionary entry and be directed to the concordance lines for the clicked word (Škrabal & Vavřín 2017).

The screenshot shows the KonText web interface. At the top, there is a navigation menu with options: Query, Corpora, Save, Concordance, Filter, Frequency, Collocations, View, and Help. Below the menu, the search parameters are displayed: Corpus: InterCorp v11 - English | Query: drive (6,383 hits). The main content area shows search results for the word 'drive'. On the left, there is a sidebar with 'Translation equivalents (via Treq)' and 'Usage tips'. The main area displays a table of concordance lines, with columns for 'InterCorp v11 - English' and 'InterCorp v11 - Portuguese'. The table contains several rows of text, each with a checkbox and a 'brown-sifra' label. The first row shows: 'Driving the bottom of the trash can into the center of the window, she shattered the glass.' and 'Bateu com o fundo de o caixote contra o vidro, estilhaçando a janela.' The second row shows: 'As the Trailor truck drove off, Captain Fache rounded up his men.' and 'Enquanto o camião TIR se afastava, o capitão Fache reuniu os seus homens.' The third row shows: 'Then, aiming at the center of the floor tile, he drove the tip into it.' and 'Nu, com excepção de a faixa que lhe envolvia os rins e as virilhas, enrolou o hábito a a volta de a ponta de a barra de ferro. Então, fazendo pontaria bem a o meio de a laje, bateu com toda a sua força.' The fourth row shows: 'He drove the pole into it again.' and 'Bateu outra vez.' The fifth row shows: 'Even as she drove, Sophie's mind remained locked on the key in her pocket, her memories of seeing it many years ago, the gold head shaped as an equal-armed cross, the triangular shaft, the indentations, the embossed flowery seal, and the letters P. S.' and 'Mesmo a conduzir, o seu pensamento continuava preso a a chave que tinha em o bolso, a as suas recordações de a ter visto muitos anos antes, com a pega de ouro em forma de cruz de braços iguais, a haste triangular, as marcas, o brasão gravado, as letras P. S.' The sixth row shows: 'Borrowing a friend's car, Sophie drove north, winding into the deserted moon-swept hills near Creully.' and 'Pedindo o carro emprestado a uma amiga, rumou a norte, seguindo a sinuosa estrada que atravessava as colinas desertas e banhadas em luar perto de Creully.' The seventh row shows: 'Pulling the door closed, she fled the deserted house.' and 'Nauseada, fez meia volta e subiu as escadas, apoiando se a as paredes de pedra. Fechou a'.

Figure 7.7: a screenshot of InterCorp in KonText

³⁴ <http://treq.korpus.cz/index.php>

The screenshot shows the Treq search interface. At the top, there are dropdown menus for 'Source language' (English) and 'Target language' (Czech), with a double-headed arrow between them. To the right is a 'Restrict to' dropdown set to 'Collection(s): 6'. Below these is a search input field containing 'rabbit' and a yellow 'Search' button. Underneath the search bar are four checkboxes: 'Lemma', 'Multiword', 'RegEx', and 'A = a', each with a help icon. A yellow banner in the top right corner says 'Ver. 2.0'.

▲ Frequency ▼	▲ Proportion ▼	▲ English ▼	▲ Czech ▼
193	23.0	rabbit	králíka
151	18.0	rabbit	králík
85	10.1	rabbit	králíci
65	7.7	rabbit	králíčko
36	4.3	rabbit	zajice
34	4.0	rabbit	zajíc
33	3.9	rabbit	králíčku
22	2.6	rabbit	králíček
21	2.5	rabbit	Králik

Figure 7.8: a screenshot of Treq

7.2.3.2 Highlighting possible equivalents

The previously mentioned tools have the benefit of working on top of long-standing tools and allowing for robust searchers. However, they are mainly directed for users with reasonable expertise in linguistics, as they require knowledge on CL methods. Hence, their interfaces might not be ideal for non-expert users, as they are not user friendly. However, recently, more applications, that will be addressed here, have been developed addressing the needs of occasional or non-expert users. As the primary goal of many of these users is to find translated words in the appropriate context, these tools facilitate the identification of possible translations.

An example of this application is the concordancer for The Parallel Corpus German / Spanish (PaGeS)³⁵, an ongoing project (2014-2020) with approximately 25 million tokens (Doval et al. 2019). PaGeS has as one of its target audience German and Spanish learners. Thus, their creators sought to provide a fast, multi-level and user-friendly search (Doval et al. 2019:114). The query runs on lemma as default, making it simple for users who want to find an equivalent to a word and its derivations. Advanced options, supported by the underlying query tool and search platform Solr³⁶, are only displayed if required. Possible matches based on pre-stored bilingual lists are shown in bold in the concordance lines to help the users quickly go through the concordance lines (figure 7.9).

Likewise, Multingwis³⁷, a project currently featuring six corpora, also provides a handy interface (figure 7.10) for a multilingual concordancer. The search, which is powered by PostgreSQL³⁸, allows the user to search for lemmas or specific part-of-speech classes. The tool is also equipped with a multilanguage dictionary, allowing the user to browse through the frequency of possible translations. Another tool that also evidences possible matches is the one used for CLUVI³⁹, an open data collection of over 24 corpora, comprising almost 50 million words. Figure 7.11 shows a

³⁵ <http://corpuspages.eu>

³⁶ <https://lucene.apache.org/solr/>

³⁷ <https://pub.cl.uzh.ch/projects/sparcling/multilingwis2.demo/>

³⁸ <https://www.postgresql.org/>

³⁹ <http://sli.uvigo.gal/CLUVI/>

screenshot of CLUVI, in which the search term is highlighted in yellow and the possible translation in green.

Although those tools highlight possible matching words, this only occurs when there is a pre-existing dictionary with previously established translations. If the users want to identify other possible matches, they will have to go through the entire aligned sentence and guess which tokens can be a translation.



Figure 7.9: a screenshot of PaGeS

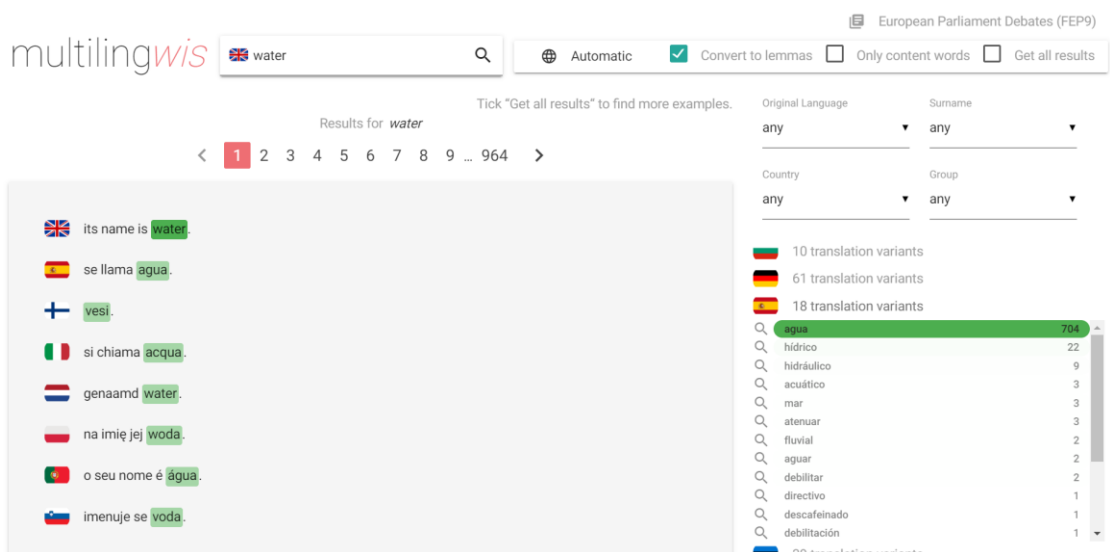


Figure 7.10: a screenshot of Multilingwis

The screenshot shows the CLUVI dictionary interface. At the top, there is a navigation bar with the SLI logo, 'Corpus CLUVI', and links for 'Inicio', 'Axuda', 'Información', and 'Recursos'. A search bar contains the text 'Pescudar nos textos'. The main content area is titled 'Dicionario' and shows the entry for 'water' in English, with Galician equivalents: 'H2O', 'auga', 'auga [2] [3]', 'irrigar', 'masa de agua', 'masa de auga', 'mexa', 'mexo', 'ourifios', 'pipí', 'regar', 'urina'. Below this, there is an 'Exemplos' section with a total of 821 examples (showing the first 20). Two examples are visible:

1- PER (141)
 EN They came to the place where the brush houses stopped and the city of stone and plaster began, the city of harsh outer walls and inner cool gardens where a little water played and the bougainvillea crusted the walls with purple and brick-red and white.
 GL Chegaron a onde terminaban as casopas de colmo e comezaba a cidade de pedra e de cemento, a cidade de grandes muros exteriores e frescos xardíns interiores nos que a auga xogaba e gurgullaba, e as buganvileas púrpuras, roxas e brancas rubian polas paredes.

2- PER (142)
 EN They heard from the secret gardens the singing of caged birds and heard the splash of cooling water on hot flagstones.
 GL Deses agochados xardíns viña o canto dos paxaros engaiolados e o larpuzar da auga fresca nas lastras de pedra quentes.

Figure 7.11: a screenshot of CLUVI

7.2.3.3 Linguee and Reverso: commercial tools

Linguee⁴⁰ (figure 7.12) and Reverso Context⁴¹ (figure 7.13) are commercial tools which work with parallel data of various languages. Instead of displaying concordance lines, these tools provide the users with possible translations and their use in context. The data used in these tools is not regarded as a corpus in the strictest sense as the query results are extracted from a collection of running texts. Nevertheless, the tools offer large multilingual dictionaries which provide information on translated texts. Despite not being CL tools, they follow a similar rationale and are more popular among learners and translators (Zanettin 2012), as they do well in meeting users' needs of easing find possible translations.

⁴⁰ <https://www.linguee.com/>

⁴¹ <https://context.reverso.net/>

Linguee

português ↔ inglês

drive

^ Dicionário inglês-português

drive *substantivo*

disco *m*

estrada *f* (plural: estradas *f*)
 I always choose to drive on the right side of the road. Eu sempre escolho dirigir do lado direito da estrada.

passaio *m*
 I took the family for a drive in the new car. Levei a família para dar um passeio no carro novo.

unidade de disco *f* (computação) (plural: unidades de disco *f*)
 My computer has two optical drives. Meu computador tem duas unidades de disco óticas.

menos frequentes:
 unidade *f* · impulso *m* · necessidade *f* · percurso *m* · propulsão *f* · movimentação *f* · viagem *f* · tracção *f* [PT] [antes AO] · acionador *m* · motivação *f* · garra *f* · ímpeto *m* · passeio de carro *m* · trajeto *m* · força motriz *f* · ânsia *f*

drive (sb./sth.) *verbo* (drove, driven)

conduzir (algo) *v*
 My sister prefers to drive an automatic car. A minha irmã prefere conduzir um carro automático.

dirigir *v*
 I drive cars, but I am not capable of driving buses. Eu dirijo carros, mas não sou capaz de dirigir ônibus.

incitar *v*
 The prospect of a promotion drives him to work hard. A perspectiva de uma promoção o incita a trabalhar duro.

menos frequentes:
 impulsionar algo *v* · guiar *v* · mover *v* · prosseguir *v* (computação) · orientar algo *v* · direcionar algo/alguém *v* · seduzir *v* · coagir *v* (computação) · rebater *v*

Exemplos:
 flash drive *s* — pen drive *m*
 test drive *s* — test drive *m*
 drive gear *s* — engrenagem de acionamento *f*

Figure 7.12: screenshot of Linguee

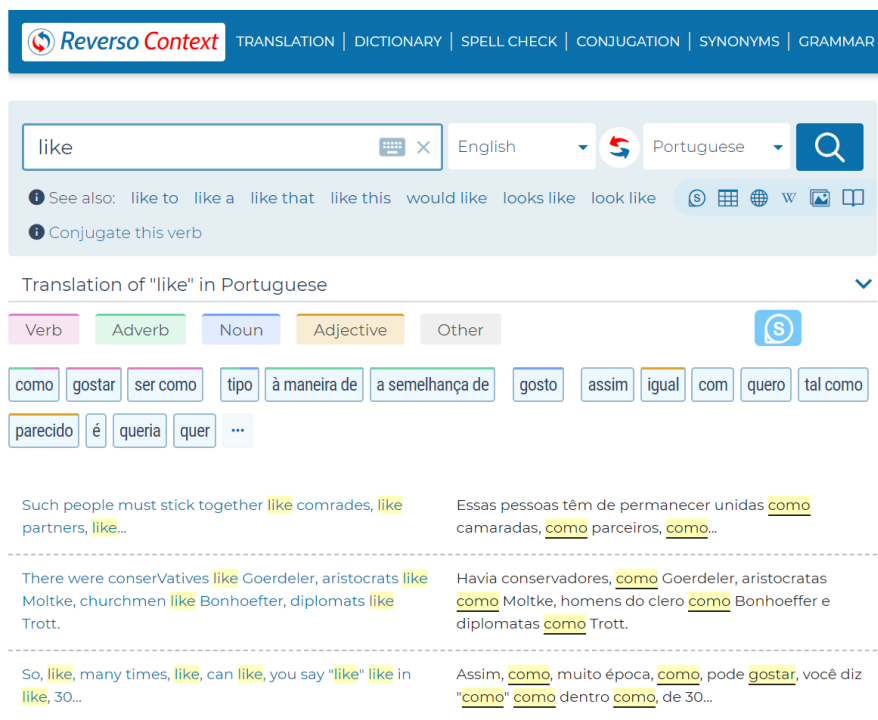


Figure 7.13: screenshot of Reverso Context

7.3 Limitations and Motivation

The previous section has briefly illustrated the growing number of parallel corpora and instruments for translation and other multilingual studies, possibly led by the now much easier retrieval and processing of data than before, as described in chapter two. Still, corpus quality and systems to analyse this rising number of corpora have not advanced with the same speed that data has become available (Eckart & Quasthoff 2013:152). Not much has been done in terms of new methods or techniques to explore parallel corpora, apart from the creation or adoption of bilingual lists, which is a mostly manual and not innovative method.

This section explores the limitations and possibilities of using parallel corpora mainly by novice translators and for second language education. In translation studies, corpus methodologies have been used to either teach novice translators or as a translation resource (e.g. Gallego-Hernández 2016; Rodríguez-Inés & Gallego-Hernández 2016).

In second language education, parallel corpora are not yet fully explored (Doval & Sánchez 2019).

7.3.1 Indirect use of parallel corpora

The previous section has shown that parallel corpora (or texts) can be indirectly used to inform learning and translation applications. Users can rely on dictionaries and term bases that were compiled based on parallel corpora without having to investigate the corpora themselves. The indirect use has the benefit of aiding user experience. However, there are a few shortcomings with that, as discussed below.

7.3.1.1 Commercial tools: inaccurate, inconsistent and obscure systems

Searching engines created for language investigations, like Linguee, have the advantage of offering fast, and in many cases, free solutions. However, there are some drawbacks with this approach. In many cases, the data used to feed these tools are not clearly described (e.g. Linguee, Reverso), and the source is not provided (Doval et al. 2019:106). This means that the suggested translations might not reflect what the user is looking for. Moreover, this data configuration prevents linguists from making claims about a language population and prevents the reproduction of research. Also affecting reproducibility is the constant change in ranking algorithm and index observed in search engines. This inconsistency in data means that same queries performed at different times are likely to yield different results (Shi & Fung 2013:33), differently from what happens with corpus linguistic tools.

7.3.1.2 Context

Meanings are context sensitive. Context affects translation choices be them at a pragmatic, syntactic, semantic and even phonetic level (e.g. Baer 1995; Pym 2007; Ji 2012). However, not all commercial tools offer the user with real occurrences of the

searched terms. Some Translation Memories can be prepared to offer context, but they are usually just some short and isolated sentences, with no or almost no information on the text source. This paucity of information may preclude the user from fully grasping meanings in context. Furthermore, the use of TMs also requires an expertise that not all users might have.

Another issue is the cross-linguistic sense of meaning displacement when comparing languages (Lewandowska-Tomaszczyk 1987:4). For instance, the verb “take” in English has a superordinate category for many actions, without an equivalent verb in Portuguese (figure 7.14). This shows that the more examples in context we have, the more chances we have of seeing the multiple meanings a word can have and its equivalent translations.

to take a shower		tomar um banho
to take the bus		pegar o ônibus
to take a break	→	dar um tempo
to take a risk		correr um risco

Figure 7.14: possible Portuguese translations for phrases with the verb to take

In any case, decision making in translation is strongly connected to the context and choices for the best translation is better made when usage is considered. Hence, relying on resources (e.g. dictionaries, MT) that preserve the source and the broad context of the text from where the term in question occurred reinforces translation quality (Koehn 2009).

7.3.1.3 External resources

Most of the existing tools that indicate possible translation equivalents for a search word require external resources such as dictionaries (Gamallo 2019:256). However, in

many cases, these dictionaries might not be available, especially for minority languages and specific dialects (Hewavitharana & Vogel 2013:192). Moreover, languages are always evolving. Identifying links for translations is an open process (Ji 2012:55) and dictionaries are not created as fast as corpora. Relying on a resource built on outdated data, as it is the case with many translation memories and dictionaries, may not reflect the current linguistic features.

7.3.2 CL tools and methods

Corpus tools can provide for the two previous points, but they also have their limitations. The most likely first mention of the use of CL in translation studies, suggesting that translation studies should move from a prescriptive to a descriptive approach (Baker 1993), is now long dated. Many studies (e.g. Malmkjaer 2003; Rabadán 2005; Tymocko 1998) have followed Baker (1993)'s idea of shifting to exploring word meaning rather than word usage. However, not much advance has been made in the sense of developing new tools for users who have not yet developed full expertise on the area.

7.3.2.1 Queries

One of the many advantages of using CL tools for translation purposes is the powerful query processor that many CL programmes offer. By using regular expressions and sophisticated query languages like Corpus Query Processors (CQP) from CWB and Corpus Query Language (CQL) from SketchEngine (see 3.4.3), users can perform elaborated searches. That means they can describe the patterns they are looking for and look for a combination of linguistic forms, rather than single words or lemmas. For example, the following CQP query

```
[lemma = "take" & pos = "V.*"][(word="shower"%c | word="bus"%c)]{1,3}
```


looks for different forms of the verb *take* that are not followed (in a span from one to three to the right) by the upper or lower case words *shower* nor *bus*. Complex queries can be beneficial when a grammatical category exists in only one of the parallel corpora, for instance. To my knowledge, there are not commercial tools that offer robust query systems like those.

7.3.2.2 Register balance and pattern identification

Extra-linguistic features such as the author's gender, text's genre, have a great impact on the translation (e.g. Hareide & Hofland 2012). For this reason, because corpora (should) have information on the texts available, they provide translators with the reassurance they need for their strategic decision making (Varantola 2003). General-purpose corpus offers users the possibility of exploring a high number of phraseology patterns across different and specific registers. Translators that rely on these corpora are said to be more aware of phraseology and register issues (Aston 1999).

One issue with that, however; is that perfect representativeness is even harder to achieve with parallel corpora than with a single corpus. Some types of texts are more translated than others, leading to a strong bias for specific genres (Hareide & Hofland 2012:78; Mauranen 2004:74; Biber 1993; Johansson 1998:6; Zanettin 2000:108-109). Hence, the translations in the parallel corpus might not be what the user wants to have as a reference because of the register influence. The ideal reference translation should roughly match the same register of the translation in process. To work around this issue when dealing with parallel corpora is to pay close attention to text metadata and concordance lines or to restrict a corpus search for only texts that meet the translation scenario.

However, for as much as CL offer full-text browsing with rich contextual information, we are still missing what is probably the main reason to rely on quantitative linguistics: identifying a statistically significant correlation between the variables under investigation. In the context of possible translation in parallel corpora, this means having a metric that helps us identifying which pair of words are strongly connected and that high frequency of cooccurrence is not happening by chance.

7.3.2.3 Difficulties in applying sophisticated methods

Translators are not very familiar with CL. A survey (Kunz & Steiner 2010) has shown that translators rely on reference texts, with translations previously done and validated, to inform their work. However, in most cases, they use tools like Microsoft Word to query those texts, instead of CL tools. This increases the workload, as retrieving specific information in specific texts by using those tools is not easy. Furthermore, in the few cases that translators do rely on CL, finding the appropriate translation pair is not always easy. In many cases, there are too many solutions for a query and browsing through most of them proves to be a colossal task.

7.3.3 A new solution to the users

The previous sections have shown that applications of parallel corpus vary from simple methods such as working with frequency and examples to more complex ones such as relying on multivariate techniques. Non-linguist and translators rely on the simplest methods, as well as linguist researchers (see chapters 4 and 5). Although these simple methods can be effective, they can also lead to poor or misleading results. Hence, they could benefit from a more sophisticated analysis that is not difficult to perform like multivariate techniques, but that also shows the difference in linguistic behaviour as extra-linguistic variables change.

This work will aim at solving the problem that users need to see possible translations in a rich context and to be able to quickly and easily tell whether a translation pair is frequent and significant. Next section presents the process of conceptualizing a tool that offers accurate measures of frequent language pairs but easy to understand.

7.4 The tool

As discussed in chapter five, the focus of this study is to develop a tool that better suits non-specialist users of corpus data and methods (NSUs). This tool has the particularity of also targeting beginners in translation and language learners. This section will present the conceptualization and development of a tool that sees for the limitations and needs described in the previous sections faced by potential target users.

7.4.1 Conceptualization of the tool

Hence, the goal here is to provide users with an intuitive tool that allows for the smooth and accurate identification of possible translations or related words; having in mind that different types of registers impact on the result. Statistical techniques should be implemented to guarantee the accuracy of the tool, but as the target audience might not be familiar with these techniques, they should be decoded to users via an easy to grasp format.

To achieve that, I have the following two assumptions. First, the tool should be able to provide some indicator that realizes the connection between the performed query and a token in the aligned corpus, as the highlighting in the tools in 7.2.3.2. Second, two specific types of users within the NSUs are clearly defined. For the tool described in this chapter, two specific user groups are also delimited: learners of translation and language learners. Both groups users can benefit from a tool that shows the translation

equivalent words in the parallel concordances, without having to rely on external resources. I used the scheme below when developing the new tool.

7.4.1.1 Basic concepts

A good starting point to create a user-friendly tool is that it should start from a basic concept, that any novice user could relate to and understand the reasoning. Considering that frequency is such a concept, I determined that the tool should provide for easy retrieval and visualization of (i) the words that frequently occur in the equivalent aligned zones of a performed query; (ii) the frequency per text of these frequently occurring words; (iii) the degree (strength) to which the frequent words found in the aligned corpus are linked (or not) to the query. Because (i) and (ii) are properties that users are most familiar with, they can be simply given as numbers. As (iii) can be harder to grasp, conveying it through visual aids can facilitate understanding.

The second point is that users should be able to use the tool to investigate any language or language variety, including low-resource ones (minority languages). Hence, no external resource such as bilingual dictionaries and schemes for dependency parsers should be needed. Finally, users need a quick and easy way of comparing how strong two words in a language pair are connect. For that they will need an easy to understand and compare value.

To develop the tool, we must assume that there are two sets of data (the source corpus and the target corpus) and that the corpora have aligned zones (see figure 7.15). The tool should (i) function without the requirement of any training data or resource other than the corpus itself; (ii) not be significantly affected by inaccuracies in alignment; and (iii) handle different types of parallel corpora and zone lengths.

```
<s> Just as if you have drunk too much you are not allowed to
<query> drive </query> , so if you smoke too much you will not be
allowed to drive . </s>

<s> Così come quando uno ha bevuto troppo non ha il diritto di
guidare , allo stesso modo quando uno avrà fumato troppo , non
avrà il diritto di guidare . </s>
```

Figure 7.15: example of comparable zones

7.4.1.2 Concordance lines and metadata

We must also assume that words have practical synonyms (similar meanings) but that they carry different layers of meaning. The many possibilities of translations for a single word demonstrate this plurality of word meanings. Hence, the tool should account for the inexistence of perfect synonyms.

Language scholars already use parallel concordances to explore AZs. As section 7.2.3.3 has shown, it would also be useful to have the equivalent term to the corpus query highlighted in the AZ of the TC. It would be even more useful to be able to do this highlighting on the basis of corpus-internal data only so that no setup of an external resource is needed.

Metadata is also extremely important, as linguistic features can significantly vary across registers (e.g. Neumann 2011; Neumann & Hansen-Schirra 2013:327). Users should have enough information from the texts they investigate. They should also be able to work with restricted queries and see the contrast results between searches in specific texts and the entire corpus. CQPweb already does parallel concordance display, provide easy access to text metadata and allow for restricted queries. The new tool described in this chapter identifies the possible translations and provides the highlighting system.

7.4.2 Parallel Link

With the assumptions and scheme aforementioned in mind, I developed a concept called parallel link, or *parlink*. Parlinks are tokens of a parallel corpus B that are strongly associated with a query performed in a corpus A. This means that the more frequent a token occurs within the corpus B zones aligned to a corpus A zone of the query results in comparison with the remainder of corpus B, the more likely it is for this token to be considered a Parlink. The reasoning behind the calculation is the same as the one used to calculate collocates. The difference here is that, instead of using the span to the left and right of the collocation node, for Parlink the span of comparison is the aligned zones.

For instance, when searching for the word “car” in a corpus in English, the stronger Parlink found in the Portuguese aligned corpus was *carro*, the exact translation for car. That means that the token *carro* in the Portuguese corpus occurs (almost) always aligned to zones where the token *car* is found in the English corpus. Other parlinks found in this query were *dirigir* (to drive), *estacionar* (to park). Hence, parlink gives not only possible translations but also related words.

To determine whether a token is a parlink or not, and to which degree, I used a parlink score. The higher the score, which ranges from 0 to 1, the stronger the Parlink is. The use of a score ensures consistency across results and aid on the identification of subtle and complex linguistic patterns in parallel corpora. Having a parlink score also helps systematize the comparison of parlinks in other parallel corpora.

7.4.3 Creation Process

Similarly to what I did Advanced Dispersion tool, I first created prototypes and then shared them with possible users, before moving on to the development itself. The

difference with the Parlink Tool development process is that all prototypes were shown at the same time for the users and the reactions to all four models were taken into consideration. This was done to reduce the time spent on the pre-development phase. As for any software development, this should be an iterative process, as defined and explained below. The process and user assessment are based on Agile methods for academic research (Hicks & Foster 2010). The cyclical steps are to (i) understand users' needs; (ii) build prototypes to match this goal; (iii) deliver the prototypes; (iv) repeat if necessary (see figure 7.16).

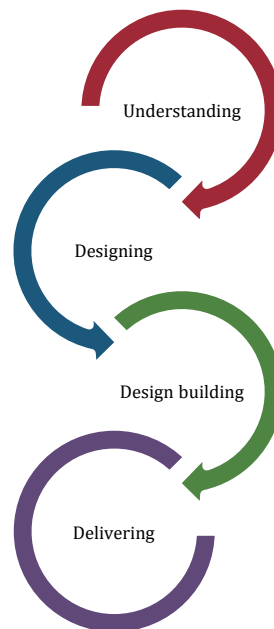


Figure 7.16: Agile Process

7.4.3.1 Prototypes

Five prototypes (figures 7.17-7.21) were developed at the same time and presented to some potential users of the tools. The main aim with all the prototypes was always the same: allow the user to quickly see which tokens in the parallel corpora are closely associated with the query performed.

The *Bar & Dots* prototype (figure 7.17) indicates with a green star all the parlinks that have a score higher than the average. For the other tokens, vertical bars are used to represent their relative frequencies in the entire corpus (yellow bar) and when occurring aligned with the query (blue bar). The legend on the bottom right is presented as a pop-up so the users can orient themselves when analysing the results. As for the dispersion tool in the previous chapter, the Parlink View was also implemented in CQPweb (see 3.4.3 and 6.3.3), hence all the prototypes were created considering CQPweb interface.

The *Arrows & Bow* (figure 7.18) style also relies on the parlink score mean to highlight the tokens. A yellow diamond indicates that the token has an average parlink score. Tokens with a score above the mean are indicated with a blue triangle (high) or a green circle (very high). Tokens with a score lower than the average, and for this reason improbable to have a link to the query, are highlighted with an orange inverted triangle and a red cross.

The third prototype conveys the parlink score by giving each token a background with different luminosity. Words with higher scores have a darker background, while words that are unlikely to be a link have a light background. The *Heatmap + sidebar* (figure 7.19) prototype also has a sidebar to be used as a reference. In the image the highlighted concordance line is indicated with a red rectangle. This bar shows the two tokens with higher parlink score for each concordance line.

Your query "Sky" returned 434 matches in 130 different texts (in 39,431,862 words [658 texts]; frequency: 11.01 instances per million words) [0.132 seconds]

Showing parallel corpus Europarl 3: Italian ▾

Solution 1 to 50 Page 1 / 9

No	Text
1	<p>freedom of movement , are turning out to be pie in the sky . Refugees are being allowed to return to their homes only after</p> <p>Vari obiettivi degli Accordi di Dayton , come la libertà di circolazione , si rivelano un nido .</p> <p>upon our votes , even if it is not shining in the sky , above Strasbourg , and that IN the voice of Parliament will be heard</p>
2	<p>Mi auguro che oggi il sole del solizio d'estate illumini la nostra votazione , anche se non splende nel cielo di Strasburgo , e che la voce del Parlamento venga ascoltata dal prossimo Consiglio dei Ministri europei</p>
3	<p>contaminated meal , we shall soon be seeing aircraft falling from the sky . We abstained in the vote to reject the Council 's common</p> <p>Se oggi vediamo contaminare le farine , domani vedremo cadere gli aspi .</p>
4	<p>along the night lines . But there is one cloud in the sky , too : it worries me when certain parties want to take</p> <p>(no alignment found)</p>
5	<p>sold to the highest bidder . Within Europe this has resulted in Sky TV having a virtual domination in this area because national television stations</p> <p>In Europa è risultata tale la Sky TV che controlla potenzialmente quest'area , giacché le emittenti televisive nazionali , come la RTE e la BBC non sono state in grado di competere a causa delle loro risorse limitate</p>
6	<p>against this . Obviously they do not like the 'spy in the sky ' . I come from the freight industry where we have had</p> <p>Non piace molto l'idea di una « spia nel cielo » .</p>
7	<p>cancer of terrorism ! A blue bond , Mr President . The sky , over the European Union today is an immense blue bond stretching from</p> <p>Oggi il cielo dell'Unione europea è un immenso nastro blu che la ricopre da La Palma a Malmo , da Rodi a Dublino .</p> <p>someone manufactured this meal ! It did not fall from the sky , as the representative of the British Government would have us believe</p>
8	<p>Non sono cadute dal cielo , come ci diceva il rappresentante del governo britannico</p>
9	<p>to resolve this kind of practical problem than to reach for the sky , by abolishing frontiers . Madam President , I would like to give</p> <p>Saràbbe più utile risolvere questo genere di problema concreto invece di fare calcoli sulla cometa dell'abolizione delle frontiere</p> <p>the point at which the planes start to fall out of the sky ? Are we to have a cost- benefit study setting the added</p> <p>Vorrei sapere fino a che punto vadano ridotte - fino al punto da far precipitare anche gli aspi ?</p>

● index > μ (or Q4)
■ total in corpus
■ observed equivalent

Strasbourg's cathedral can one more reach in into a triv/blue **sky** Obviously Mr President that aim will not be achieved

Figure 7.17: Bars & Dots Prototype

Your query "Sky" returned 434 matches in 130 different texts (in 39,431,862 words [658 texts]; frequency: 11.01 instances per million words) [0.132 seconds]

No	Text
1	<p>freedom of movement , are turning out to be pie in the sky . Refugees are being allowed to return to their homes only after</p> <p>Vari obiettivi degli Accordi di Dubron , come la libertà di circolazione , si rivelano un mito .</p>
2	<p>upon our votes , even if it is not shining in the sky above Strasbourg , and that/IN the voice of Parliament will be heard</p> <p>Mi auguro che oggi il sole del sole del solstizio d'estate illumini la nostra votazione , anche se non splende nel cielo di Strasburgo , e che la voce del Parlamento venga ascoltata dai prossimi Consigli dei Ministri europei .</p>
3	<p>contaminated meal , we shall soon be seeing aircraft falling from the sky . We abstained in the vote to reject the Council 's common</p> <p>Se oggi vediamo contaminare le farnie , domani vedremo cadere gli aerei .</p>
4	<p>along the right lines . But there is one cloud in the sky , too : it worries me when certain parties want to take</p> <p>(no alignment found)</p>
5	<p>sold to the highest bidder . Within Europe this has resulted in Sky . TV having a virtual domination in this area because national television stations</p> <p>In Europa è risultata tale la Sky TV , che controlla potenzialmente quest'area , giacché le emittenti televisive nazionali , come la RAI e la BBC non sono state in grado di competere a causa</p>
6	<p>against this . Obviously they do not like the 'spy in the sky ' . I come from the freight industry where we have had</p> <p>Non piace molto l'idea di una spia del cielo .</p>
7	<p>cancer of terrorism ! A blue bond , Mr President . The sky over the European Union today is an immense blue bond stretching from</p> <p>Oggi il cielo dell'Unione europea è un immenso nastro blu che ricopre la Patina a Marino , da Rodi a Dublino .</p>
8	<p>someone manufactured this meal ! It did not fall from the sky , as the representative of the British Government would have us believe</p> <p>Non sono cadute dal cielo , come ci diceva il rappresentante del governo britannico .</p>
9	<p>to resolve this kind of practical problem than to reach for the sky , by abolishing frontiers . Madam President , I would like to give</p> <p>Sarebbe più utile risolvere questo genere di problema concreto invece di fare calcoli sulla competenza dell'abolizione delle frontiere .</p>
<p>the point at which the planes start to fall out of the sky . ? Are we to have a cost-benefit study setting the added</p> <p>Vorrei sapere fino a che punto vadano ridotte - fino al punto da far precipitare anche gli aerei . ?</p>	
<p>Strasbourg 's cathedral can one more reach up into a trabe blue sky . Obviously Mr President that aim will not be achieved</p>	

ARROWS & BOW

very high ●

high ▲

neutral ◆

low ▼

very low ✖

Figure 7.18: Arrows and Bow prototype



Figure 7.19: Heatmap + Sidebar

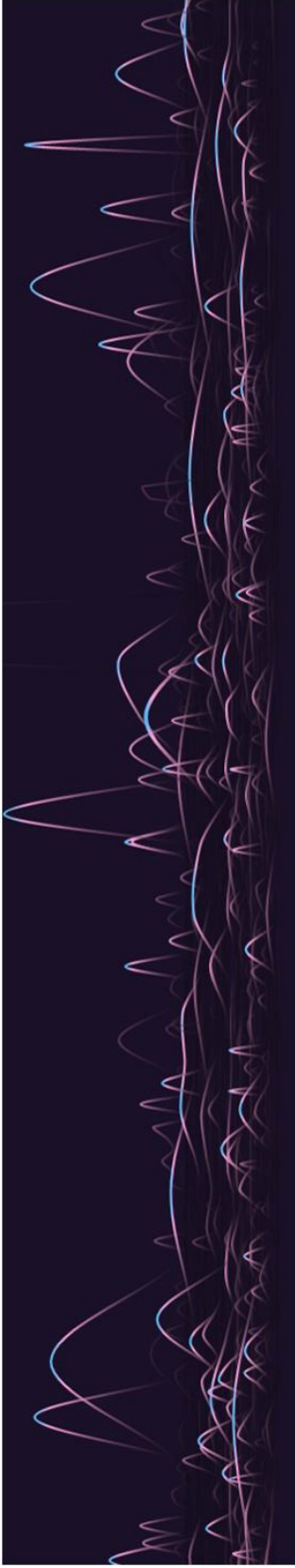
In the *Mountain View* prototype (figure 7.20), peaks indicate high scores and almost flat lines suggests no weak connection between the token and the query. An overview for each page of the concordance line is displayed in a rectangle above the solutions. When hovering over a specific concordance line, the user can see the visualization for that specific line. The top image is a summary of all the concordance lines in the solution page.

The last prototype, *Sum up Flower* (figure 7.21), presents a summary of the parlinks, instead of indicating the parlink for each token in the concordance lines, display only the strongest parlinks as a flower visualization. Each petal in the flower conveys the following information: total occurrence of the parlink (green); expected (blue) and real (yellow) frequency of parlink within the aligned sentences; number of texts in which the token occurs (grey); and parlink score (orange), here given as dice coefficient.

Your query "songs" returned 19 matches in 17 different texts (in 39,431,862 words [658 texts]; frequency: 0.48 instances per million words) [0.021 seconds - retrieved from cache]

Showing parallel corpus Europarl 3: Italian ▾ Switch
 Choose action... ▾ Go!

Show in random order Line View
 Show Page: 1 > | >> << <<< >>>



Solution 1 to 19 Page 1 / 1

No	Text
1	<p>ep_96_06_20</p> <p>music to the ears of everyone here - one of your best <u>songs</u> . Mr President , Commissioner , ladies and gentlemen , let me</p> <p>Le sue parole , che sottolineano l'importanza delle lingue come elemento di cultura , rappresentano indubbiamente un aspetto molto importante e le posso assicurare che il suo intervento è suonato a noi tutti come una delle sue migliori melodie .</p>
2	<p>ep_96_10_21</p> <p>fact that/IN most people can manage to understand instruction sheets and hit <u>songs</u> does not make the danger any less real . Mrs Mouskouri 's</p> <p>Il fatto che i più siano capaci di leggere i foglietti delle istruzioni per l'uso o i testi di canzoni alla moda non significa molto .</p>
3	<p>ep_98_01_13</p> <p>is that/IN we must safeguard our rich diversity . Music and popular <u>songs</u> are , of course , part of that diversity and must be</p> <p>Bisogna salvaguardare tutto questo .</p>
Mountain View	
4	<p>ep_98_05_13</p> <p>which Parliament accepted . At that time , when Parliament broke into <u>songs</u> of triumph , I said that/IN it was somewhat premature because I</p> <p>(no alignment found)</p>

Figure 7.20: Mountain View Prototype

Your query "songs" returned 19 matches in 17 different texts (in 39,431,862 words [658 texts]; frequency: 0.48 instances per million words) [0.021 seconds - retrieved from cache]

canzoni

Metric	Value
Total	9
Observed	3
Dice coeff.	0.0948

canti

Metric	Value
Total	7
Observed	5
Dice coeff.	0.0769

musicale

Metric	Value
Total	65
Observed	1
Dice coeff.	0.706

No	Text
1	ep_96_06_20 music to the ears of everyone here - one of your best <u>songs</u> . Mr President , Commissioner , ladies and gentlemen , let me Le sue parole , che sottolineano l' importanza della lingua come elemento di cultura , rappresentano indubbiamente un aspetto molto importante e le posso assicurare che il suo intervento è suonato a not tutti come una delle sue migliori melodie .
2	ep_96_10_21 fact that/IN most people can manage to understand instruction sheets and hit <u>songs</u> does not make the danger any less real . Mrs Mouskouri 's Il fatto che i più siano capaci di leggere i foglietti delle istruzioni per l' uso o i testi di canzoni alla moda non significa molto .
3	ep_98_01_13 is that/IN we must safeguard our rich diversity . Music and popular <u>songs</u> are , of course , part of that diversity and must be is that/IN we must safeguard our rich diversity . Bisogna salvaguardare tutto questo .
Sum up Flower	
4	ep_98_05_13 which Parliament accepted . At that time , when Parliament broke into <u>songs</u> of triumph , I said that/IN it was somewhat premature because I

(no alignment found)

Figure 7.21: Sum up Flower prototype

7.4.3.2 User's response

I asked seven potential users of the parlink tool to look at the prototypes and give their feedback and comments. I also used techniques described in 5.3 to study their reactions. All the participants preferred the *Heatmap & Side Bar* to the other prototypes. However, most of the participants found it hard to understand what was conveyed with the sidebar. The sum up flower prototype also attracted the attention of some of the participants. The design of the prototype was not much appreciated, but users pointed at the convenience of having a summary of parlinks handy.

Based on this feedback, I chose to implement an enhanced version of the Heatmap prototype. However, instead of having a sidebar, the visualization has a pop-up table with the parlinks sorted according to their score. The concordance line view makes it easy for users to spot the parlinks with high scores readily, and the table allows them to see detailed information. With this visualization, users can perform a blend of qualitative and quantitative analysis.

7.4.4 The tool development

7.4.4.1 The testing data

To develop and test the tool, I used a corpus composed of two novels written in Brazilian Portuguese and a parallel corpus with the English translation of the two novels. The two novels, *Dom Casmurro* (1899) and *Memórias Póstumas de Brás Cubas* (1881), were written by the Brazilian author Joaquim Maria Machado de Assis. There were two main reasons for this choice. The first is that the books are in the public domain; hence they are easy to be retrieved. The second is that because I am very familiar with the two books, it is easier to understand any unexpected results with the data.

To align the corpora, I used *cwb-align* (Evert & Hardie 2011). This alignment tool basically relies on looking for cognates, i.e. similar character n-grams; therefore, it does not offer optimal results. However, its implementation is quick and easy and meets the purpose of testing the parlink tool. It also allows me to test whether the parlink tool will work well even when the alignment is not extremely accurate.

7.4.4.2 Parlink score: using dice coefficient

To calculate the parlink score, I took into account three quantitative properties I could retrieve from the parallel corpora in order to verify the strength of the parlink in relation to the query. These properties were (i) the frequency the potential parlink in segments aligned to the query occurrences, (ii) the significance, or not, of a high frequency, i.e., the high frequency is not by chance; and (iii) the effect size, or here, how strongly connected the parlink and the query are.

Several measures can be used with the properties described above, and they all have their strengths and weakness. The data and the application can directly affect the performance of the measure. Hence, empirically defining which measure would have the best performance should be the best way. For this first version of the Parlink tool, dice coefficient is the adopted measure. Shall the combination of other languages or corpora prove that this measure leads to bad performance, we should then try different approaches.

Dice coefficient is an association coefficient that favours strong combination cases (Evert 2008). For this reason, it is commonly used to identify collocations with strong patterns, such as multiword units (McKeown et al. 1996; Dias et al. 1999). Assuming that translations and their sources are also relatively rigid combinations, dice coefficient is a plausible choice here. Moreover, being an effect size measure, Dice

Coefficient is comparable across corpora and does not overweight low-frequency items (Evert 2009).

Its calculation, applied to parlink, takes the harmonic mean of (a) the frequency of a type occurring in a zone aligned to the result as a proportion of the type's overall frequency, (b) the frequency of the type occurring in a zone aligned to the result as a proportion of the overall frequency of results for the query in question. The closer the score is to 1, the stronger the parlink is. The formula, where P_{az} is the frequency of parlinks in the AZ, T_{az} is the total number of tokens in the aligned zone and P_{ec} is the frequency of the Parlink in the entire corpus, is given below.

$$Dice\ Coefficient = \frac{2P_{az}}{T_{az} + P_{ec}}$$

7.4.4.3 Technicalities: frequency per text, lemma, restricted search

Some specific technicalities were considered when developing the Parlink tool. First, when designing the structure for the Parlink database, I established that the frequency of the query for each text would be preserved. This information is not necessary when calculating dice coefficient and, depending on the corpus size, keeping this data can be expensive for the system. However, a text cannot be considered a random sample, but a semiotic system in its own. Hence, considering the parlink frequencies for each text can help the user understand the words' ambiguity and their translations.

For this first version, parlink will be only calculated on the bases of tokens. This is because all corpora have word types, whereas not all corpora have other layers of annotation as lemma. However, parlink could be expanded for lemmas and other types of annotation, as some studies suggest their importance. Ji & Oakes (2012:185), for instance, study the difference in frequencies of emotion and value words in two

aligned corpora. For this, the authors used the UCREL Semantic Analysis System (USAS). This study suggests that further implementation of parlink with annotation might also be useful. Lemma for parlink might also be implemented, as it will make a massive difference for agglutinative languages such as Turkish, Finnish and Hungarian.

Because the extra-linguistic features can have a high impact on the translation, allowing the user to restrict the search is crucial. Parlink accounts for that. If a user restricts the search, the parlink score will represent only the queries subcorpus.

7.4.4.4 Interface

The final interface of the tool was designed with the aim of simplifying the metrics through a visualization that relies on the use of colours. However, if the user wants, they can also easily retrieve the statistical information. The data is not obscured by the visualization but clarified.

Two parlink options, highlight and table, will be visible when the aligned sentences are in display on the concordance lines. The highlight button adds a colour mark to each token in the aligned line. The parlink score is used to alter the luminosity in the colour of those marks. The stronger the parlink is, the darker the colour will be (figure 7.22).

Your query "dog" returned 6 matches in 2 different texts (in 174,369 words [2 texts]; frequency: 34.41 instances per million words) [0.003 seconds - retrieved from cache]

Showing parallel corpus Machado Portugues AI | Switch | Choose action... | Go!

Highlight | Table

Show in random order

Line View

Show Page: 1

<< >> > |

No	Text
1	<p>bracubasen</p> <p>... " Was n't anything ? You treated me like a dog ... " With that word I took her hands , kissed them</p> <p>Tratou -me como não se trata um cachorro ... A esta palavra , peguei -lhe nas mãos , beijei -as , e duas lágrimas rebentaram -lhe dos olhos .</p>
2	<p>bracubasen</p> <p>of silence passed . We could only hear the harking of a dog and . I 'm not sure , the sound of the water</p> <p>Decorreram alguns instantes de silêncio ; ouvíamos somente o latir de um cão , e não sei se o rumor da água , que morria na praia .</p>
3	<p>bracubasen</p> <p>you see ? The man who fights over a bone with a dog , has the great advantage over him of knowing that he 's</p> <p>Logo , o homem que disputa o osso a um cão tem sobre este a grande vantagem de saber que tem fome ; e é isto que torna grandiosa a luta , como eu dizia .</p>
4	<p>domcasmurroen</p> <p>the matter as a joke , laughing and calling me a sly dog . Afterwards , she said she believed I would keep my word</p> <p>Capitu meteu o negócio à bulha , rindo e chamando -me disfarçado .</p>
5	<p>domcasmurroen</p> <p>'s no trouble to for give them , as I forgave a dog which robbed me of my rest in worse circumstances . I 'll</p> <p>Em verdade , um rói -me os livros , outro o queijo ; mas não é muito que eu lhes perdoe , se já perdoei a um cachorro que me levou o descanso em piores circunstâncias .</p>
6	<p>domcasmurroen</p> <p>think that it was the smell of the meat that made the dog quieten down . I do n't say it was n't so ;</p> <p>Ao leitor pode parecer que foi o cheiro da carne que remeteu o cão ao silêncio .</p>

Help! for this screen

You are logged in as user [chico]

CCPweb v3.2.40 © 2008-2019

Figure 7.22: screenshot of the parlink tool

When clicking on the table button, a pop-up table of sorted parlinks appears (Figure 7.23). The table allows for a general view of parlinks as well as helping spot the difference between words with very similar parlinks, as you have all parlinks listed together. The pop-up page also has a printing button, so the results can either be directly printed or downloaded.

7.5 Conclusion

Developing is a continuous process. There is always room for improvement and work to be done. Parlink was developed with flexibility in mind, and thus to make it possible to be used for several other purposes, including small personal projects or creation of translational databases. With informal testing, potential users have shown positive responses and will to use the tool for research, language teaching and translation practice.

The next step for this work is to test it with corpora with different characteristics, such as size; degree of comparability; different zone attributes. By doing that, new suggestions for the tool are very likely to emerge. For instance, dice coefficient has shown excellent results with the tested dataset. However, shall it become necessary; some other metrics could also be implemented.

Your query "dog" returned 6 matches in 2 different texts (in 174,369 words [2 texts]; frequency: 34.41 instances per million words) [0.003 seconds - retrieved from cache]

Showing parallel corpus Machado Portuguese AI | Switch | Choose action... | Go!

Highlight | Table

ion 1 to 6 Page 1 / 1

like a **dog** ... " With that word I took her hands , kissed them
 vra , peguei -lhe nas mãos , beijei -as , e duas lágrimas rebentaram -lhe dos olhos .
 g of a **dog** and , I 'm not sure , the sound of the water
 somente o latir de um **cão** , e não sei se o rumor da água , que morria na praia .
 with a **dog** , has the great advantage over him of knowing that he 's
 ande vantagem de saber que tem fome ; e é isto que torna grandiosa a luta , como eu dizia .
 : a sly **dog** . Afterwards , she said she believed I would keep my word
 cio à bulha , rindo e chamando -me disfarçado .
 gave a **dog** which robbed me of my rest in worse circumstances . I 'll
 e eu lhes perdoe , se já perdoei a um cachorro que me levou o descanso em piores circunstâncias .
 de the **dog** quieten down . I do n't say it was n't so ;
 : foi o cheiro da carne que remeteu o **cão** ao silêncio .

this screen You are logged in as user [chico]

No.	Word	Total no. in corpus	Expected parallel link frequency	Observed parallel link frequency	In no. of texts	Dice coefficient
1	cão	3	0.0031	3	2	0.6667
2	cachorro	2	0.0021	2	2	0.5000
3	ouviamos	1	0.0010	1	1	0.2857
4	remeteu	1	0.0010	1	1	0.2857
5	perdoei	1	0.0010	1	1	0.2857
6	rebentaram	1	0.0010	1	1	0.2857
7	disfarçado	1	0.0010	1	1	0.2857
8	descanso	2	0.0021	1	1	0.2500
9	piores	2	0.0021	1	1	0.2500
10	Decorreram	2	0.0021	1	1	0.2500
11	grandiosa	2	0.0021	1	1	0.2500
12	latir	2	0.0021	1	1	0.2500
13	cheiro	3	0.0031	1	1	0.2222
14	chamando	3	0.0031	1	1	0.2222

Figure 7.23: screenshot of the pop-up table for parlink

8 Conclusion

This chapter discusses how the work presented in the thesis has answered the research questions in chapter one. I start by presenting a summary of the thesis (8.1) and how it answered the research questions (8.2) and contributed to the field (8.3). The final sections discuss the limitations of this work (8.4) and set some future work to be done (8.5).

8.1 Thesis Summary

Chapter two discussed the evolution of computer-based studies of language. It has shown how, in the beginning, language studies were limited by the capability of the computers to process the data. As machines become more powerful than before, the present issue is not what computers can do, but what users need to know and do to achieve the most from them.

In chapter three, I reviewed four pieces of software for corpus analysis: AntConc, CQPweb, Quanteda and #LancsBox. With the framework set in 3.2, I was able to detect the standard functionalities offered by CL software mainly used by non-specialist users of corpus data and methods (NSUs).

In chapter four, I identified commonly used software and how these tools' characteristics converge or differ across different subareas of language studies. This was done via a literature investigation of more than 5,000 academic published papers that relied on corpus-based methods.

I analysed researchers using CL software in their daily routine in chapter five. In this chapter, I first outlined the advantages of using a user-centred development process. I then described the steps I took to identify user needs and requirements concerning CL software.

Chapter six discussed the growing need for taking dispersion measures in consideration when performing corpus analysis. It then analysed some CL software that already offers means to measure dispersion, revealing points for improvements. These points were then addressed with the design of a new tool which provides accurate dispersion measures.

A new tool for parallel corpora was introduced in chapter seven. This tool was designed having as target audience, mainly NSUs with interest in translations studies and second language education. The tool's primary function is to make it easier for users to see possible translations for corpus queries in the parallel concordances, without the need to use any external resource, such as translation memories.

8.2 Answers to the research questions

As stated in chapter one, it was expected with this thesis to identify a gap between commonly adopted CL methods and potential usage of corpus data exploration; and to develop and deliver two new CL tools for statistical analysis.

Chapters three covered how CL tools and methods are reported in the literature and chapter four and five discussed how users deal with these tools. Based on the observations made in chapters three to five, two statistical tools for beginners in CL studies were conceived and developed.

8.2.1.1 Advanced Dispersion Tool

This new tool allows for graphical data exploration, which helps the users conceptualise dispersion easily. The users can easily see and compare how often and where the corpus query occur in each text.

8.2.1.2 Parlink Tool

The Parlink tool offers a sophisticated data analysis method which allows users to intuitively identify, in an aligned corpus, related or possible translations for a corpus query. The tool has a very user-friendly interface, with an intuitive display of the strength of the relation between the two words.

8.3 Significance and Contribution

This thesis is significant in terms of theoretical and practical contributions. It will affect the field going forward in terms of impact, by demonstrating common practices in corpus-based studies and in terms of novelty, by delivering two new tools for corpus analysis.

8.3.1 Theoretical contribution

Chapter four presents a methodology to investigate CL software usage through a corpus-based study of academic publications. The procedure I introduced in 4.3 can serve as an incentive for a regular practice of identifying the emergence of new tools and to understand how CL software is used. Similarly, chapter five shows the

importance of studying how corpus linguists relate to computational tools. By observing and listening to users, more user-friendly tools can be developed.

8.3.2 Practical contribution

I believe that with an intuitive tool such as the Dispersion Tool (chapter six), researchers will be more likely to consider dispersion when performing corpus analysis. From that, I expected that users of corpus tools, mainly NSUs, will consider other factors in their data exploration, rather than rely solely on frequency.

With the Parlink tool, users can clearly see (and have a numerical indicator) of the connection between the query performed in a corpus and a type in an aligned corpus. This is particularly beneficial for two groups: language learners and translators. They can promptly identify possible translation equivalents without the need for external resources, such as translation memories.

8.4 Limitations

The previous sections have shown that the work presented in this thesis was successfully within the scope of this project. However, the work here had some limitations due to time constraints.

Although the articles in the literature investigation (chapter four) were sufficient in presenting an overview of tools used for corpus-based research, a broader period combined with the inclusion of other databases could have yield more findings. Relying on various databases from a variety of specialised fields could give a more reliable indication of subfield preferences. Moreover, the dataset could also be manually checked by skimming all articles. That would possibly lead to the retrieval of software that was not identified with the methods described in 4.3, as many tools

were not in the previous existing lists (4.3.2.2.3) neither occurring near words related to software (4.3.2.2.2).

Also in the sense of expanding the data for analysis, the low number of participants and the low variety of research fields were limitations. Although the literature has suggested (5.3.2.1) that five is a sufficient number for user observation, the study could benefit from observations with different pieces of software. The same applies for the software review in chapter three. On the one hand, relatively narrow scope limits the research to the selected tools. On the other hand, it allowed a higher level of detail that I would not be able to cover if I had evaluated a high number of programs. A more detailed analysis for performance, obtained by testing them with corpora of different size and language, would also take place.

Considering the development of the tools, there are still some limitations of the two new features that needs addressing. Those are mainly the zoom feature in the Advanced Dispersion tool and the processing time in the Parlink tool. These minor bugs present in the first versions of the tools reflect the challenges I face when learning two new programming languages during my PhD.

8.5 Future work

Overall, both new tools would benefit from testing with different corpora and different users. The two tools delivered in this thesis were implemented in CQPweb version 3.2. Because CQPweb has just gone through significant changes in its system, the first next step is to edit the codes so they will also work on version 3.3 the most recent version of CQPweb and also the version of CQPweb hosted at Lancaster University. This step is essential because this is possibly the installation of CQPweb with a higher number of users. A high number of accesses means that more users can test the newly

developed tools and provide feedback on their functioning. The logical and next step then is to address the user requirements and needs that will emerge with the use of the tools and undertake user acceptance study.

I also intend to do another literature review capable of identifying more detailed contrasts between a wider range of disciplines and identify other possible tool needs or trends. I am currently considering developing a new tool for corpus lexicography, focusing on the different meanings a word can have in a language according to its variety.

8.6 Concluding remarks

This thesis has made a significant contribution by identifying how CL software is used (8.3.1) and by delivering two new tools (8.3.2). Much remains to be done (8.4), especially in the within the scope of software testing. But despite the limitations, I am confident that the work will help the field move forward.

References

- Abegg, M. G., Flint, P. W. & Ulrich, E. (1999). *The Dead Sea scrolls Bible: the oldest known versions of the books of the Bible translated for the first time into English*. International Clark.
- Abercrombie, D. (1965). Parameters and phonemes. *Studies in phonetics and linguistics*, Oxford University Press, London, pp. 120-124.
- Abran, A., Khelifi, A., Suryan, W. & Seffah, A. (2003). *Software Quality Journal*, 11(4), 325-338. <https://doi.org/10.1023/a:1025869312943>.
- Ahnert, R., Ahnert, S. (2015). Protestant Letter Networks in the Reign of Mary I: A Quantitative Approach. *ELH*, 82(1), 1-1. <https://doi.org/10.1353/elh.2015.0000>
- Aijmer, K. (Ed.). (2009). *Corpora and language teaching* (Vol. 33). John Benjamins.
- Aijmer, K., Altenberg, B., & Johansson, M. (1996). *Text-based contrastive studies in English, presentation of a project*.
- Alcina Caudet, A. (2011). *Teaching and learning terminology*. John Benjamins.
- Altmann, G. (1980). Prolegomena to Menzerath's law. *Glottometrika*, 2(2), 1-10.
- Anderson, W. & Corbett, J. (2009). *Exploring English with online corpora: an introduction*. Palgrave Macmillan.

- Anthony, L. (2002). AntConc (Version 1.0) [Computer Software]. Tokyo, Japan: Waseda University. Available from <http://www.laurenceanthony.net/software>
- Anthony, L. (2017). AntPConc (Version 1.2.1) [Computer Software]. Tokyo, Japan: Waseda University. Available from <https://www.laurenceanthony.net/software>
- Anthony, L. (2019). AntConc (Version 3.5.8) [Computer Software]. Tokyo, Japan: Waseda University. Available from <http://www.laurenceanthony.net/software>
- Anthony, L. & Hardaker, C. (2017). FireAnt (Version 1.1.4) [Computer Software]. Tokyo, Japan: Waseda University. Available from <https://www.laurenceanthony.net/software>
- The Apache Software Foundation*. Apache.org. (2020). Retrieved 16 August 2020, from <https://apache.org/>.
- Aston, G. (1999). “Corpus use and learning to translate”. *Textus*, 12, 289-314.
- Aston, G. & Burnard, L. (1998). *The BNC handbook: exploring the British National Corpus with SARA*. Capstone.
- Atkinson, B. (1987). HyperCard [computer software]. Apple Computer.
- Baayen, H. (2001). *Word frequency distributions*. Kluwer Academic Publishers.
- Baayen, H. (2008). *Analyzing Linguistic Data. A Practical Introduction to Statistics Using R*. New York: Cambridge University Press. <https://doi.org/10.1017/CBO9780511801686>.
- Baayen, R., Piepenbrock, R. & Van Rijn, H. (1993). The CELEX lexical database—Dutch, English, German.
- Baker, M. (1993). Corpus linguistics and translation studies: Implications and applications. In M. Baker, G. Francis & E. Tognini-Bonelli, *Text and technology: In honour of John Sinclair*. John Benjamins.

- Baker, P. (2009). The BE06 Corpus of British English and recent language change. *International Journal Of Corpus Linguistics*, 14(3), 312-337.
<https://doi.org/10.1075/ijcl.14.3.02bak>
- Baker, P., Gabrielatos, C., Khosravini, M., Krzyżanowski, M., McEnery, T. & Wodak, R. (2008). A useful methodological synergy? Combining critical discourse analysis and corpus linguistics to examine discourses of refugees and asylum seekers in the UK press. *Discourse & society*, 19(3), 273-306.
- Baker, P., Hardie, A. & McEnery, T. (2006). *A Glossary of Corpus Linguistics*. Edinburgh University Press.
- Barlow, M. (2002). MonoConc 1.0 [computer software]. *Athelstan*.
- Barlow, M. (2004). MonoConc 2.2 [computer software]. *Athelstan*.
- Baltazani, M. & Kainada, E. (2015). Drifting Without an Anchor: How Pitch Accents Withstand Vowel Loss. *Language and Speech*, 58(1), 84-113.
<https://doi.org/10.1177/0023830914565192>
- Beeby Lonsdale, A., Rodríguez Inés, P. & Sánchez-Gijón, P. (2009). *Corpus Use And Translating*. John Benjamins.
- Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller S. & Matsuo, A. (2018). Quanteda: An R package for the quantitative analysis of textual data” *Journal of Open Source Software*. 3(30), 774. <https://doi.org/10.21105/joss.00774>.
- Bernardini, S. (2002). Exploring New Directions for Discovery Learning. In *Teaching and Learning by Doing Corpus Analysis*, 165–82. Brill.
https://doi.org/10.1163/9789004334236_015.

- Bernardini, S. & Ferraresi, A. (2013). Old needs, new solutions: Comparable corpora for language professionals. In *Building and Using Comparable Corpora* (pp. 303–319). Springer. https://doi.org/10.1007/978-3-642-20128-8_16
- Bestgen, Y. (2017). Getting rid of the Chi-square and Log-likelihood tests for analysing vocabulary differences between corpora. *Quaderns De Filologia - Estudis Lingüístics*, 22(22), 33. <https://doi.org/10.7203/qf.22.11299>
- Bhate, S., & Kak, S. (1991). Pāṇini's Grammar and Computer Science. *Annals of the Bhandarkar Oriental Research Institute*, 72/73(1/4), 79-94. Retrieved October 31, 2020, from <http://www.jstor.org/stable/41694883>
- Biber, D. (1993). Representativeness in Corpus Design. *Literary and Linguistic Computing*, 8(4), 243-257. <https://doi.org/10.1093/lc/8.4.243>
- Biber, D., Conrad, S. & Reppen, R. (1996). Corpus-based investigations of language use. *Annual Review of Applied Linguistics*, 16 (1996), pp. 115-136
- Biber, D., Conrad, S. & Reppen, R. (1998) *Corpus linguistics: investigating language structure and use*. Cambridge University Press.
- Biber, D. & Finegan, E. (1988). Adverbial stance types in English. *Discourse processes*, 11(1), 1-34.
- Biber, D., Johansson, S., Leech, G., Conrad, S. & Finegan, E. (1999). *Longman grammar of spoken and written English*. Longman.
- Biber, D., Reppen R., Schnur E. & Ghanem, R. (2016). “On the (Non)Utility of Juilland’s *D* to Measure Lexical Dispersion in Large Corpora.” *International Journal of Corpus Linguistics* 21 (4):439–64. <https://doi.org/10.1075/ijcl.21.4.01bib>.

- Bick, E. (2014). PALAVRAS-A Constraint Grammar-Based Parsing System for Portuguese. In *Working With Portuguese Corpora* (pp. 279-302). Bloomsbury Academic.
- Blaxill, L. (2016). *A War of Words? Text Mining Political Speeches in Britain in the 19th and 20th Centuries*. UCREL Presentation, Lancaster.
- Bloomfield, L. (1914). *An introduction to the study of language*. H. Holt.
- Bloomfield, L. (1926). A set of postulates for the science of language. *Language*, 2(3), 153-164.
- Borin, L. (Ed.) (2002). *Parallel corpora, parallel worlds*. Rodopi.
- Boulton, A. (2012) Beyond concordancing: Multiple affordances of corpora in university language degrees. *Languages, Cultures and Virtual Communities. Elsevier: Social and Behavioral Sciences*, 34: 33-38.
- Bowker, L. (2018) Corpus linguistics is not just for linguists: Considering the potential of computer-based corpus methods for library and information science research, *Library Hi Tech*, Vol. 36 Issue: 2, pp.358-371. <https://doi.org/10.1108/LHT-12-2017-0271>.
- Bradley, D. C. (1978). *Computational distinctions of vocabulary type*. Unpublished Doctoral dissertation, Massachusetts Institute of Technology.
- Bradley, J., Lancashire, I, Presutti, L. & Stairs, M. (1989). TACT. [Computer Software]. Available from <http://projects.chass.utoronto.ca/tact/index.html>
- Brezina, V. (2018). *Statistics in Corpus Linguistics*. Cambridge University Press. <http://doi.org/10.1017/9781316410899>.
- Brezina, V. & Gablasova, D. (2015). Is There a Core General Vocabulary? Introducing the New General Service List. *Applied Linguistics* 36 (1):1–22. <https://doi.org/10.1093/applin/amt018>.

- Brezina, V., McEnery, T. & Wattam, S. (2015). Collocations in context: A new perspective on collocation networks. *International Journal of Corpus Linguistics*, 20(2), 139-173. <https://doi.org/10.1075/ijcl.20.2.01bre>.
- Brezina, V. & Meyerhoff, M. (2014). Significant or random?: A critical review of sociolinguistic generalisations based on large corpora. *International Journal Of Corpus Linguistics*, 19(1), 1-28. <https://doi.org/10.1075/ijcl.19.1.01bre>.
- Brezina, V., Timperley, M. & McEnery, A. (2018). #LancsBox v. 4.x. [computer software]. Available from: <http://corpora.lancs.ac.uk/lancsbox/>.
- Brhel, M., Meth, H., Maedche, A. & Werder, K. (2015). Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology*, 61, 163-181. <https://doi.org/10.1016/j.infsof.2015.01.004>.
- Burch, B., Egbert J. & Biber, D. (2016). “Measuring and Interpreting Lexical Dispersion in Corpus Linguistics.” *Journal of Research Design and Statistics in Linguistics and Communication Science* 3 (2):189–216.
- Burnard, L. D. (1980). Software review: CLOC. *Computers and the Humanities*, 14(4), 259-260.
- Burton, D. M. (1981). Automated concordances and word indexes: The fifties. *Computers and the Humanities*, 15(1), 1-14.
- Busa, R. (1987). *Fondamenti di informatica linguistica*. Vita e pensiero.
- Carroll, J.B. (1970) An alternative to Juilland’s usage coefficient for lexical frequencies and a proposal for a standard frequency index. *Computer Studies in the Humanities and Verbal Behaviour* 3(2): 61-65.
- Čermák, P. (2019). Intercorp: A parallel corpus of 40 languages. In I. Doval & M. Sánchez Nieto, *Parallel Corpora for Contrastive and Translation Studies: New resources and applications* (pp. 93-102). Jonh Benjamins.

- Chandler, B. (1989). Longman Mini-Concordancer [Computer Software]. Longman.
- Chomsky, N. (1957). *Syntactic Structures*. Mouton.
- Christ, O. (1994). A modular and flexible architecture for an integrated corpus query system. In *Papers in Computational Lexicography* (COMPLEX '94) 22–32.
- Cooper, A. (2004). *The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity* (Vol. 2). Sams.
- Corbeill, A. (2007). The TLL and the Sustaining of Scholarship. *Transactions of the American Philological Association* 137.2: 503-507.
- Davies, M. (2004-). *BYU Corpora*. Available online at <https://corpus.byu.edu/>
- Davies, M. (2008-). *The Corpus of Contemporary American English (COCA): One billion words, 1990-2019*. Available online at <https://www.english-corpora.org/coca/>.
- Davies, M. (2018-). The 14 Billion Word iWeb Corpus. Available online at <https://www.english-corpora.org/iWeb/>
- Davies, M. & Gardner, D. (2010). *A Frequency Dictionary of Contemporary American English. Word sketches, collocates, and thematic lists*. Routledge. ISBN 978-0- 415-39063-4.
- Davies, M. & Kim, J. B. (2019). The advantages and challenges of 'big data': Insights from the 14 billion word iWeb corpus. *Linguistic Research*, 36, 1-34.
- Dias, G., Guillor S. & Pereira Lopes, J. (1999). Language independent automatic acquisition of rigid multiword units from unrestricted text corpora. *Traitement Automatique des Langues Naturelles, Institut d'Etudes Scientifiques*. Cargese 333–339

- Díaz-Negrillo, A., Ballier, N., & Thompson, P. (Eds.). (2013). *Automatic Treatment and Analysis of Learner Corpus Data*. Jonh Benjamins.
<http://doi.org/10.1075/scl.59>
- Dinet, J., Favart, M., & Passerault, J. M. (2004). Searching for information in an online public access catalogue (OPAC): the impacts of information search expertise on the use of Boolean operators. *Journal of computer assisted learning*, 20(5), 338-346.
- Doval, I. (2018). The PaGeS Corpus, a Parallel Corpus of the Contemporary German and Spanish Language. *Revista de Filología Alemana*, 26, 181-201.
- Doval, I., Fernández, Lanza F., Jiménez, Juliá T., Liste Lamas, E. & Lübke, B. (2019). Corpus PaGeS: A multifunctional resource for language learning, translation and cross-linguistic research (103-121). In I. Doval & M. Sánchez Nieto (Eds), *Parallel Corpora for Contrastive and Translation Studies: New resources and applications*. Jonh Benjamins.
- Doval, I. & Sánchez, M. (Eds.) (2019). *Parallel corpora for contrastive and translation studies*. Jonh Benjamins.
- Eckart, T. & Quasthoff, U. (2013) Statistical Corpus and Language Comparison on Comparable Corpora. In: Sharoff S., Rapp R., Zweigenbaum P. & Fung P. (Eds) *Building and Using Comparable Corpora*. Springer.
- Evert, S. (2008). Corpora and collocations. In A. Lüdeling & M. Kytö (eds.), *Corpus Linguistics. An International Handbook*, pages 1212-1248. Mouton de Gruyter.
- Evert, S. & Hardie, A. (2011). Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium. In *Proceedings of the Corpus Linguistics 2011 conference* University of Birmingham.

- Feller J., Finnegan P., Kelly D. & MacNamara M. (2006) *Developing Open Source Software: A Community-Based Analysis of Research*. In: Trauth E.M., Howcroft D., Butler T., Fitzgerald B. & DeGross J.I. (Eds) *Social Inclusion: Societal and Organizational Implications for Information Systems*. IFIP International Federation for Information Processing, vol 208. Springer
- Ferraresi, A. & Bernardini, A. (2019). Building EPTIC: A many-sided, multi-purpose corpus of EU parliament proceedings. In I. Doval & M. Sánchez Nieto (Eds.), *Parallel Corpora for Contrastive and Translation Studies: New resources and applications*. (pp. 123-139) John Benjamins.
- Few, S. (2009). *Now you see it*. Analytics Press.
- Franklin, N. (2013). *The UX book: Process and guidelines for ensuring a quality user experience* by Rex Hartson and Pardha A. Pyla, San Diego: Morgan Kaufmann.
2012. *Communication Design Quarterly Review*, 2(1), 67-72.
- Franzén, K., & Karlgren, J. (2000). *Verbosity and interface design Verbosity and interface design*. SICS Technical report.
- Fillmore, C. J. (1992). 'Corpus Linguistics' or 'Computer-aided armchair linguistics'. In *Directions in corpus linguistics. Proceedings of Nobel Symposium* (Vol. 82, pp. 35-60).
- Ford, N., & Richards, M. (2020). *Fundamentals of Software Architecture*. O'Reilly Media, Inc.
- Francis, W. N., & Kucera, H. (1964). *Brown corpus. Department of Linguistics, Brown University, Providence, Rhode Island, 1*.
- Francis, W. N., Kucera, H., Kučera, H., & Mackie, A. W. (1982). *Frequency analysis of English usage: Lexicon and grammar*. Houghton Mifflin.

- Gale, W. & Church, K. (1991). A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Conference of the Association for Computational Linguistics*, 177-184, Berkeley.
- Gale, W. & Church, K. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*. 19, 1. 75–102.
- Gallego-Hernández, D. (2016). *New insights into corpora and translation*. Cambridge Scholars Publishing.
- Gambette P. & Véronis J. (2010). Visualising a Text with a Tree Cloud. In: Locarek-Junge H. & Weihs C. (eds) *Classification as a Tool for Research. Studies in Classification, Data Analysis, and Knowledge Organization*. Springer.
- Gamallo, P. (2019). Strategies for building high quality bilingual lexicons from comparable corpora. In I. Doval & M. Sánchez Nieto, *Parallel Corpora for Contrastive and Translation Studies: New resources and applications*. John Benjamins.
- Garcia, M., García-Salido, M., & Alonso-Ramos, M. (2019). Discovering bilingual collocations in parallel corpora: A first attempt at using distributional semantics. In I. Doval & M. Sánchez Nieto (Eds), *Parallel Corpora for Contrastive and Translation Studies: New resources and applications*. (pp. 267-279) John Benjamins.
- Garside, R. (1987). The CLAWS word-tagging system. *The Computational analysis of English: A corpus-based approach*. London: Longman, 30-41.
- Gilmore, A. (2015). Research into practice: The influence of discourse studies on language descriptions and task design in published ELT materials. *Language Teaching*, 48(4), 506.

- Gladkova, A., Vanhatalo, U., & Goddard, C. (2016). The semantics of interjections: An experimental study with natural semantic metalanguage. *Applied Psycholinguistics*, 37(4), 841.
- Gleicher, M., Albers, D., Walker, R., Jusufi, I., Hansen, C. D., & Roberts, J. C. (2011). Visual comparison for information visualization. *Information Visualization*, 10(4), 289–309. <https://doi.org/10.1177/1473871611416549>
- Granger, S. (1996). From CA to CIA and back: An integrated approach to computerized bilingual and learner corpora.
- Granger, S. (1998). The computer learner corpus: a versatile new source of data for SLA research (pp. 3-18). na.
- Granger, S. (2002). A bird's-eye view of learner corpus research. In S. Granger, J. Hung and S. Petch-Tyson (Eds.) *Computer Learner Corpora, Second Language Acquisition and Foreign Language Teaching*. (pp. 3–33). Jonh Benjamins.
- Graves, K. (2016). Possibilities and realities. *The Routledge handbook of English language teaching*.
- Greene, B. B., & Rubin, G. M. (1971). *Automatic grammatical tagging of English*. Department of Linguistics, Brown University.
- Gries, S. T. (2008). Dispersions and adjusted frequencies in corpora. *International Journal of Corpus Linguistics*, 13(4), 403–437. <http://doi.org/10.1075/ijcl.13.4.02gri>
- Gries, S. T. (2009). *Quantitative Corpus Linguistics With R: A Practical Introduction*. New York: Routledge.
- Gries, S. T. (2010). Useful statistics for corpus linguistics. *A Mosaic of Corpus Linguistics: Selected Approaches*, 269–291.

- Gries, S. T. (2013). *Statistics for Linguistics with R: A Practical Introduction. Language* (2nd ed.). De Gruyter Mouton. <http://doi.org/10.1353/lan.2012.0032>
- Gries, S. T. (2015). Some current quantitative problems in corpus linguistics and a sketch of some solutions. *Language and Linguistics*, 16(1), 93–117. <http://doi.org/10.1177/1606822X14556606>
- Gries, S. T. (2016). *Quantitative corpus linguistics with R: A practical introduction*. Taylor & Francis.
- Gries, S. T. (forthcoming). Analyzing dispersion. In Magali Paquot & Stefan Th. Gries (Eds.), *Practical handbook of corpus linguistics*. Springer.
- Gries, S. T., & Wulff, S. (2012). Regression analysis in translation studies. *Quantitative methods in corpus-based translation studies: A practical guide to descriptive translation research*, 35, 52.
- Guinovart, X. (2019). Enriching parallel corpora with multimedia and lexical semantics: From the CLUVI Corpus to WordNet and SemCor. In I. Doval & M. Sánchez Nieto (Eds.), *Parallel Corpora for Contrastive and Translation Studies: New resources and applications*. (pp141-156). Jonh Benjamins.
- Gupta, D., Ahlawat, A. & Sagar, K. (2014) A critical analysis of a hierarchy based Usability Model. *International Conference on Contemporary Computing and Informatics (IC3I)*, Mysore, 2014, pp. 255-260, doi: 10.1109/IC3I.2014.7019810.
- Halliday M., Teubert W., Yallop C., & Cermáková, A. (2004). *Lexicology and corpus linguistics*. Continuum
- Hammond, M. (2002). *Programming for linguists: Java technology for language researchers*. Blackwell.

- Hardie, A. (2012). CQPweb — combining power, flexibility and usability in a corpus analysis tool. *International Journal of Corpus Linguistics*, 17(3), 380–409.
<http://doi.org/10.1075/ijcl.17.3.04har>
- Hardie, A. (2014). Modest XML for Corpora: Not a standard, but a suggestion. *ICAME Journal*, 38(1), 73-103.
- Hardt-Mautner, G. (1995). How does One Become a Good European?: The British Press and European Integration. *Discourse & Society*, 6(2), 177-205.
- Hareide, L., & Hofland, K. (2012). Compiling a Norwegian-Spanish parallel corpus: Methods and challenges. In: Oakes, M & Ji, M (Eds). *Quantitative Methods in Corpus-Based Translation Studies: A practical guide to descriptive translation research*.
- Hartson, H. R. & Pyla, P. S. (2012). *The UX book: process and guidelines for ensuring a quality user experience*. Morgan Kaufmann.
- Hassenzahl, M. (2004) The Interplay of Beauty, Goodness, and Usability in Interactive Products. *Human-Computer Interaction*, 19:4, 319-349, DOI: 10.1207/s15327051hci1904_2
- Hearst, M. (2009). *Search user interfaces*. Cambridge University Press.
- Heid, U. (2008). Corpus linguistics and lexicography. *Corpus Linguistics. An International Handbook*, 1, 131-153.
- Hertzum, M., & Frøkjær, E. (1996). Browsing and querying in online documentation: a study of user interfaces and the interaction process. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(2), 136-161.
- Hewavitharana, S. & Vogel, S. (2013). Extracting parallel phrases from comparable data. In *Building and Using Comparable Corpora* (pp. 191-204). Springer.

- Hicks, M. & Foster, J. S. (2010). Score: Agile research group management. *Communications of the ACM*, 53(10), 30-31.
- Hockey, S., & Martin, J. (1987). The Oxford concordance program version 2. *Literary & Linguistic Computing*, 2(2), 125-131.
- Hockey, S. (2004). The history of humanities computing. *A companion to digital humanities*, 3-19.
- Hockey, S. (2004). The history of humanities computing. *A companion to digital humanities*, 3-19.
- Hoffmann, S., & Evert, S. (2006). BNC web (CQP-edition): The marriage of two corpus tools. *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, (3), 177–195.
- Hoffmann, S., Evert, S., Smith, N., Lee, D. Y. W. & Berglund Prytz, Y. (2008). *Corpus Linguistics with BNCWeb — a Practical Guide*. Peter Lang.
- Holtzblatt, K., & Beyer, H. (2014). Contextual design: evolved. *Synthesis Lectures on Human-Centered Informatics*, 7(4), 1-91.
- International Organization for Standardization. (1998). *ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs): Part 11: Guidance on usability*.
- Janssen, M. (2018). TEITOK as a tool for Dependency Grammar. In: *Procesamiento del Lenguaje Natural*, vol. 61, 185-188.
- Ji, M. (2012). Hypothesis testing in corpus-based literary translation studies. In Oakes, M. P., & Ji, M. (Eds.) *Quantitative Methods in Corpus-based Translation Studies*. (53-72). John Benjamins.

- Ji, M. & Oakes, M. P. (2012). A corpus study of early English translations of Cao Xueqin's *Hongloumeng*. (pp. 177-208) In Oakes, M. P., & Ji, M. (Eds.) *Quantitative Methods in Corpus-based Translation Studies*. Benjamins.
- Johansson, S. (1998). On the role of corpora in cross-linguistic research. *Corpora and cross-linguistic research: Theory, method, and case studies*, (24), 3.
- Johns, T. (1986). Micro-concord: A Language Learner's Research Tool. *System*, 14(2), 151–162.
- Johns, T. (1991). Should you be persuaded: Two Samples of data-driven learning materials. (pp. 1–16) In T. Johns & P. King (Eds.), *Classroom Concordancing*.
- Johns, T. (1994) From printout to handout: Grammar and Vocabulary teaching in the context of data-driven learning. In Odlin, T. (Ed). *Perspectives on pedagogical grammar*. Cambridge University Press.
- Johnson, K. (1992). *Communicate in Writing. A Functional Approach to Writing Through Reading Comprehension*.(12. Impr.)-Suppl: *Teacher's Book*. Longman.
- Jones, S. E. (2016). *Roberto Busa, SJ, and the emergence of humanities computing: the priest and the punched cards*. Routledge.
- Juilland, A., & Chang-Rodríguez, E. (1964). *Frequency dictionary of Spanish words*. Mouton de Gruyter..
- Juilland A.G., Brodin D.R., & Davidovitch C. (1970) *Frequency dictionary of French words*. Mouton de Gruyter.
- Juilland, A., & Traversa, V. (1973). *Frequency dictionary of Italian words*. Walter de Gruyter GmbH & Co KG.
- Kaye, G. (1990). A Corpus Builder and Real-Time Concordance. *Theory and practice in corpus linguistics*, (4), 137.
- Kennedy, G. (2014). *An introduction to corpus linguistics*. Routledge.

- Kilgarriff, A. (1997). I Don't Believe in Word Senses. *Computers and the Humanities*, 31(2), 91–113. <http://doi.org/10.1023/A:1000583911091>
- Kilgarriff, A. (2001). Comparing Corpora. *International Journal of Corpus Linguistics*, 6(1), 97-133. <https://doi.org/10.1075/ijcl.6.1.05kil>
- Kilgarriff, A., Rychly, P., Smrz, P., & Tugwell, D. (2004). The Sketch Engine. *Proceedings of the Eleventh EURALEX International Congress*, 105–116.
- Kilgarriff, A., Rychlý, P., Kovář, V., & Baisa, V. (2012). Finding multiwords of more than two words. *Proceedings of EURALEX 2012*.
- Kirk, U. (2012). The Modularity of Aesthetic Processing and Perception in the Human Brain. *Aesthetic Science: Connecting Minds, Brains, and Experience*, 318-336.
- Kirk, J. M. (2016). The pragmatic annotation scheme of the SPICE-Ireland corpus. *International Journal of Corpus Linguistics*, 21(3), 299-322.
- Kirjavainen, M., Kidd, E., & Lieven, E. (2017). How do language-specific characteristics affect the acquisition of different relative clause types? Evidence from Finnish. *Journal of Child Language*, 44(1), 120-157.
- Kirk, J. M. (1994). Corpus-Concordance-Database-VARBRUL'. *Literary and linguistic computing*, 9(4), 259-266.
- Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.
- Koh, A. (2014). Inspecting the nineteenth-century literary digital archive: Omissions of empire. *Journal of Victorian Culture*, 19(3), 385-395.
- Krause, T., & Zeldes, A. (2016). ANNIS3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities*, 31(1), 118-139.
- Kučera, H., & Francis, W. N. (1967). *Computational analysis of present-day American English*. Brown University Press.

- Kuniavsky, M. (2003). *Observing the user experience a practitioner's guide to user research*. Morgan Kaufmann.
- Kunz, K. & Steiner, E. (2010). Towards a comparison of cohesive reference in English and German: System and text. *Linguistics & the Human Sciences*, 6, 1-3, pp 219-251.
- Kupiec, J. (1993). An algorithm for finding noun phrase correspondences in bilingual corpora. (pp. 17-22). In *31st Annual Meeting of the Association for Computational Linguistics*.
- Laviosa, S. (1997). How comparable can comparable corpora be? *Target. International Journal of Translation Studies*, 9(2), 289-319.
- Leech, G. (1991). The State of the art in corpus linguistics, English corpus linguistics Aijmer K., Altenberg B.(eds) pp. 8-29.
- Lehmann, H-M., Schneider, P. & Hoffmann, S. (2000). "BNCweb". In J. Kirk (ed.), *Corpora Galore: Analysis and Techniques in Describing English*. Amsterdam: Rodopi, 259–266.
- Lerner, F. A. (1999). *Libraries through the ages*. A&C Black.
- Levshina, N. (2015). *How to do Linguistics with R*. <http://doi.org/10.1075/z.195>
- Lewandowska-Tomaszczyk, B. (1987). *Conceptual analysis, linguistic meaning, and verbal interaction*. Wydawn. Uniwersytetu Łódzkiego.
- Lewandowska-Tomaszczyk, B. (2012). Explicit and tacit: An interplay of the quantitative and qualitative approaches to translation. In Oakes, M. P., & Ji, M. (Eds.) *Quantitative methods in corpus-based translation studies* (pp. 1-34). John Benjamins.
- LIFE magazine. (1957). Bible labor of years is done in 400 hours. (Vol 42, n 7), 92.

- Lijffijt, J., & Gries, S. T. (2012). Correction to Stefan Th. Gries' "Dispersions and adjusted frequencies in corpora" *International Journal of Corpus Linguistics* 13: 4 (2008), 403-437.
- Lindgaard, G., & Dudek, C. (2003). What is this evasive beast we call user satisfaction?. *Interacting with computers*, 15(3), 429-452.
- Lindgaard, G., & Chattratchart, J. (2007). Usability testing: what have we overlooked?. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1415-1424).
- Love, R., Dembry, C., Hardie, A., Brezina, V. & McEnery, T. (2017). The Spoken BNC2014: designing and building a spoken corpus of everyday conversations. *International Journal of Corpus Linguistics*, 22(3), 319-344.
- Luhn, H. P. (1966). Keyword-in-Context Index for Technical Literature (KWIC Index). *Readings in automatic language processing*, 1, 159.
- Lyne, A. (1985). Dispersion. *The vocabulary of French business correspondence*, 101-124.
- Machálek, T. (2020). KonText: Advanced and Flexible Corpus Query Interface. In *Proceedings of The 12th Language Resources and Evaluation Conference* (pp. 7003-7008).
- Mahlberg, M. (2012). The corpus stylistic analysis of fiction—or the fiction of corpus stylistics?. In *Corpus Linguistics and Variation in English* (pp. 77-95). Rodopi.
- Mahlberg, M., Stockwell, P., de Joode, J., Smith, C. & O'Donnell, M. B. (2016). CLiC Dickens: Novel uses of concordances for the integration of corpus stylistics and cognitive poetics. *Corpora*, 11(3), 433–463.

- Maia, B. & Santos, D. (2018). Language, emotion, and the emotions: The multidisciplinary and linguistic background. *Language and Linguistics Compass*, 12(6), e12280. <http://doi.org/10.1111/lnc3.12280>.
- Malmkjaer, K. (2003). On a pseudo-subversive use of corpora in translator training. *Corpora in translator education*, 119-134. Routledge.
- Manning, C., & Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT press.
- Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank.
- Mauranen, A. (2004). Contrasting languages and varieties with translational corpora. *Languages in Contrast*, 5(1), 73-92.
- Mazibuko, G. & Ndebele, H. (2017). Corpora and Corpus Tools for Indigenous African Languages: The Case of an IsiZulu-English Spoken Word Code-Switching Corpus. *Mankind Quarterly*, 58(2), 268.
- McEnery, T., & Baker, P. (Eds.). (2015). *Corpora and discourse studies: Integrating discourse and corpora*. Springer.
- McEnery, T., & Hardie, A. (2012). *Corpus Linguistics: Method, Theory and Practice*. Cambridge University Press.
- McEnery, A. M., & Wilson, A. (2001). *Corpus linguistics: an introduction*. Edinburgh University Press.
- McEnery, T., Xiao, R., & Tono, Y. (2006). *Corpus-based language studies: An advanced resource book*. Taylor & Francis.
- McKeown, K., Smadja, F., & Hatzivassiloglou, V. (1996). Translating collocations for bilingual lexicons: A statistical approach.

- Molés-Cases, T. & Oster, U. (2019). Indexation and analysis of a parallel corpus using CQPweb. *Parallel Corpora for Contrastive and Translation Studies: New resources and applications*, 90, 197.
- Moran, S., & Cysouw, M. (2018). *The Unicode cookbook for linguists*. Language Science Press.
- Morris, W. (1969). *American heritage dictionary of the English language*. American heritage.
- Murray, K. M. E. (2001). *Caught in the web of words: James AH Murray and the Oxford English Dictionary*. Yale University Press.
- Neumann, S. (2012). Applying register analysis to varieties of English. *Anglistentag 2011 Freiburg proceedings*, 75-94.
- Neumann, S. & Hansen-Schirra, S. (2013). Exploiting the incomparability of comparable corpora for Contrastive Linguistics and Translation Studies. In Sharoff, S., Rapp, R., Zweigenbaum, P., & Fung, P. (Eds.) *Building and Using Comparable Corpora* (pp. 321-335). Springer.
- Nielsen, J. (2000). *Designing web usability: the practice of simplicity*. New Riders.
- Norman, D. A. (2004). *Emotional design: why we love (or hate) everyday things*. Basic Books.
- Nyhan, J. & Flinn, A. (2016). *Computation and the humanities: towards an oral history of digital humanities*. Springer Nature.
- Oakes, M. P. (2012). Describing a translational corpus. In Oakes, M. P., & Ji, M. (Eds.) *Quantitative methods in corpus-based translation studies*, 115-147.
- Och, F. J. & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29 (1), 19-51.

- Olesen-Bagneux, O. (2014). The memory library: How the library in Hellenistic Alexandria worked. *KO KNOWLEDGE ORGANIZATION*, 41(1), 3-13.
- Olohan, M. (2004). *Introducing corpora in translation studies*. Routledge.
- Olohan, M. & Baker, M. (2000). Reporting that in translated English. Evidence for subconscious processes of explicitation? *Across languages and cultures*, 1(2), 141-158.
- Owens, T. (2011). Defining data for humanists: Text, artifact, information or evidence. *Journal of Digital Humanities*, 1(1), 6-8.
- Paquot, M. & Plonsky, L. (2017). Quantitative research methods and study quality in learner corpus research. *International Journal of Learner Corpus Research*, 3(1), 61-94. <https://doi.org/10.1075/ijlcr.3.1.03paq>
- Partington, A. (2004). Corpora and discourse, a most congruous beast. *Corpora and discourse*, 11-20.
- Parush, A., Nadir, R. & Shtub, A. (1998). Evaluating the layout of graphical user interface screens: Validation of a numerical computerized model. *International Journal of Human-Computer Interaction*, 10(4), 343-360.
- Patton, J. M. & Can, F. (2012). Determining translation invariant characteristics of James Joyce's *Dubliners*. *Quantitative Methods in Corpus-Based Translation Studies. Philadelphia: John Benjamins*, 209-230.
- Pawłowski, A. (2008). Prolegomena to the History of Corpus and Quantitative Linguistics. Greek Antiquity. *Glottology*, 1(1), 48-54.
- Piao, S. S., Bianchi, F., Dayrell, C., D'egidio, A., & Rayson, P. (2015). Development of the multilingual semantic annotation system. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1268-1274).

- Piao, S. S., Rayson, P., Archer, D., Bianchi, F., Dayrell, C., El-Haj, M., & Nawab, R. M. A. (2016). Lexical coverage evaluation of large-scale multilingual semantic lexicons for twelve languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 2614-2619).
- Pozzi, F. A., Fersini, E., Messina, E., & Liu, B. (2016). *Sentiment analysis in social networks*. Morgan Kaufmann.
- Pym, A. (2007). Natural and directional equivalence in theories of translation. *Target. International Journal of Translation Studies*, 19(2), 271-294.
- Rabadán, R. (2005). Proactive Description for Useful Applications: Researching Language Options for Better Translation Practice?. *Meta: journal des traducteurs/Meta: Translators' Journal*, 50(4).
- Rabadán, R. (2019). Working with parallel corpora. *Parallel Corpora for Contrastive and Translation Studies: New resources and applications*, 90, 57.
- Reddick, A., & Johnson, S. (1996). *The Making of Johnson's Dictionary 1746-1773*. Cambridge University Press.
- Reed, A. (1978). *CLOC User Guide*. University of Birmingham, Computer Centre.
- Reed, A. (1997). Simple Concordance Program [Computer Software]. Available from <http://www.textworld.com/scp/>.
- Reinke, U. (2018). State of the art in translation memory technology. *Language technologies for a multilingual Europe*, 4, 55.
- Robins, R. H. (2013). *A short history of linguistics*. Routledge.
- Rodríguez-Inés, P., & Gallego-Hernández, D. (2016). Corpus Use and Learning to Translate, almost 20 years on. *Cadernos de Tradução*, 36(SPE), 9-13.

- Rodríguez-Inés, P. (2010). Electronic corpora and other ICT (Information and communication technologies) tools: an integrated approach to translation teaching". *The Interpreter and Translator Trainer* 4(2), 251-282
- Rosengren, I. (1971) The quantitative concept of language and its relation to the structure of frequency dictionaries. *Études de linguistique appliquée (Nouvelle Série)* 1: 103-127.
- Russell, D. B. (1967). *Cocoa Manual*. Science Research Council Atlas Computer Laboratory.
- Rybicki, J. (2012). The great mystery of the (almost) invisible translator. *Quantitative Methods in Corpus-Based Translation Studies: A practical guide to descriptive translation research*, 231.
- Rychlý, P. (2007). Manatee/Bonito-A Modular Corpus Manager. In *RASLAN* (pp. 65-70).
- Sanjurjo-González, H., & Izquierdo, M. (2019). P-ACTRES 2.0: A parallel corpus for cross-linguistic research. In *Parallel Corpora for Contrastive and Translation Studies: New resources and applications* (pp. 215-231). John Benjamins.
- Santos, D., & Bick, E. (2000). Providing Internet access to Portuguese corpora: the AC/DC project. In *quot; In Maria Gavrilidou; George Carayannis; Stella Markantonatou; Stelios Piperidis; Gregory Stainhauer (ed) Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000) (Athens 31 May-2 June 2000)*.
- Šarčević, S. (2016). *Language and culture in EU law: multidisciplinary perspectives*. Routledge.
- Sardinha, T. B. (2000). Lingüística de corpus: histórico e problemática. *Delta: documentação de estudos em lingüística teórica e aplicada*, 16(2), 323-367.

- Savický P. & Hlaváčová, J. (2002) Measures of word commonness. *Journal of Quantitative Linguistics* 9(3): 15-31.
- Schmid, H. (1994). TreeTagger-a language independent part-of-speech tagger. <http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>.
- Scott, M. (1996). Wordsmith Tools. [Computer Software]. Oxford University Press. ISBN 0-19-458984-6.
- Scott, M. (2008). “Developing WordSmith” in special issue of *International Journal of English Studies* entitled Monograph: Software-aided Analysis of Language, edited by M. Scott, P. Pérez-Paredes & P. Sánchez-Hernández. Vol 8, No. 1. pp. 153-172.
- Scott, M. (2012). Looking back or looking forward in corpus linguistics: What can the last 20 years suggest about the next? *Iberica*, 24, 75–86.
- Scott, M. (2020). WordSmith Tools version 8. [Computer Software]. Lexical Analysis Software.
- Seuss, D. (1957). *The cat in the hat*. Random House.
- Shi, S., & Fung, P. (2013). Mining parallel documents using low bandwidth and high precision CLIR from the heterogeneous web. In *Building and Using Comparable Corpora* (pp. 21-49). Springer.
- Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S. & Elmqvist, N. (2017). *Designing the user interface: strategies for effective human-computer interaction* (6th ed.). Pearson.
- da Silva, E. B., Orenha-Ottaiano, A. & Babini, M. (2017). Identification of the most common phraseological units in the English language in academic texts: Contributions coming from corpora. *Acta Scientiarum Language and Culture*, 39(4), 345–353. <http://doi.org/10.4025/actascilangcult.v39i4.31811>

- Simonsen, J., & Kensing, F. (1997). Using ethnography in contextual design. *Communications of the ACM*, 40(7), 82-88.
- Sinclair, S. & Rockwell, G. (2020). Voyant Tools. [Computer Software]. Available from <http://voyant-tools.org/>.
- Smadja, F. (1992). XTRACT: an overview. *Computers and the Humanities*, 26(5-6), 399-413.
- Škrabal, M. & Vavřín, M. (2017). The Translation Equivalents Database (Treq) as a Lexicographer's Aid. In *Electronic lexicography in the 21st century. Proceedings of eLex 2017 conference*. Lexical Computing.
- Smith, N., Hoffmann, S. & Rayson, P. (2008). Corpus Tools and Methods, Today and Tomorrow: Incorporating Linguists' Manual Annotations. *Literary and Linguistic Computing*, 23, 2, 163–180, <https://doi.org/10.1093/lc/fqn004>.
- Soehn, J. P., Zinsmeister, H., & Rehm, G. (2008). Requirements of a user-friendly, general-purpose corpus query interface.
- Spool, J., & Schroeder, W. (2001). Testing web sites: Five users is nowhere near enough. In *CHI'01 extended abstracts on Human factors in computing systems* (pp. 285-286).
- Svartvik, J. (2007). Corpus linguistics 25+ years on. In *Corpus linguistics 25 years on* (pp. 9-25). Brill Rodopi.
- Tasman, P. (1958). *Indexing the Dead Sea Scrolls: By Electronic Literary Data Processing Methods*.
- Thomas, M. (2011). *Fifty key thinkers on language and linguistics* (pp. 2-6). Taylor & Francis Group.
- Thorndike, E. L. (1921). *Educational Psychology: Mental work and fatigue and individual differences and their causes*. Teachers college, Columbia University.

- Thorndike, E. L., & Lorge, I. R. V. I. N. G. (1944). The teacher's word book of 30,000 words. Teachers college. Columbia University.
- Tidwell, J., Brewer, C., & Valencia, A. (2020). *Designing Interfaces, 3rd Edition* (3rd ed.). O'Reilly.
- Tiedemann, J. (2011). Bitext alignment. *Synthesis Lectures on Human Language Technologies*, 4(2), 1-165.
- Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *Lrec* (Vol. 2012, pp. 2214-2218).
- Tillmann, C., & Hewavitharana, S. (2013). A unified alignment algorithm for bilingual data. *Nat. Lang. Eng.*, 19(1), 33-60.
- Tognini-Bonelli, E. (2001). *Corpus Linguistics at Work*. John Benjamins. <http://doi.org/10.1075/scl.6>
- Tóth, K., Farkas, R., & Kocsor, A. (2008). Sentence alignment of Hungarian-English parallel corpora using a hybrid algorithm. *Acta Cybernetica*, 18(3), 463-478.
- Tribble, C. T. (2012). *Managing change in ELT: lessons from experience*. The British Council.
- Tribble, C. (2015). Teaching and language corpora: Perspectives from a personal journey. In Leńko-Szymańska, A. & Boulton A. (Eds.) *Multiple Affordances of Language Corpora for Data-driven learning*: pp. 37-62. John Benjamins.
- Tymoczko, M. (1998). Computerized corpora and the future of translation studies. *Meta: journal des traducteurs/Meta: Translators' Journal*, 43(4), 652-660.
- The Unicode Consortium*. The Unicode Consortium. (2020). Retrieved 16 August 2020, from <https://unicode.org/>.

- Varantola, K. (2003). Translators and disposable corpora. *Corpora in translator education*, 55-70.
- Varga, D. (2012). *Natural Language Processing of Large Parallel Corpora*. Unpublished Ph.D. thesis, Eotvos Loránd University.
- Varga, D., Halácsy, P., Kornai, A., Nagy, V., Németh, L., & Trón, V. (2007). Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, 292, 247.
- Volk, M., Graën, J., & Callegaro, E. (2014). Innovations in parallel corpus search tools.
- Washtell, J. (2007) Co-dispersion by nearest-neighbour: adapting a spatial statistic for the development of domain-independent language tools and metrics. Unpublished M.Sc. thesis, School of Computing, University of Leeds.
- Weik, M. (2000). *Computer science and communications dictionary*. Springer Science & Business Media.
- Weisser, M. (2009). *Essential Programming for Linguistics*. Edinburgh University Press.
- Wiechmann, D., & Fuhs, S. (2006). Concordancing software. *Corpus linguistics and linguistic theory*, 2(1), 107-127.
- Wilkinson, M. (2011). WordSmith Tools: The best corpus analysis program for translators? *Translation Journal*, Vol. 15, No 3. Online at: <http://translationjournal.net/journal/57corpus.htm>
- Wilson, J., Hartley, A., Sharoff, S., & Stephenson, P. (2010). Advanced corpus solutions for humanities researchers. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation* (pp. 769-778).

- Winter, T. N. (1999). Roberto Busa, SJ, and the invention of the machine-generated concordance. *Faculty Publications, Classics and Religious Studies Department*, 70.
- Xiao, R. (2006). Xaira—an XML aware indexing and retrieval architecture. *Corpora*, 1(1), 99-103.
- Young, L., & Soroka, S. (2012). Affective news: The automated coding of sentiment in political texts. *Political Communication*, 29(2), 205-231.
- Zahidi, Z., Lim, Y. P., & Woods, P. C. (2014). Understanding the user experience (UX) factors that influence user satisfaction in digital culture heritage online collections for non-expert users. In *2014 Science and Information Conference* (pp. 57-63). IEEE.
- Zanettin, F. (1998). Bilingual comparable corpora and the training of translators. *Meta: journal des traducteurs/Meta: Translators' Journal*, 43(4), 616-630.
- Zanettin, F. (2000). Parallel corpora in translation studies: Issues in corpus design and analysis. *Intercultural Faultlines*, 105-118.
- Zanettin, F. (2012). Translation practices explained: translation-driven corpora. *St Jerome Publishing*.
- Zariņa, I., Ņikiforovs, P., & Skadiņš, R. (2015). Word alignment based parallel corpora evaluation and cleaning using machine learning techniques. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation* (pp. 185-192).
- Zipf, G. K. (1935). The psycho-biology of language. *Boston, Houghton*. Cited by Bookstein, Abraham (1979). *Explanation of the bibliometric law*. *Collection Management*, 3(2-3), 151-161.

Zipf, G. K. (1949). Human behaviour and the principle of least-effort. Cambridge MA
edn. *Addison-Wesley*.

Zipser, F., Zeldes, A., Ritz, J., Romary, L., & Leser, U. (2011). Pepper: Handling a
multiverse of formats. 33. *Jahrestagung der Deutschen Gesellschaft für
Sprachwissenschaft*

