Cambridge Handbook of English Corpus Linguistics
Chapter 2: Computational Tools and Methods for Corpus Compilation and Analysis[1]

Paul Rayson
UCREL, Lancaster University

1. Introduction

The growing interest in corpus linguistics methods in the 1970s and 1980s was largely enabled by the increased power of computers and the use of computational methods to store and process language samples. Before this, even simple methods for studying language such as extracting a list of all the different words in a text and their immediate contexts was incredibly time consuming and costly in terms of human effort. Only concordances of books of special importance such as the Qur'an, the Bible and the works of Shakespeare were made before the 20th century and required either a large number of scholars or monks or a significant investment in time by a single individual, in some cases more than ten years of their lives. In these days of web search engines and vast quantities of text that is available at our finger tips, the end user would be mildly annoyed if a concordance from a one billion word corpus took more than five seconds to be displayed.

Other text rich disciplines can trace their origins back to the same computing revolution. Digital Humanities scholars cite the work of Roberta Busa working with IBM in 1949 who produced his Index Thomisticus, a computer-generated concordance to the writings of Thomas Aquinas. Similarly, lexicographers in the 19th century used millions of handwritten cards or quotation slips but the field was revolutionised in the 1980s with the creation of machine-readable corpora such as COBUILD and the use of computers for searching and finding patterns in the data.

This chapter presents an introductory survey of computational tools and methods for corpus construction and analysis. The corpus research process involves three main stages: corpus compilation, annotation, and retrieval (see Rayson 2008). A corpus first needs to be compiled via transcription, scanning, or sampling from on-line sources. Then, the second stage is annotation, through some combination of manual and automatic methods to add tags, codes, and documentation that identify textual and linguistic characteristics. A snapshot of tools and methods that support the first and second stages of the corpus research process are described in sections 2.1 and 2.2.

Retrieval tools and methods enable the actual linguistic investigations based on corpora: i.e. frequency analysis, concordances, collocations, keywords and n-grams. These tools are introduced in Section 2.3, together with a brief timeline tracing the historical development of retrieval tools and methods and the current focus on web-based interfaces for mega-corpora. Corpus tools and methods are now being applied very widely to historical

[1] The survey presented in this paper was supported by the ESRC Centre for Corpus Approaches to Social Science, ESRC grant reference ES/K002155/1

data, learner language and online varieties (Usenet, Emails, Blogs and Microblogs) so I also consider the effect of non-standard or 'dirty data' on corpus tools and methods, e.g. where spelling variation affects their robustness. Although the focus in this chapter and the handbook is on tools and methods for English corpus linguistics, I highlight issues of support for other languages and corpora and tools that support multiple languages where they are relevant.

Corpus linguistics as a discipline has matured in parallel with the development of more powerful computers and software tools. In section 2.4, I will reflect on the question of whether corpus linguistics is now tool-driven, i.e. whether researchers can only ask the research questions that are supported by the existing tools and methods, and whether other important questions are not tackled due to a lack of tool support. I highlight some limitations of the existing tools and methods, which include for example limited support of manual categorisation of concordance lines and categorisation of key words. The empirical study presented in section 3 will investigate the relative strengths and weakness of tools and methods for studying n-grams, also called lexical bundles (Biber et al, 1999), recurrent combinations (Altenberg, 1998) or clusters (Scott[2]). This will serve to highlight practical analysis problems such as the vast quantity of n-grams that are extracted from a corpus, and overlaps between shorter 2-grams that form part of longer 3, 4 or 5-grams. An approach called c-grams (collapsed-grams) will be presented alongside other proposed solutions to this problem.

Finally, the chapter will conclude in section 4 with a peek into the future taking some current tools, describing what research gaps need to be addressed and what tools and methods might look like in the future. Improvements in speed and usability of corpus tools are important as well as interoperability between the tools. In addition, the sheer scale of mega corpora such as those derived from online varieties of language suggests that better support for the visualisation of results would be beneficial.

2. Survey of tools and methods

Corpus linguistic research differs from most research in theoretical linguistics in that the language sample analysed, and the methods used for that analysis, are central concerns. As a result, most research articles in corpus linguistics include discussion of corpus compilation, annotation, and/or the computational methods used to retrieve linguistic data. To illustrate this, I have undertaken a small quantitative analysis of recent papers published in the field and how often they discuss each of the three considerations. The source data for this analysis is the academic papers published in four leading corpus linguistics journals:

- *International Journal of Corpus Linguistics* published by John Benjamins[3]
- *Corpora* published by Edinburgh University Press[4]

[2] http://www.lexically.net/wordsmith/
[3] http://benjamins.com/#catalog/journals/ijcl/main
[4] http://www.euppublishing.com/journal/cor

- *Corpus Linguistics and Linguistic Theory* published by De Gruyter[5]
- *ICAME Journal* published by UCREL, Lancaster University[6]

The analysis considers papers published in 2012, and for the International Journal of Corpus Linguistics it includes only the first two issues for that year since these were the ones published at the time of writing (early 2013). A total of 32 papers in the four journals have been categorised into the three main areas in the survey: compilation, annotation and retrieval. Although the vast majority of studies refer to corpus data as would be expected, I have only counted a paper in the compilation set where it refers to new corpus compilation activity rather than the use of a pre-existing corpus.

Table 1 provides the results of the survey. Each paper could count towards any or possibly all three of the categories; therefore the figures in the total column do not add up to 32 since some papers can fall into multiple categories.

	IJCL	Corpora	CLLT	ICAME	Total
Compilation	5	5	2	0	12
Annotation	1	3	3	1	8
Retrieval	8	6	9	4	27
No computational methods	0	1	1	0	2
Number of papers	9	8	11	4	

Table 1: Quantitative analysis of papers published in 2012

Table 1 shows that of the 32 papers published in the four journals in 2012, there are 27 (84%) which describe some sort of retrieval tool or method. 12 (38%) papers describe some new corpus compilation activity and only 8 (25%) include tools or methods related to corpus annotation. Although this is just a small snapshot and the results would no doubt change if another year's worth of published papers were considered, it does illustrate the focus of recent work in the field and serves to justify the choice of the three categories presented in this chapter. The survey also throws up another interesting result. In the vast majority of cases, the software tools used in these papers are employed off-the-shelf, pre-built rather than tools which researchers have created themselves or programming languages and environments such as R where corpus methods can be implemented by the corpus researcher directly. Such approaches are described elsewhere in this handbook (e.g. chapter three) and in other publications (Mason, 2000; Gries, 2009). Therefore in this chapter, I will focus on these pre-existing software tools that implement the methods that are described.

Many similar computational methods and tools would be seen if areas such as content analysis, Computer Assisted Qualitative Data Analysis (CAQDAS), digital humanities and text mining had been considered, however, in this chapter, the scope needs to be limited carefully to computational

[5] http://www.degruyter.com/view/j/cllt
[6] http://icame.uib.no/journal.html

methods and tools employed for corpus linguistics research. The following sub-sections will focus in turn on tools and methods related to the three key phases of corpus linguistics methodology that have already been highlighted, i.e. compilation, annotation and retrieval. After that I will reflect on the current state of the art in corpus tools and methods.

2.1 Compilation

Unless the corpus linguist is planning to use an existing off-the-shelf corpus, the first thing they need to do is to compile one of their own. Unfortunately, considering the plethora of text books in the field, it is the practical aspects of this process that are dealt with least out of the three key phases of corpus linguistics methodology. Kennedy (1998: 70) states that there are three stages to corpus compilation: “corpus design, text collection or capture and text encoding or markup” and reminds us of the importance of a catalogue which is needed along with safe backup and storage. Adolphs (2006: 21) also says there are three stages in the compilation process: “data collection, annotation and mark-up, and storage” and the overlap between these two definitions can be clearly seen although Adolphs includes the annotation step as well which I discuss separately below. This chapter is not the place to launch into detailed descriptions about the many issues and ongoing discussions related to the design and representativeness of corpora since that will be dealt with elsewhere (see chapter one). In this section, I will focus on general methods more than specific software tools that will very quickly go out of date, but inevitably this part of the survey will reflect the state of the art at the time of writing. In the corpus compilation stage, there are few tools and methods aimed specifically at the corpus linguist. Processes such as transcription, scanning, OCR, encoding and documenting tend to be supported by software that is used in many other document handling arenas. Conversely, for corpora of web, online or other computer-mediated communication (CMC) language varieties there is more software support. The new paradigm of web-as-corpus has only recently started to be presented in the text books in the field e.g. Cheng (2012: 36).

Creating a machine-readable corpus can be a very costly and time consuming exercise. The accuracy of any transcription and scanning is a primary consideration. In the next few paragraphs I will focus in turn on spoken, written and web-based language sampling and examine compilation issues specific to each type.

For spoken corpora, hardware and software recorders can be used for data collection. It is clearly important to obtain as high quality recording as possible and digital recorders are available quite cheaply. Next, transcription editing software is used to create a word level transcript alongside the audio data. Systems such as Voicewalker was used for the Santa Barbara corpus and SoundScriber was used for compiling MICASE. Praat can be employed for phonetic analysis. Unfortunately, speech recognition software is not yet accurate enough to automatically create text from sound recordings unless they are of broadcast quality. Even then, significant manual checking is required to prepare the high-quality, error-free transcriptions required for linguistic analysis. Some online sources of spoken data from broadcasters do include subtitles that may be extracted. Spoken corpora are often multimodal,

incorporating a video stream as well e.g. SCOTS and SACODEYL, so this entails the recording, editing and processing of video data. Ideally the transcription that is produced by these different methods would be aligned with the audio and video streams using software such as EXMARaLDA and the NITE XML Toolkit.

The considerations for written corpora are quite different. If the source material is available in hardcopy form e.g. a printed book or magazine, then a scanner is required in order to turn the printed version into a digital image and then OCR software creates a machine-readable version of the text contained in the image. A significant investment of time may be needed to manually check the OCR output and correct mistakes made by the software. Where the printed material is not of good clarity or the image has degraded over time, perhaps from a large newspaper sheet printed from a microfilm archive or photocopied from an original source, then OCR software may struggle to correctly block out multiple columns and recognise characters. In these cases, it may be better to resort to conversion by keyboarding of the original. This also applies to historical material or where the original is handwritten e.g. children's school projects or diaries. The approach taken by the Early English Books Online (EEBO) Text Creation Partnership (TCP) is to have an original book manually keyboarded by two different individuals. Then these two versions are compared and a third editor manually intervenes when differences arise. Such processes are vital in order to ensure that the machine-readable text is as accurate as possible. Depending on the type of the corpus and the age of the sources, it may be possible to find corpus material in electronic form already and then the keyboarding or scanning stages can be avoided. Out of copyright texts are more readily available, otherwise publishers need to be contacted to secure access and obtain copies of the data. Most corpus tools require plain text versions with optional XML encoding, so where material is sourced in another form, some format conversions will be in order. There are many tools available to assist in the conversion of Word, RTF and PDF file formats to TXT. These vary in quality and it is obviously important for later linguistic analysis to check that the original text flow has been preserved especially where the source has multiple columns or tabular formatting.

With the advent of the web and online data sources, it is easier to obtain electronic copies of written material. For example, the BE06 corpus (Baker, 2009) contained published British English written material that was sourced from the web to match the sampling model of the LOB/Brown family. Sites such as Project Gutenburg[7] contain large quantities of out of copyright or self-published books. In addition, text is made available online in such plentiful and continually growing quantities that linguists have started using the web as a source of corpus data directly resulting in the so-called web-as-corpus (WaC) paradigm. In contrast to the other areas of corpus compilation described above, there are a number of computational tools and methods to support the use of WaC data that are aimed at the corpus linguistic community. A typical WaC study involves a number of steps. First, you require a web crawler or downloader to collect the raw data, followed by language identification and filtering tools, then tools to aggregate multiple pages and

[7] http://www.gutenberg.org/

clean them of superfluous or boilerplate material such as navigation bars and adverts. It may also be necessary to detect duplicate pages or content and determine domain and genre information in order to select the most appropriate material for the corpus. Fortunately, most of these processes have been combined into simple tools such as BootCat (Baroni and Bernardini, 2004), WebBootCat (Baroni et al., 2006) and the WebGetter utility in WordSmith Tools[8]. Other types of online or CMC data can also be collected and cleaned in similar ways e.g. Usenet newsgroups (Hoffmann, 2007) although specific collections will require new tools to be developed.

Whether the corpus contains written, spoken or online varieties, computational tools and methods for record keeping, cataloguing and documenting the results are largely general purpose. Tools such as spreadsheets, databases and word processors are usually sufficient here although the relevant information may be stored alongside the corpus data itself for later retrieval in headers encoded within the files. In this case, XML editing software may be required to simplify the process and check for consistency of the results.

For further reading in the area of corpus compilation, Meyer (2002: 55-80) describes in detail the process of collecting and computerising data, although some of the technical details have changed in the ensuing decade. Good summaries of the basic design, practical collection and mark-up issues in compiling written and spoken corpora are covered in three chapters from the same handbook: Reppen (2010), Nelson (2010) and Adolphs and Knight (2010). The collection edited by Wynne (2005) covers issues of character encoding, archiving, distribution and preservation that are out of scope for this survey. Finally, legal and ethical issues of corpus construction and use are described in more detail in McEnery and Hardie (2012: 57-70).

2.2 Annotation

After a corpus has been collected, compiled and marked-up as described in the previous section, the second stage of the typical corpus linguistics methodology is to annotate the data. This can take many forms depending on the linguistic features that are to be investigated: morphological, lexical, syntax, semantic, pragmatic, stylistic or discoursal. Annotation can also be applied using manual (human-led) and/or automatic (machine-led) methods. Adding annotation allows the researcher to encode linguistic information present in the corpus for later retrieval or extraction using tools described in the next section. If the text is annotated or corrected by hand then this could form the basis of a training corpus for an automatic tagging system which can then learn from the human annotators in order to attempt to replicate their coding later on larger amounts of data. Part of the manually annotated dataset could also be used as a gold standard corpus against which to compare the output of the automatic tagging system in order to evaluate its accuracy. Computational methods and tools for corpus annotation therefore take two forms. First, intelligent editors to support manual annotation and second, automatic taggers which apply a particular type of analysis to language data.

[8] http://www.lexically.net/wordsmith/

Focussing on the first kind for a moment, it would be possible of course to manually annotate texts using any standard word processor, but here it is useful to have software tools that check annotation as it is added e.g. to ensure that typos in tags or category labels do not occur, and to allow standard mark-up formats (such as XML) to be employed consistently and correctly in the resulting corpus, e.g. as in the Dexter software.[9] In addition, such editors may be coupled with a central database or web server to allow teams of researchers to collaborate on a large-scale manual corpus annotation effort, e.g. the eMargin software.[10] Intelligent editors and manual tagging tend to be used for levels of linguistic annotation that are currently not feasible through automatic means, usually at the discourse level. A first example of this would be coreference annotation in which relationships between pronouns and noun phrases are marked up allowing the cohesion in a text to be studied. The Xanadu editor (Garside, 1993) was created for the manual mark-up of anaphor/cataphor and antecedent/postcedent relationships and other related features. Further examples of annotation that are carried out manually are pragmatic (speech or dialogue act) and stylistic (speech, thought and writing presentation) annotation although these do not tend to be directly supported by tailor-made software tools. Tagging of errors in learner corpora also proceeds at multiple linguistic levels and is generally carried out by hand although some automatic computational tools are now beginning to assist in this research for spelling and grammar level errors.

Now, turning to the automatic taggers which apply annotation without human intervention, many such systems exist and it is only possible to scratch the surface in a short survey. Annotation can be carried out automatically and with high levels of accuracy at the level of morphology (prefix and suffix), lexical (part-of-speech and lemma), syntax (parsing) and semantics (semantic field and word sense). For annotation in English and other major world languages, many of these tools are well developed and mature, but for other languages where corpus resources are scarce, basic language resource kits (BLARKs) are now becoming available. The commonest form of corpus annotation is part-of-speech (POS) tagging where a label (tag) is assigned to each word in the text representing its major word class and further morpho-syntactic information. POS tagging is essential for the study of grammatical change in language but also forms the basis of other levels such as parsing and semantic annotation as well as collocation analysis. In POS tagging as in other types of automatic corpus annotation, different computational methodologies have emerged with varying degrees of success. Rule-based methods rely on large manually constructed knowledge-bases encoding linguistic information such as the possible POS tags that a word or suffix may take and templates giving contexts where specific POS tags are ruled in or out. Statistical approaches draw their information from large corpora and use probabilities to calculate which POS tag is most likely in a given context. The most successful taggers employ a combination of the two kinds to provide robust results across multiple types of text e.g. CLAWS[11].

[9] http://www.dextercoder.org/
[10] http://emargin.bcu.ac.uk/
[11] http://ucrel.lancs.ac.uk/claws/

For further information about the corpus annotation process and computational tools and methods that support it, the reader is referred to Garside et al (1997), Mitkov (2003) and McEnery et al (2006: 29-45).

2.3 Retrieval

Once a corpus has been compiled and annotated using the methods and tools described in the previous two sub-sections, it is ready for the third stage i.e. retrieval. Retrieval methods and tools are those most commonly and prototypically associated with the corpus user's toolbox because many linguists use pre-existing corpora and so can skip the first two stages. Central amongst these methods is the concordance which displays all examples of a particular linguistic feature retrieved from the corpus and displayed in context, usually presented as one example per line, with a short section of surrounding text to the left and right of the example itself as shown in figure 1.

o that banks pay for the financial	support	they have received , until such tim
ehave responsibly . And we need to	support	and develop new ways of financing g
sk or offers rewards for failure .	Support	the establishment of Local Enterpri
their own part of the country and	support	the development of new products fro
a Liberal Democrat government will	support	them in doing so . Liberal Democrat
re existing RDAs have strong local	support	, they may continue with refocused
be covered in company reporting .	Support	public investment in the roll-out o
ing up opportunities and providing	support	and help when needed . a fair deal
ing up opportunities and providing	support	and help when needed . Despite incr
ovide for after-school and holiday	support	. This will allow an average primar
sed Graduate Teacher Programme and	support	the expansion of Teach First to att
e medium-sized firms that need the	support	. The money saved will Prime Minist
dent commission , with cross-party	support	, to develop proposals for long-ter
directly to prevention measures .	support	a ban on below-cost selling , and a
ent has failed to provide adequate	support	for those affected by the contamina
Paralympic Games in 2012 , and we	support	bids for other high-profile events
ic funding and dominant position .	Support	a diverse regional and local media
elieve every family should get the	support	it needs to thrive . a fair deal fo
elieve every family should get the	support	it needs to thrive , from help with

Figure 1: Concordance example for the word 'support'

All concordance tools provide for searching by a simple word and some tools permit searching for suffixes, multiple word phrases, regular expressions, part-of-speech tags, other annotation embedded within the corpus, or more complex contextual patterns. The idea of such a concordance arrangement predates the computer by quite a significant margin and scholars have in the past created concordances by hand for significant texts such as the Qur'an and the Bible. For example, Cowden-Clarke (1881) took 16 years to manually produce a complete concordance of all words (apart from a small set of words considered insignificant and occurring frequently such as *be*, *do* and *have*) in Shakespeare's writings. The concordance arrangement with the search item aligned centrally in the middle of each line provides the main window on to the underlying text for a corpus linguist.

Alongside the concordance method, a further four methods have emerged as central to the work of the corpus user: frequency lists, keywords, n-grams and collocations. Frequency lists, usually of words, provide a list of all the items in the corpus and a count of how often they occur and in some cases how widely dispersed the items are across multiple sections of a corpus. Again, this is something the computer is really good at doing efficiently and accurately. Different software tools do however produce slightly different frequency information and word counts for the same corpus data due to the way words are defined and delimited e.g. whether punctuation is

counted, capitalisation is significant and contractions are counted as one item or two. The keywords approach is a method to compare two word frequency lists using statistical metrics in order to highlight interesting items whose frequency differs significantly between one corpus that is being analysed and a much larger reference corpus. The keywords method can also be extended by comparing three or more word frequency lists representing distributions in a larger number of corpora. The keyness metric (usually Chi-squared or Log-Likelihood) provides complementary information to word frequency alone and gives an indication of the aboutness of a text, or what items are worthy of further investigation. The next method in the expanding corpus toolbox is usually referred to in the computational linguistics community as n-grams. In the corpus linguistics field, it is also known as lexical bundles, recurrent combinations or clusters. This method is fairly simple minded, is easy for the computer to calculate and represents the ability to count repeated phrases or continuous word sequences that occur in corpus data. N-grams of different lengths are counted separately i.e. repeated sequences of pairs of words are counted as 2-grams, three word sequences as 3-grams and so on. These lists can be seen as extensions of the simple word frequency list which is identical to a 1-gram list. An important variant of the n-gram approach is referred to as concgrams since they are derived from concordances and n-grams. Concgrams are repeated sequences of words that may be discontinuous and in any order, and this allows the user to find possibly interesting phraseological patterns in text which contain optional intervening items. In addition, the keyness method and the n-gram method can be combined in order to highlight key clusters i.e. repeated sequences whose frequency differs significantly in one corpus compared to a reference corpus. I will return to consider n-grams in more detail in section 3.

The final method in the corpus toolbox is collocation. In Firthian terms, collocation refers to the relationship between a word and its surrounding context where frequent co-occurrence with other words or structures help to define the meaning of the word. In practical terms, collocation as a method refers to the counting of the co-occurrence of two words in a corpus depending on their relative proximity to one another, and usually includes the calculation of a statistic or metric to assign significance values to the amount or type of co-occurrence relationships. Unlike the previous four methods where some minor operational differences that exist in tokenisation for frequency lists, concordances, keywords and n-grams could produce slightly different results in different tools, the collocation method itself is less tightly defined. Results can vary greatly depending on the parameters and metrics chosen. Many different statistics can be selected to determine the significance of the difference in the frequency of a word that occurs in close proximity to the node word against its frequency in the remainder of the corpus e.g. simple frequency, Mutual Information, Log-likelihood, Z-score, T-score, MI2 or MI3. Altering the span of the window around the node word where possible collocate words are considered can also significantly affect the results. Further typical options include whether to consider punctuation as a boundary to collocation window spans or impose minimum frequencies on collocates or node words.

The five methods described above have all been defined in relation to words contained in a corpus. They can equally well apply to tags within a

corpus, if any levels of annotation have been applied. For instance, a concordance can be produced for a certain part-of-speech tag, a frequency list of lemmas, key semantic tags, and calculate collocation statistics for which semantic tags relate to a given word.

The discussion so far in this subsection has been deliberately focused on general methods rather than specific software tools, but it is useful to include a brief timeline describing the development of retrieval tools in order to put them into context alongside the methods. The historical timeline of corpus retrieval software can be divided into four generations. In the first generation that developed alongside machine-readable corpora, software tools running on large mainframe computers simply provided concordance or key-word-in-context (KWIC) displays and separate tools were created in order to prepare frequency lists e.g. as used by Hofland and Johansson (1982). These tools were usually tied to a specific corpus. In the second generation, applications such as the Longman Mini-Concordancer, Micro-Concord, Wordcruncher and OCP were developed to run on desktop computers and were capable of dealing with multiple corpora and extra features to sort concordance lines by left and right context were added. Increased processing capabilities of PCs and laptops in the 1990s led to the third generation of retrieval software with systems such as WordSmith, MonoConc, AntConc and Xaira being developed. They were able to deal with corpora of the order of 10s of millions of words, containing languages other than English and they included implementations of the other methods outlined above in one package rather than as separate tools. The fourth generation of corpus retrieval software has moved to web-based interfaces. This allows developers to exploit much more powerful server machines and database indexes, provide a user-friendly web interface and host corpora that cannot otherwise be distributed for copyright reasons. Most of the web-based interfaces only permit access to pre-existing corpora rather than texts that the users collect themselves. For example, Mark Davies' corpus.byu.edu interface permits access to very large corpora: 450 million words of Contemporary American English (COCA), 400 million words of Historical American English (COHA), 100 million words of the TIME Magazine, and 100 million words of the British National Corpus. Other systems tend to rely on the Corpus Query Processor (CQP) server (part of the Open Corpus Workbench) or Manatee server. Their web-facing front ends are well known as BNCweb (providing access to the British National Corpus), SketchEngine (aimed at lexicographers) and CQPweb (based on the BNCweb design but suitable for use with other corpora). Other web-based tools in a similar mold are Intellitext (aimed at humanities scholars), Netspeak and ANNIS. The web-based Wmatrix software (Rayson, 2008) allows the user to perform retrieval operations but it also annotates uploaded English texts with two levels of corpus annotation: part-of-speech and semantic field. For further information on the four generations of corpus retrieval tools, a good survey can be found in McEnery and Hardie (2012: 37-48).

2.4 Critical reflection

Although I have presented corpus software as distinct tools used for the three stages of compilation, annotation and retrieval, it needs to be highlighted that the separation between these stages is not always clear cut. As mentioned,

Wmatrix performs both automatic annotation and retrieval. Other tools, such as WordSmith and BNCweb permit the user to manually categorise concordance lines and this can be viewed as a form of corpus annotation. It should therefore be clear that a specific piece of corpus software cannot always be pigeonholed into one of these three categories.

Looking back on the brief survey in the preceding three sub-sections, it can be seen that a wide range of computational methods and tools are available to the corpus linguist. Updated versions of corpus software are being delivered on a regular basis, however the corpus toolkit is in need of a methodological overhaul on a number of fronts. Words of caution have been expressed over the use of the keywords technique (Kilgarriff, 1996; Baker, 2004; Rayson, 2008; Culpeper, 2009) related to the choice of reference corpus, the statistic used in the calculation, the sometimes overwhelming number of significant results, the use of range and dispersion measures, and the focus on lexical differences rather than similarities. As discussed in the next section, n-gram results as currently presented can be large in number and difficult to analyse. Concordance software does not fully support linguists' requirements for the manual categorisation of concordance lines (Smith et al, 2008). The use of different software and methods sometimes produces different analyses (Baker, 2011). And finally, the methodology for the study of large-scale diachronic datasets is just beginning (Hilpert and Gries, 2009) and lacks good tool support.

One notable issue is the goodness of fit of current annotation and retrieval software where the corpus data is non-standard or 'noisy'. Vast quantities of historical data are now being digitised, and billions of words are available on the web or in online social networks. In both these cases, automatic tagging tools have been shown to be less accurate and robust. In particular, spelling variation causes problems for POS tagging, concordancing, keywords, n-grams and collocation techniques. Even simple frequency counting is more difficult for the computer since multiple spelling variants will disperse the counts across different surface forms. Fortunately, tools such as VARD[12] have been developed in order to counter this problem by pre-processing the corpus data and linking standard forms to spelling variants.

As described in the introduction, corpus linguistics matured following hardware and software developments in computers and text processing methods. These developments have enabled much larger corpora to be collected and analysed. However, it could be argued that corpus linguistics is now very tool-driven (Rayson and Archer, 2008), in other words we are "counting only what is easy to count" (Stubbs and Gerbig, 1993: 78) rather than what we would like to count. There are some areas of linguistic study where automatic annotation tools are not yet accurate enough or not available at all, and so studies have to proceed with manual corpus annotation e.g. at the discourse, stylistic or pragmatic levels with very little computational tool support. Finally, as has been seen, corpus retrieval software in general does not permit the concordancing of audio and video material apart from in some notable cases, e.g. the SCOTS corpus.

[12] http://ucrel.lancs.ac.uk/vard/

3. Empirical study

Following on from the critical reflection about the limitations of computational methods and tools in corpus linguistics, this section will zoom in on one of the standard methods and illustrate some of the potential problems with it for corpus linguists and some possible solutions. As described in the previous section, the computational n-grams method appears under various guises in corpus linguistics. Biber et al (1999) name the outputs of the n-gram method as *lexical bundles*, Altenberg (1998) refers to them as *recurrent combinations* and Scott calls them *clusters* in WordSmith Tools[13]. All these names refer to the results of the same procedure which counts continuous multiword sequences and produces frequency lists very much like a word frequency list. The simple word frequency list consists of n-grams of length one, but n-grams of length 2, 3, 4, and 5 are usually counted. Longer sequences occur in larger corpora but are less frequent than their shorter counterparts.

Let us now consider some example lists in a short empirical study to consider the usefulness of the n-gram method itself. The n-gram procedure was applied to the full text of “Alice’s Adventures in Wonderland” (one of the most frequently downloaded texts from the Internet Archive and Project Gutenburg[14]) using Ted Pedersen’s N-gram Statistics Package (NSP)[15]. The text is only 26,400 words long but it produces 1810 2-grams, 737 3-grams, 192 4-grams and 51 5-grams that occur three times or more. This illustrates the first problem with the n-gram method, since even with a small text such as this, a large number of results is generated. Table 2 shows the top 10 n-grams of length between 2 to 5, and their frequencies.

[13] http://www.lexically.net/wordsmith/
[14] http://www.archive.org/details/alicesadventures00011gut
[15] http://www.d.umn.edu/~tpederse/nsp.html

	2-gram	Freq.	3-gram	Freq.
1	said the	210	the mock turtle	53
2	of the	133	the march hare	30
3	said alice	116	i don t	30
4	in a	97	said the king	29
5	and the	82	the white rabbit	21
6	in the	80	said the hatter	21
7	it was	76	said to herself	19
8	the queen	72	said the mock	19
9	to the	69	said the caterpillar	18
10	the king	62	she went on	17
	4-gram	**Freq.**	**5-gram**	**Freq.**
1	said the mock turtle	19	will you won t you	8
2	she said to herself	16	won t you will you	6
3	a minute or two	11	the moral of that is	6
4	you won t you	10	you won t you will	6
5	said the march hare	8	as well as she could	6
6	will you won t	8	and the moral of that	5
7	said alice in a	7	as she said this she	5
8	i don t know	7	the dance will you won	4
9	well as she could	6	you will you won t	4
10	in a great hurry	6	dance will you won t	4

Table 2: Top 10 n-grams from Alice's Adventures in Wonderland"

Unsurprisingly, the top 2-grams often are dominated by high frequency words such as 'the', 'of', 'in' and 'it'. In the 3-gram list there are potentially more useful entries, depending on the research question in mind, which contain information about the main characters of the story. Higher order clusters may be more useful for analysis as they correspond to longer phrasal or clausal-like fragments and help to disambiguate and contextualise some frequent words. The top frequencies of 3-grams and 4-grams are much lower and a total of only 51 5-grams are reported with a frequency of three or more. In terms of practicalities for analysis and categorising these items, it would be useful to look further into concordances but that is beyond the scope of this small case study here. Especially at the 2-gram level there are too many patterns (1810) to analyse by hand, so some further filtering would be required. These lists are already reduced to patterns occurring three or more times, but the dispersion or range information across chapters might also be considered in order to remove n-grams which only occur in one or two chapters. When the n-gram method is applied to larger texts, significantly many more results are produced and the practical analysis problems are only exacerbated, so much more stringent filtering will be required. For example, Biber et al (1999: 992) used a frequency cut-off of 10 occurrences per million words in a register and the occurrences must be spread across at least five different texts in the register. A second option is to use the keyness calculation in a similar way to how it is applied to a word frequency list. A p-value or Log-Likelihood value cut-off or sorting of results could then be applied in order to filter or rank key clusters in terms of keyness. For example,

Mahlberg (2007) examines key clusters in a Dickens corpus compared against a similar sized corpus of other nineteenth century authors and this allows her to home in on more interesting n-grams more quickly.

Despite the practical issues reported here, n-grams in the form of lexical bundles have been used very successfully to differentiate registers by comparing their frequency of occurrence, lexico-grammatical type and typical discourse function across different texts. Biber (2009) puts the n-gram method in the corpus-driven category in order to distinguish it from other approaches which put more emphasis on the structures informed by linguistic theory and phraseology such as formulaic and idiomatic expressions. The obvious conclusion from looking at n-gram lists such as those in table 2 is that there are few meaningful, structurally complete or idiomatic phrases that have been extracted by this method. Some manual work needs to be done in order to match the n-grams to grammatical or clausal elements or fragments of elements. In practice, a common approach is to combine the automatic simple minded corpus-driven techniques with the linguistically informed corpus-based methods. Hence, collocation statistics and collocational frameworks can be combined, possibly with the addition of POS templates in order to filter out n-gram patterns that are not of interest. A further extension as mentioned briefly before, is to use concgrams to allow more flexibility in the order and placement of unspecified elements of the phrases or chunks (O'Donnell et al, 2012). The kfNgram software tool also allows discovery of phrase-frames which are groups of n-grams that are identical except for a single word.[16]

Returning to the example text and the results above, it can be observed that there is a further cause of redundancy in the n-gram lists. This redundancy emerges if the patterns from multiple n-gram lists are compared. For example, the most frequent 2-gram "said the" also appears in four of the top 10 3-grams "said the king", "said the hatter", "said the mock" and "said the caterpillar" and two of the top 10 4-grams "said the mock turtle" and "said the march hare". The 3-gram "said the mock" overlaps with the 4-gram "said the mock turtle". Further overlaps can be seen for "in a", "the king", "and the", "the march hare" and others. The practical upshot of this recursion would be a duplication of effort considering and investigating these patterns in turn, or worse still, missing the occurrence of shorter or longer overlapping recurrent sequences if a study is limited to one length of n-grams as quite often happens for reasons of time. In addition, partial overlaps can be seen between the end of one pattern "dance will you won t" and another "you won t you will". Given the small size of this text such things are relatively easy to spot, but some automatic tool support is required to facilitate this process for larger corpora. One approach adopted in the Wmatrix software (Rayson, 2008) is to provide c-grams or 'collapsed grams' which combines these overlaps and subsequences into one tree view. An example of this can be seen in table 3 for the 2-gram "and the".

and the	82
and the queen	5
and the moral	5
and the moral of	5

[16] http://www.kwicfinder.com/kfNgram/kfNgramHelp.html

and the moral of that	5
and the little	4
king and the	4
the king and the	4
and the other	4
and the words	3
duchess and the	3
the duchess and the	3
and the gryphon	3
gryphon and the	3
the gryphon and the	3
and the baby	3

Table 3: C-gram tree view for “and the”

This tree view shows longer n-grams indented and arranged underneath shorter n-grams that they contain. With software support, this tree view can be expanded or collapsed in order to focus on the important details. When linking to concordance views, if the user clicks on “and the” they can choose whether to include or exclude longer patterns identified elsewhere. Such an approach significantly reduces the manual labour required to analyse n-grams.

Similar motivations lay behind the proposal for an adjusted frequency list (O’Donnell, 2011) where clusters and single words appear alongside each other in a single cluster-sensitive frequency list. Rather than use frequency or collocation statistics to rank or filter n-grams, Gries and Mukherjee (2010) recommend the use of the measure of lexical gravity which employs type (rather than token) frequencies. Their method also considers n-grams for multiple ‘n’ at the same time rather than separate lists, i.e. rather than fixing the length in advance, they use a collocational measure to determine the most appropriate length for the n-grams. It is now abundantly clear from this short empirical excursion into n-grams that there is much still to learn about how this computational method can be used for corpus linguistics and that accompanying software tools need further development to enable us to find and filter results more accurately and efficiently.

4. Conclusion

This chapter has very briefly surveyed the three stages in the corpus research process: compilation, annotation and retrieval. To survey this whole area in one chapter is an almost impossible task and pointers to further book level treatments of each of these areas were provided. It would also have been possible to widen out this survey of computational tools and methods to include very similar approaches undertaken elsewhere. There are at least three groups of tools and their related disciplines which are relevant. First, tools which provide Computer Assisted Qualitative Data Analysis (CAQDAS), such as ATLAS.ti, NVivo, QDA Miner and Wordstat incorporate some very similar methods to those described here but are not widely used in corpus linguistics. Their application tends to be in areas other than linguistics research but where language, texts or documents are key sources, e.g. for political text analysis or other social science research questions. Second, tools such as Linguistic Inquiry and Word Count (LIWC) are used in

psychology for counting emotional and cognitive words and other psychometric properties of language. Third, another similar set of tools is employed in the field of Digital Humanities for text mining of language properties in order to answer traditional humanities research questions and the formation of new research questions that are more difficult to answer with small scale manual text analysis. Software tools such as Voyant and MONK are designed to allow large quantities of text to be searched, analysed and visualised alongside other tools such as Geographical Information Systems (GIS) and Social Network Analysis (SNA). However here the focus has been on the tools and methods used in the field of (English) corpus linguistics. I uncovered some limitations of the current crop of computational tools and methods and reflected on whether corpus linguistics could be said to be becoming tool-driven.

Methods and tools for corpus linguistics have developed in tandem with the increasing power of computers and so it is to the computational side I look in order to take a peek into the future of corpus software. In order to deal with the increasing scale of mega corpora derived from the web or historical archives, significantly more processing power is needed. Cloud computing may offer a solution here where (possibly multiple) virtual machines are used to run software tasks in parallel thereby making the results quicker to retrieve. For example, the GATE system (General Architecture for Text Engineering) now runs in the cloud, and on a smaller scale, so do Wmatrix and CQPweb. In order to analyse and automatically tag a 2-billion-word Hansard dataset consisting of data from 200 years of the UK parliament[17], we recently estimated that it would take 41 weeks of computer time. However, using a High Performance Cluster (multiple connected computers running small batches of text) at Lancaster, we were able to complete the task in three days.[18] A similar approach was taken by Oldham et al. (2012) to apply corpus methods to a vast collection of scientific articles in synthetic biology. Other computational projects offer a wider distributed approach where tools and corpora are connected across Europe in a large research infrastructure e.g. CLARIN[19] and DARIAH[20].

The sheer scale of corpora and the consequential numbers of results obtained from keyness and n-gram analyses are becoming increasingly hard to analyse by hand in a sensible time-scale. This suggests that better visualisation techniques would be beneficial in order to home in on interesting results, and simple histograms or bar charts to utilise frequency, range or dispersion data and display large collocation networks are no longer sufficient. Another visualisation approach is to use a Geographic Information System (GIS) to locate corpus results on a map and this may enable the analysis of texts in literature, history and the humanities via computational techniques to extract place names.

---

[17] http://www.gla.ac.uk/schools/critical/research/fundedresearchprojects/parliamentarydiscourse/

[18] This work was implemented by Stephen Wattam, currently a PhD student at Lancaster University

[19] http://www.clarin.eu/

[20] http://www.dariah.eu/

Finally, gazing into a crystal ball, it is possible to see corpus linguistics techniques spreading not just to other areas within linguistics (e.g. stylistics) but also to other disciplines: e.g. psychology, history and the social sciences in general where large quantities of text are used in research.[21] Following the four generations of corpus retrieval software discussed in section 2.3, this development will form the fifth generation where the specific disciplinary needs of the end-user will need to be taken into account and greater interoperability between different software tools will be vital to facilitate the research.

References

Adolphs, S. and Knight, D. 2010. Building a spoken corpus: what are the basics? In A. O'Keeffe and M. McCarthy (eds.) *The Routledge Handbook of Corpus Linguistics*. Routledge, London, pp. 38-52.

Altenberg, B. 1998. On the phraseology of spoken English: The evidence of recurrent word combinations. In A. Cowie (Ed.), *Phraseology: Theory, Analysis and Applications*. Oxford: Oxford University Press, 101–122.

Baker, P. 2004. Querying keywords: questions of difference, frequency and sense in keywords analysis. *Journal of English Linguistics*, 32:4, 346–359.

Baker, P. 2009. The BE06 Corpus of British English and recent language change. *International Journal of Corpus Linguistics*. 14:3, 312-337.

Baker, P. 2011. Discourse, news representations and corpus linguistics. *Presented at the Corpus Linguistics Conference 2011*, Birmingham, UK.

Baroni, M. and Bernardini, S. 2004. BootCaT: Bootstrapping corpora and terms from the web. *Proceedings of LREC 2004*.

Baroni, M., Kilgarriff, A., Pomikálek, J., Rychlý, P. 2006. WebBootCaT: a web tool for instant corpora. In *proceedings of Euralex 2006*, Torino, Italy.

Biber, D. 2009. A corpus-driven approach to formulaic language in English: multi-word patterns in speech and writing. *International Journal of Corpus Linguistics*, 14:3, 275-311.

Biber, D., Johansson, S., Leech, G., Conrad, S. & Finegan, E. 1999. *The Longman Grammar of Spoken and Written English*. London: Longman.

Cowden-Clarke, M. V. 1881. The complete concordance to Shakespeare: being a verbal index to all the passages in the dramatic works of the poet, new and revised edition. Bickers & Son, London.

[21] This is beginning to take place in initiatives such as CASS, the ESRC Centre for Corpus Approaches to Social Science (http://cass.lancs.ac.uk/)

Culpeper, J. 2009. Keyness: Words, parts-of-speech and semantic categories in the character-talk of Shakespeare's Romeo and Juliet. *International Journal of Corpus Linguistics*. 14:1, 29-59.

Garside, R. 1993. The marking of cohesive relationships: tools for the construction of a large bank of anaphoric data. *ICAME Journal* 17: 5-27.

Garside, R., Leech, G. and McEnery, T. (eds.) 1997. *Corpus annotation: Linguistic information from computer text corpora.* Harlow: Longman.

Gries, S. Th. 2009. Quantitative corpus linguistics with R: a practical introduction. Routledge, London.

Gries, S. Th. and Mukherjee, J. 2010. Lexical gravity across varieties of English: An ICE-based study of n-grams in Asian Englishes. *International Journal of Corpus Linguistics* 15:4, 520–548.

Hilpert, M. and Gries, S. Th. 2009. Assessing frequency changes in multistage diachronic corpora: Applications for historical corpus linguistics and the study of language acquisition. *Literary and Linguistic Computing*, 24:4, 385-401.

Hoffmann, S. 2007. Processing Internet-derived text — creating a corpus of Usenet messages. *Literary and Linguistic Computing*, 22:2, 151-165.

Hofland, K. and Johansson, S. 1982. *Word frequencies in British and American English*. Bergen: Norwegian Computing Centre for the Humanities.

Kilgarriff, A. 1996. Why chi-square doesn't work, and an improved LOB-Brown comparison. In *proceedings of the ALLC-ACH Conference*. Bergen, Norway.

McEnery, T., Xiao, R. and Tono, Y. 2006. *Corpus-based language studies: an advanced resource book.* London: Routledge.

McEnery, T. and Hardie, A. 2012. *Corpus linguistics: method, theory and practice*. Cambridge: Cambridge University Press.

Mahlberg, M. 2007. Clusters, key clusters and local textual functions in Dickens. *Corpora*, 2:1, 1–31.

Mason, O. 2000. *Programming for corpus linguistics: how to do text analysis with Java*. Edinburgh: Edinburgh University Press.

Meyer, C. 2002. *English corpus linguistics: an introduction.* Cambridge: Cambridge University Press.

Mitkov, R. (ed.) 2003. *Oxford Handbook of Computational Linguistics.* Oxford: Oxford University Press.

Nelson, M. 2010. Building a written corpus: what are the basics? In A. O'Keeffe and M. McCarthy (eds.) *The Routledge Handbook of Corpus Linguistics*. Routledge, London, pp. 53-65.

O'Donnell, M. B. 2011. The adjusted frequency list: a method to produce cluster-sensitive frequency lists. *ICAME Journal*, 35, 135-169.

O'Donnell, M.B., Scott, M., Mahlberg, M. and Hoey, M. 2012. Exploring text-initial words, clusters and concgrams in a newspaper corpus. *Corpus linguistics and linguistic theory*, 8:1, 73-101.

Oldham, P., Hall, S., and Burton, G. 2012. Synthetic Biology: Mapping the Scientific Landscape. *PLoS ONE* 7(4): e34368. doi:10.1371/journal.pone.0034368

Rayson, P. 2008. From key words to key semantic domains. *International Journal of Corpus Linguistics*. 13:4, 519-549.

Rayson, P. and Archer, D. 2008. Key domain analysis: mining text in the humanities and social sciences. *Workshop on Text mining and the social sciences, 4th International Conference on e-Social Science*, University of Manchester, UK.

Reppen, R. 2010. Building a corpus: what are the key considerations? In A. O'Keeffe and M. McCarthy (eds.) *The Routledge Handbook of Corpus Linguistics*. Routledge, London, pp. 31-37.

Smith, N., Hoffmann, S. and Rayson, P. 2008. Corpus Tools and Methods, Today and Tomorrow: Incorporating Linguists' Manual Annotations. *Literary and Linguistic Computing*, 23: 2, 163-180.

Stubbs, M. and Gerbig, A. 1993. Human and Inhuman Geography: On the Computer-Assisted Analysis of Long Texts. In: M. Hoey (ed.), *Data, Description, Discourse. Papers on the English Language in honour of John McH Sinclair on his sixtieth birthday*. Harper Collins, London, pp. 64-85.

Wynne, M (editor). 2005. *Developing Linguistic Corpora: a Guide to Good Practice*. Oxford: Oxbow Books. Downloaded from http://www.ahds.ac.uk/creating/guides/linguistic-corpora/index.htm