Searching and concordancing

Martin Wynne, University of Oxford
Pre-publication draft. To appear in *Handbook of Corpus Linguistics*, edited by Merja Kytö & Anke Lüdeling, Mouton de Gruyter, 2007.

Contents

1. Introduction

This chapter will deal with the basic techniques of linguistic analysis which involve searching for and finding things in a corpus, and displaying the results in useful ways. The most popular way to display the results of a search in a corpus is in the form of a concordance. In corpus linguistics, a simple concordance is a list of examples of a word as they occur in a corpus, presented so that the linguist can read them in the context in which they occur in the text.

One of the first uses of the computer for processing texts was the started in 1946 by Father Roberto Busa with the help of IBM to do searches in and generate concordances from the works of Saint Thomas Aquinas. Busa produced the Index Thomisticum, which is available online (Bernot and Alarcón 2005). This work can be seen both as a precursor for work in modern corpus linguistics, but also as the continuation of a long tradition of non-computational work in generating concordances from important texts. In the older tradition, a concordance is an alphabetical list of the principal words used in a book, or body of work, with their immediate text surrounding them.

For many linguists, searching and concordancing is what they mean by "doing corpus linguistics". The availability of an electronic corpus allows the linguist to use a computer to search quickly and efficiently through large amounts of

language data for examples of words and other linguistic items. When the results of these searches are displayed as a concordance, as in figure 1, the linguist can view the data in a convenient format and start to analyse it in various ways.

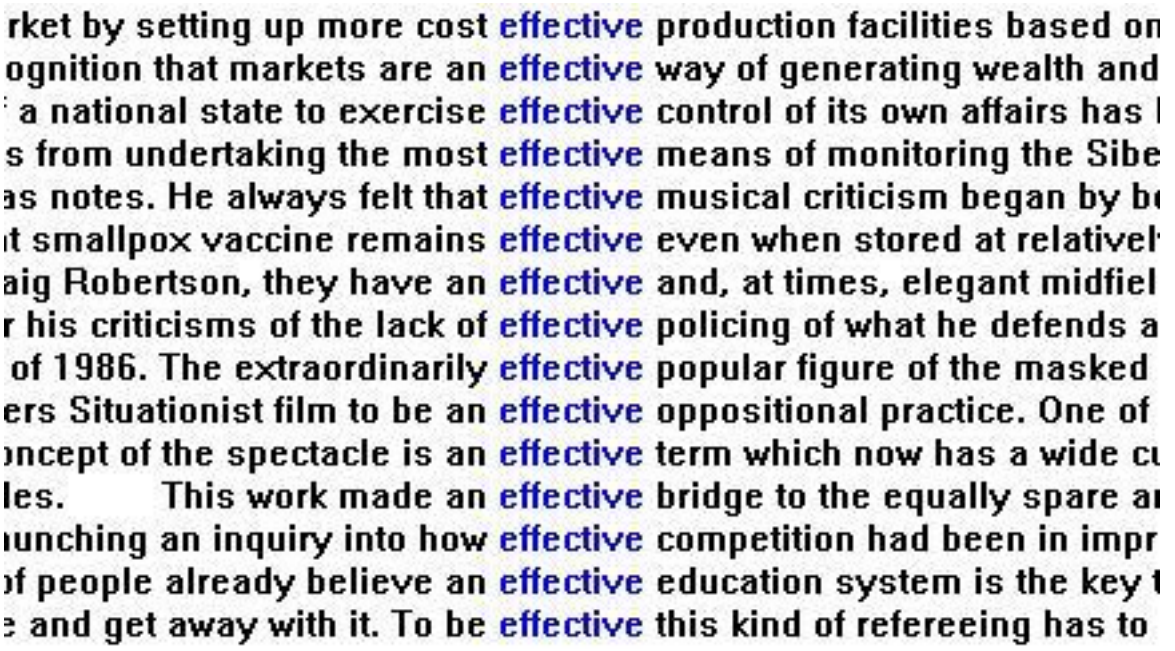


Figure 1: Concordance of 'effect' in the BNC-Baby corpus.

Searching and concordancing may only be the start of a linguistic investigation. They are ways to verify, identify or classify examples in a corpus, in order to start to develop a hypothesis or a research methodology. Searching and concordancing are important elements in the basic toolkit of techniques which the linguist uses. They are essential for checking results derived by automatic procedures and to examine examples in a text in more detail. These techniques will be examined in more depth in this chapter.

Tognini-Bonelli (2000) explores the theoretical basis for reading concordances. She draws attention to the differences between, on the one hand, a linguist reading a text in the usual linear fashion from beginning to end, and on the other hand, a linguist reading the lines of a concordance. When reading a concordance, the linguist is looking for patterns of similarity or contrast in the words surrounding the search term. In structuralist terms, when the linguist reads a text, they are reading parole, or the way meaning is created in this particular text, and when analysing a corpus, they can also gain insights into langue, or the way that the language system works (Saussure 1922/1983). In functionalist terms, reading texts allows the reader to concentrate on the poetic, emotive, rhetorical, referential and phatic functions, while concordancing a corpus can foreground the metalingual function of a text (Jakobson 1960).

When the linguist reads concordance lines, the focus of attention is usually on repeated patterns in the vertical direction, or paradigmatic plane. It is also necessary to be able to read each one horizontally, from left to right (in languages written this way), to interpret the meaning of the particular example. Reading a set of concordance lines vertically, from top to bottom of the screen, and sorting them in various ways, allows the linguist to see lexical, grammatical and textual paradigms. Simply searching through a corpus and looking at examples one by one is to treat the corpus like a text; it is through

concordancing that the patterns of usage and the paradigms are revealed.

Each of the sections below in this chapter will examine one the various functions which are available for searching a corpus and for generating and analysing concordances. This chapter will focus on searching and concordancing in a monolingual text corpus, and the examples given are from English. Some different functions may apply to other types of corpus and work on other languages, and are referred to briefly in section 16 below.

2. Searching for words, phrases and other patterns

2.1. Description

Corpus linguists will typically wish to find certain linguistic items, or sets of linguistic items, in a corpus. The item may be a word, a phrase or some other more complex entity.

When the user has access to a corpus in electronic form, it is possible to search in the corpus for patterns. At the simplest level, a search may display the first, or next, occurrence of a word in the corpus. More usefully, all the occurrences may be found and displayed for the user.

The process of searching for patterns in the corpus underpins all of the functions which are described in this chapter below. They are all either more elaborate ways of searching, or more elaborate ways of displaying the information extracted by a search routine. This section will deal only with the most simple and unelaborated methods of searching in a corpus.

2.2. Example

In figure 2 the results of loading a corpus file and searching for a word ('effect') are displayed. Note that only the next example is displayed.

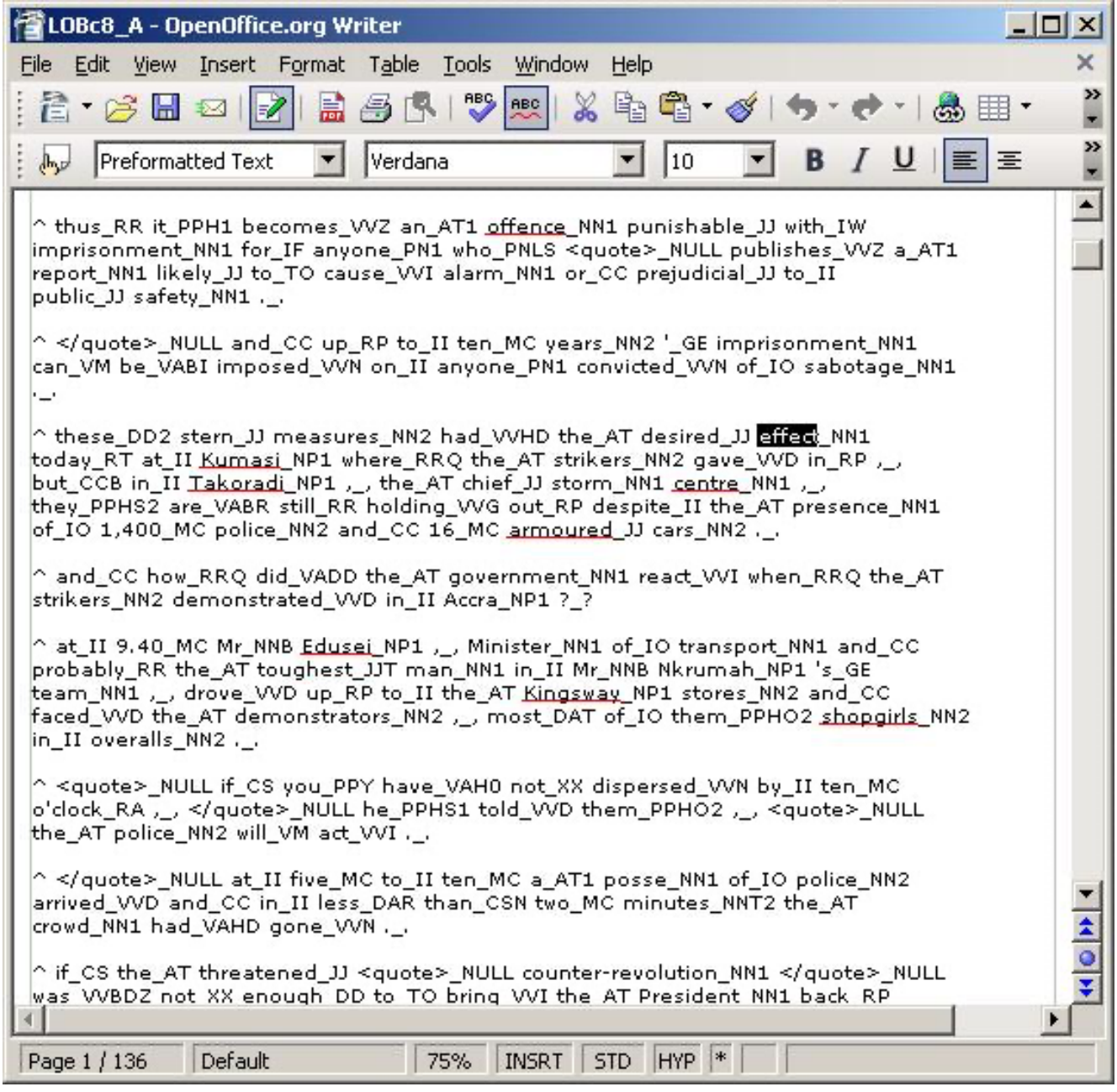


Figure 2: Searching for 'effect' in the LOB corpus with a word-processing program

2.3. Analysis with this function

Searching for and finding examples of linguistic items in a corpus are useful to find examples of known textual phenomena. Real examples are useful evidence for dictionaries, grammars, textbooks or lecture notes. With the use of a corpus, linguists can find examples which

were really used, rather than invented examples. And rather than simply using real examples found when reading a text, using a corpus makes it possible to quickly search across a large collection of texts and find examples which are more frequent, or more typical, or simply chosen more randomly.

Searching for occurrences of a word or phrase can also be used to test for their existence in a particular corpus. In this way, a hypothesis that a given feature does not occur can be disproved. However, the non-existence of a linguistic feature in a particular variety of language cannot be proved by its non-existence in a corpus, as the corpus is only a sample and the feature might occur in the much greater population of texts not present in the corpus. Indeed, Chomsky (1957) claimed that a corpus (meaning any collection of utterances) can only ever represent a trivial number of the infinite number of possible sentences in a language. Corpus linguists argue, however, that frequencies are interesting and important.

While simply searching for and reading examples can be useful, it is preferable to obtain a concordance because then, if results are found, the types of usage which are present can be examined together and compared.

A further use of simple searching, rather than concordancing, is to search the vast amounts of electronic text available on the web, when the texts can't

all be loaded into a concordance program on the user's computer. Using search engines, the user can search for words or phrases and find many examples of This can be particularly useful for using online text collections, examining new usages in languages which are not yet represented in available corpora, and for investigating emerging modes and styles of electronic communication. Systems of returning the results of online searches in the form of concordances are also under development (Renouf, forthcoming).

2.4. Technical requirements

Searching in corpora is a simple task which can be done by a variety of means including the use of scripts, programs, web browsers, word processors, or specialised corpus analysis applications, such as concordancers

It is possible for users to write their own scripts or programs to search a corpus, if they have the expertise and software tools available to them. However, searching with scripts and programs is unlikely to be useful for producing output in a form which is easy for the user to read, but can be quick and useful in order to generate examples for further processing, or simply to test for the existence of examples. In order to obtain a listing of only and all the required examples, in a form that is easy to use, dealing appropriately with markup, along with a

suitable amount of co-text, and in a form in which they easily be read, re-sorted and further analyse, requires intensive programming and refining of the search program. Concordance programs aim to do all of these jobs for the user and present the results of searching a corpus in a useful way.

A simpler way for linguists to start to search for an example of a string in a corpus is to use a text viewing or editing application, such as a web browser or a word processor. This can be done by loading the corpus text into the application, and then carrying out a simple 'string search' (often called the 'Find' command in these programs), where the user types in a word or phrase, and the program displays the section of the text where the next example of the word or phrase occurs. Many applications will allow more complex patterns to be entered as the search term, including wild cards, character ranges, and optional elements. Some applications will allow regular expressions, which are a powerful way of using a variety of special characters to obtain matches with a set of patterns.

The result of such a search will take the user to the next place in the text where the search term occurs. However, a corpus may be stored in numerous text files and maybe directories, and many applications will not search across more than one file. Loading a corpus into a text editor or word processing program may also be risky, because the

user can easily edit the text, perhaps inadvertently, and the program may to correct spelling, silently insert formatting tags and alter the file in other ways. The user should also be aware that loading a corpus into a web browser may lead to the browser trying to interpret the tags as HTML or as XML; if there are other types of markup in the text it may cause unpredictable and unwanted results in the way the text is displayed. The basic problem is that these programs are designed for reading or editing text documents, but not for searching text corpora.

Concordance programs aim to do all of these jobs for the user and present the results of searching a corpus in a useful way. These limitations of simple scripts and programs mean that most users find it better to develop or use specialist corpus analysis software for anything other than preliminary investigations, or 'quick and dirty' research.

3. Concordances

3.1. Description

A concordance is a listing of each occurrence of a word (or pattern) in a text or corpus, presented with the words surrounding it. A simple concordance of "Key Word In Context" (KWIC) is what is usually referred to when

people talk about concordances in corpus linguistics, and an example is shown in figure 3. Concordances are essentially a method of data visualisation. The search term and its co-text are arranged so that the textual environment can be assessed and patterns surrounding the search term can be identified visually. Michael Barlow (2004) defines concordances (and wordlists) as transformations of a text, giving the analyst the opportunity to view different perspectives on a text.

Often a concordance of a particular search term in a corpus will produce too many results for a linguist to read and analyse. In this case a reduced number of examples can be selected. Often around 40 examples (or a number providing up to two screenfuls in the working environment) is useful for providing the analyst with at least a preliminary view of the relevant patterns, although the number necessary to examine is heavily dependent on the structure of the corpus, the total number of examples and the type of investigation which is being carried out. If such a sample is chosen, it is usually important to select them either randomly from the total, or to select every nth example (e.g. every 20th example if there are around 8000 in total). Otherwise, the software is likely to provide by default the first 40 examples, which may all come from one file, and thus there would be a high risk of a highly biased sample, reflecting only the language usage of one text or variety. Selecting every nth examples is one method of thinning a concordance (see section 8 below).

3.2. Example

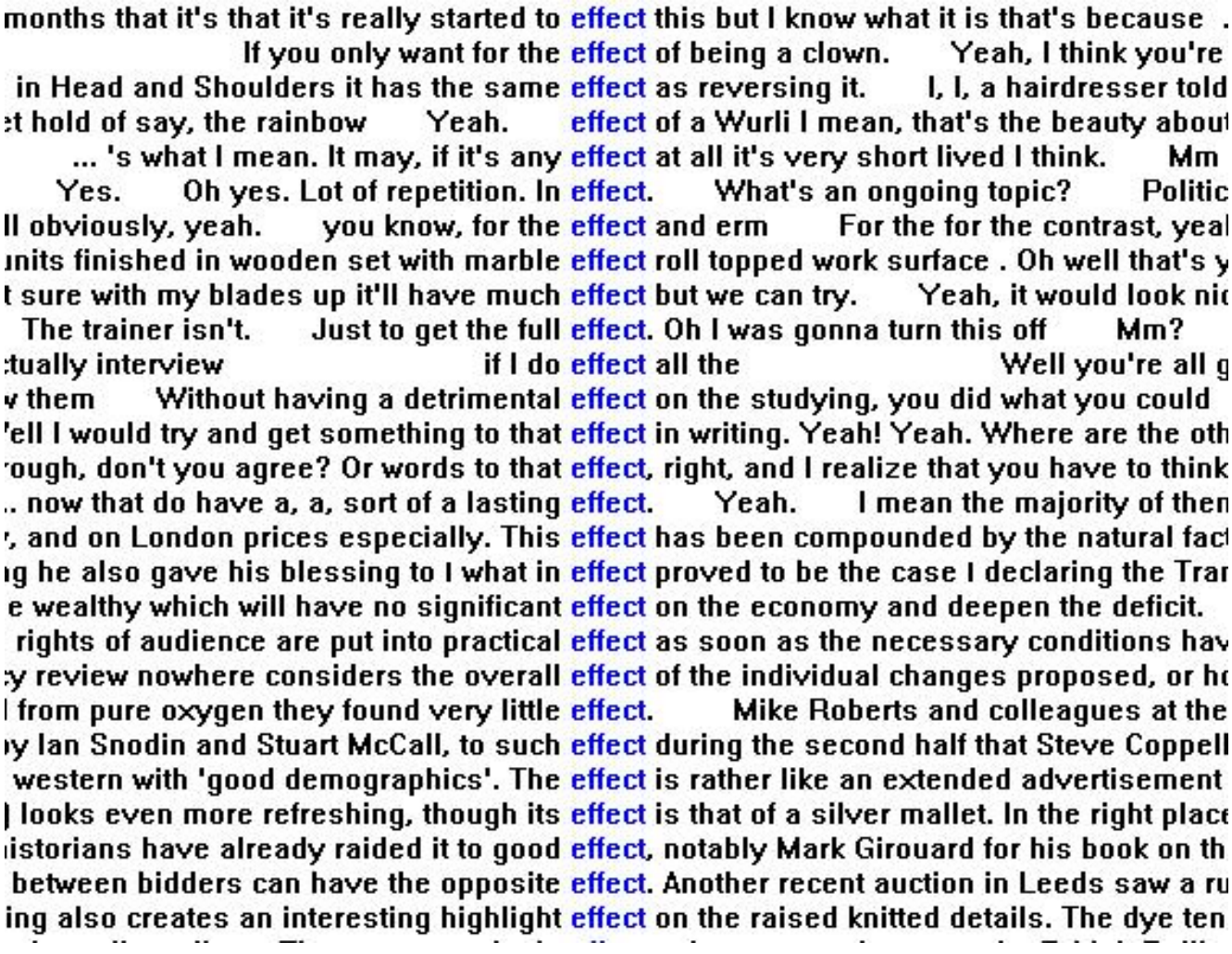


Figure 3: A KWIC concordance of the word 'effect' in the BNC-Baby corpus

3.3. Analysis with this function

The primary motivation for the use of concordance data in modern corpus linguistics is the belief that interesting insights into the structure and usage of a language can be obtained by looking at words in real texts and seeing what patterns of lexis, grammar and meaning surround them.

The use of concordances is essentially a manual task for human analysts, unlike the use of many computer algorithms to automatically extract information about the occurrence and co-occurrence of words in texts. Automatic extraction of wordlists, collocate lists, etc. can lead the analyst to deal only with words abstracted from the texts where they occur, and taken away from the place where meaning is created. Reading concordances means looking at words in their context of occurrence in texts, and allows the analyst to see what the meaning of the word is in the text, and to see how that meaning is created in the particular case.

Furthermore, reading concordances allows the user to examine what occurs in the corpus, to see how meaning is created in texts, how words co-occur and are combined in meaningful patterns, without any fixed preconceptions about what those units are. It can be a method of approaching the corpus in a theory-neutral way, and is what Tognini-Bonelli (2000) calls corpus-driven linguistics.

However, interesting results do not spring out as soon as the corpus is loaded into the software. To generate a concordance, the user must select what to search for, and this means approaching the corpus with some pre-conceptions about what words (or other features) will be interesting to look at. One way of avoiding this bias is to make use of a function which some programs have to

provide a complete concordance of a text or a corpus. This can be useful for a text, and was the traditional way of making concordances for the study of literary or religious works before the era of the computer. However, a complete concordance of a corpus will usually produce more data than human analysts can cope with. Even major lexicographic projects are likely to be selective with what words to search for and how many examples to look at in a large corpus. There are other functions, described below, such as making wordlists, collocates and keywords, which can be used as starting points which allow the corpus to suggest things to look for and investigate.

Use of a concordance program does not necessarily imply that research is corpus-driven. It is perfectly possible to use a concordance program simply to look for data to support a hypothesis which has been arrived at by some means other than analysing the corpus, and most research done using a corpus is probably of this type.

Another important type of work which concordances make possible is data-driven learning. For the language learner, use of a corpus can be a substitute for intuitions which the native speaker acquires through exposure to the mother tongue (e.g. Lamy and Klarskov Mortenson, no date).

There are many other areas where the qualitative analysis

of concordances is essential for identifying and analysing patterns in language. One important cluster of related concepts relating to collocations, but which rely on examining concordances, are semantic prosody (Louw 1993/2004), semantic preference (Sinclair 2004) and lexical priming (Hoey 2005).

3.4. Technical requirements

Corpus analysis tools will either search through the corpus as a set of text files, or corpora may be pre-indexed, allowing for faster retrieval and more powerful queries. Some tools require corpora to be in particular formats (e.g. plain text, XML, or some non-standard format). Care should be taken to ensure that the particular forms of character and text encoding, file format and markup are being interpreted in a sensible way by the program. This will be more straightforward if the corpus itself is constructed in a fairly standard way and the corpus design and encoding are well documented.

Concordances are usually generated by a program for on-screen display, but it may be essential to save them for use again, or in a different way. While a concordance can often easily be generated again by submitting the same query to the same corpus, this may not be possible in some cases. If some complex series of processing steps has been taken, such as sorting, categorising, or thinning the lines, then it may be difficult to reproduce the results.

Some of this processing may have to be done by manual selection or annotation, and then this work certainly needs to be saved. There are other reasons why concordances may need to be saved: access to the tools or corpus may be temporary; the corpus may be under development and may change; the tools may be updated and change their functionality in subtle ways. Furthermore, it may be necessary to make the concordance available outside of the program which generates the concordance, so that it can be processed with other tools, or used in teaching, on a website or in a publication. It would therefore be necessary to save the concordance in some portable format, such as HTML. A user should consider whether these functions are available or necessary when selecting (or developing) concordancing software.

The following sections deal with further refinements and enhancements to the concordance function.

4. Searching using annotation

4.1. Description

Corpora often contain various types of tagging. These tags exist in the files in addition to the words which make up the texts, and can include tags which encode descriptions of the corpus and its constituent texts

(descriptive metadata), tags which encode information about the text structure, formatting and appearance (structural markup), and tags which encode various levels of linguistic categorisation or analysis of the text (linguistic annotation). It can be useful for the purposes of linguistic analysis to search for examples of words or other units which have been categorised by the use of these tags. In particular, the analyst may wish to exploit the linguistic annotation in a text, such as wordclass tagging, or lemmatisation. If a corpus analysis program offers the necessary functionality to interpret the tagging in a sensible way, then it should be possible to search for all examples of a particular word when it is tagged with a particular wordclass categorisation, for example 'effect' as a verb (see figure 4).

Methods for using the descriptive metadata and structural markup are described in the section 'Searching with metadata' below.

4.2. Example

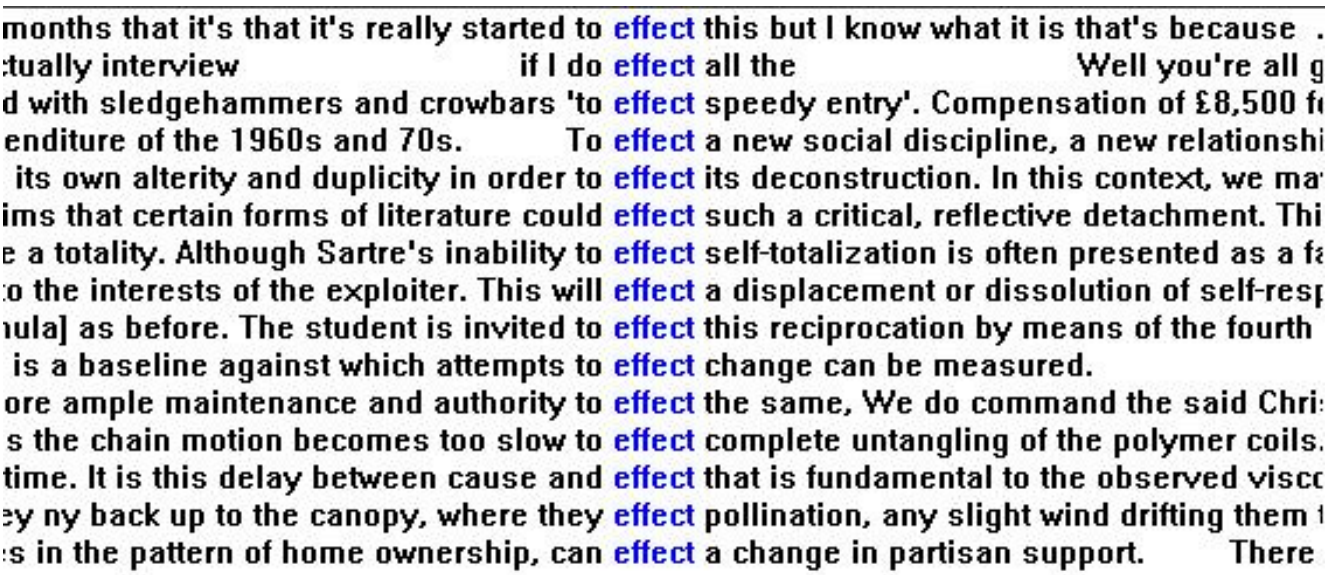


Figure 4: 'effect' used as a verb

Figure 4 shows a concordance of the word 'effect' where it has been tagged as a verb in the BNC-Baby corpus.

4.3. Analysis with this function

Exploiting the annotation to specify search terms can help to make more refined, and more grammatically targeted searches. For example, grammar books may say that it is not permissable to say 'less books' or 'less examples', and that it should be 'fewer books' and 'fewer examples'. It is possible to test this prescriptive rule by looking in a corpus at the evidence of what native speakers really say and write. For this example, the BNC-baby corpus was used. BNC-baby is a corpus which is a subset of the British National Corpus, containing 4 million words of written and spoken English. Searching for 'less' immediately followed by a plural noun in the corpus yielded no results, while there were 40 examples of 'fewer' immediately followed by a plural noun.

This not does give conclusive proof that 'less' does not occur before plural nouns, or that the prescriptive rule is correct. A slightly more sophisticated search pattern (allowing adjectives to occur between 'less' and the plural noun) yielded the following example from this corpus: "even if on the lower rungs with less promotion chances

than white men". It is also worth noting that wordclass tags were assigned in this corpus automatically, and it is a possibility that the tagging program would not have been willing to assign a noun tag to a word following 'less', because of the "rule" which does not permit this sequence.

Searching using the annotations can help to reveal grammatical patterns in the corpus, and it can also be used to find the grammatical patterns which tend to occur with certain words. This tendency for certain grammatical patterns to associate with certain words is known as colligation.

In can also be argued that there is a danger of analysts focussing on the interrogation and analysis of the more abstract and interpreted categories - the annotations - rather than words of the text itself. Indeed there is a danger of circularity in this methodology, if the user simply retrieves the information which has been inserted in the form of annotations by other linguists, or even by themselves, without retrieving any other useful information about the text.

4.4. Technical requirements

This function depends on the presence of markup tags in the corpus. The usefulness of the function depends on the quality of the markup, and on the accessibility and

quality of the documentation of the markup which is available to the analyst. If the user does not know the tags, or understand the ways in which they have been applied, then it is very difficult to use them and it is easy to misinterpret results. In order to exploit the tagging with software, the software needs to know how to identify and process the tags in the text. (To put this in a more technical way, it is necessary for the corpus analysis application software to be interoperable with the text markup formalism.) For this reason it is useful for the tagging in the corpus to be inserted into the text in a reasonably standardised way, for example as XML tags. XML is a standard way of inserting metadata and markup in a document. If a non-standard form of tagging is used, concordance software is less likely to recognise the markup, and may be unable to differentiate it from the text, or to make use of it in any useful ways. Using non-standard markup can mean that the user is tied to software written specifically to process that markup, which is likely to restrict access to the corpus and reduce it's usefulness, especially if the documentation and software specific to the corpus do not survive in the long term.

If the tags are stored separate from the corpus text, in a separate file as "stand-off" markup, then the risk of the tags interfering with the processing of the text is diminished. On the other hand, the computational task of using the tags is made more difficult, and there may be

few, if any, standard corpus analysis tools available which can successfully process the tags. However, this type of markup is likely to become more standardised and widely used in the future.

An alternative to the use of annotation to carry out research of this type is to use grammatical information held separately from the text. Rather than inserting the information in the text, or as stand-off markup, it is the software which holds the information, or makes the link with the information held outside of the corpus in dictionary or grammar files. For example, morphological tables may hold lists of inflected forms, which can be used for searches for verb paradigms, rather than using lemma annotations in the corpus. One important disadvantage of this approach is that ambiguous occurrences in the corpus are unresolved, or must be resolved "on the fly" by the software each time that the user wishes to make use of them.

5. Expanding the co-text

5.1. Description

Expanding the co-text is a function which can be applied to concordance lines. A concordance line will often be presented to the user as one line on the screen, with perhaps 4 or 5 words visible either side of the search

term. In order for the analyst to be able to read and understand a particular concordance line, it is often necessary to be able to read beyond this limited amount of co-text. It is therefore useful to be able expand the amount of co-text which is available. Some concordancers only give access to a few extra words or lines; some restrict the scope to the context to some textual unit such as the sentence or paragraph; others allow the analyst to start from the concordance line and read as far as they wish.

5.2. Example

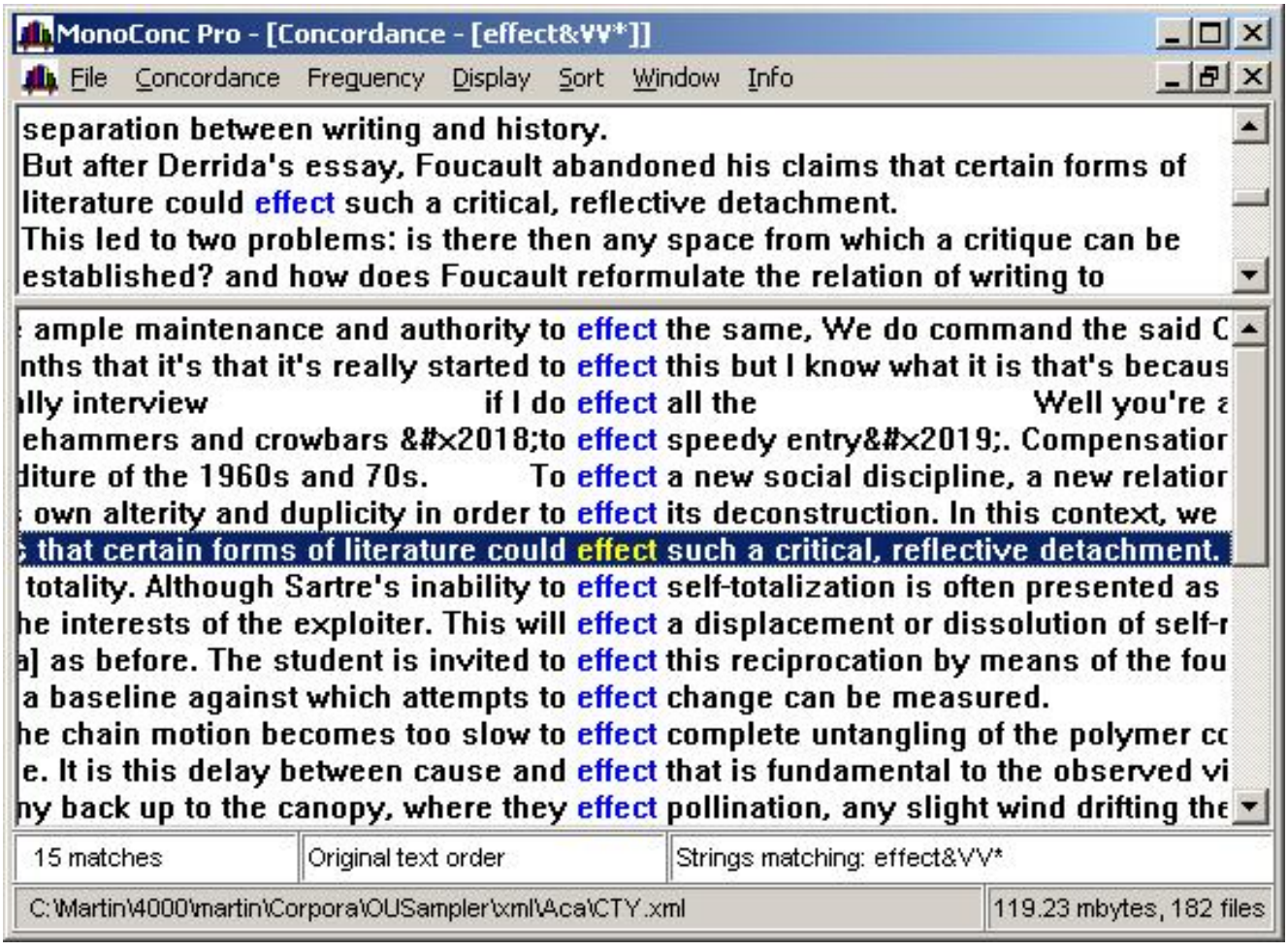


Figure 5: Concordance of 'effect' used as a verb, with expanded context for the selected line

Figure 5 shows the concordance of 'effect' in the BNC-baby corpus (displayed in this case by the Monoconc program), with the expanded co-text for the selected line displayed in the box above the concordance lines.

5.3. Analysis with this function

Being able to read more of the co-text is essential when the analyst wishes to take account of meaning in the text. It may be necessary to read a long way in a text to get enough information to be able to account satisfactorily for the occurrence of a particular linguistic feature in a text.

5.4 Technical Requirements

Most concordance programs which make use of a corpus installed locally offer the functionality to expand the co-text. Online concordance programs may restrict the amount of co-text available in order to prevent the user from downloading an entire text or corpus. This may be necessary because of licensing restrictions on the distribution of the corpus. In this case, it may be necessary to attempt to negotiate access to the full corpus text for more detailed analysis of concordance lines.

6. Sorting

6.1. Description

Sorting is a function which can be applied to concordance lines from a corpus. Sorting the lines on the basis of various criteria may be necessary to reveal patterns in the words surrounding the search term. The criteria according to which lines can be sorted include:

- the order in which they occur in the corpus (often the default, and may be called 'original order');
- alphabetically by node word (which is only relevant if there is some variation in the node word, such as when wildcards have been used, or when there have been multiple search terms, or search by wordclass or lemma, etc.);
- alphabetically by co-text: words in certain positions around the node word, e.g. one word to the right or left of the node;
- annotations (e.g. grouped together by wordclass tag, or user-defined annotations);
- metadata categories (e.g. text type, sex of speaker);

A further set of options is for each of these searches may be made in ascending or descending order (e.g. a-z or z-a in English).

When searching by co-text, it is often useful to try sorting by words in various positions to the left or right of the search term, and for values of n up to 5, or more where

there the search term is part of a long fixed phrase.

It is also possible to sort on the basis of more than one criterion, so that lines which are grouped together according to the first sort criterion are then further ordered, as in figure 6.

6.2. Example

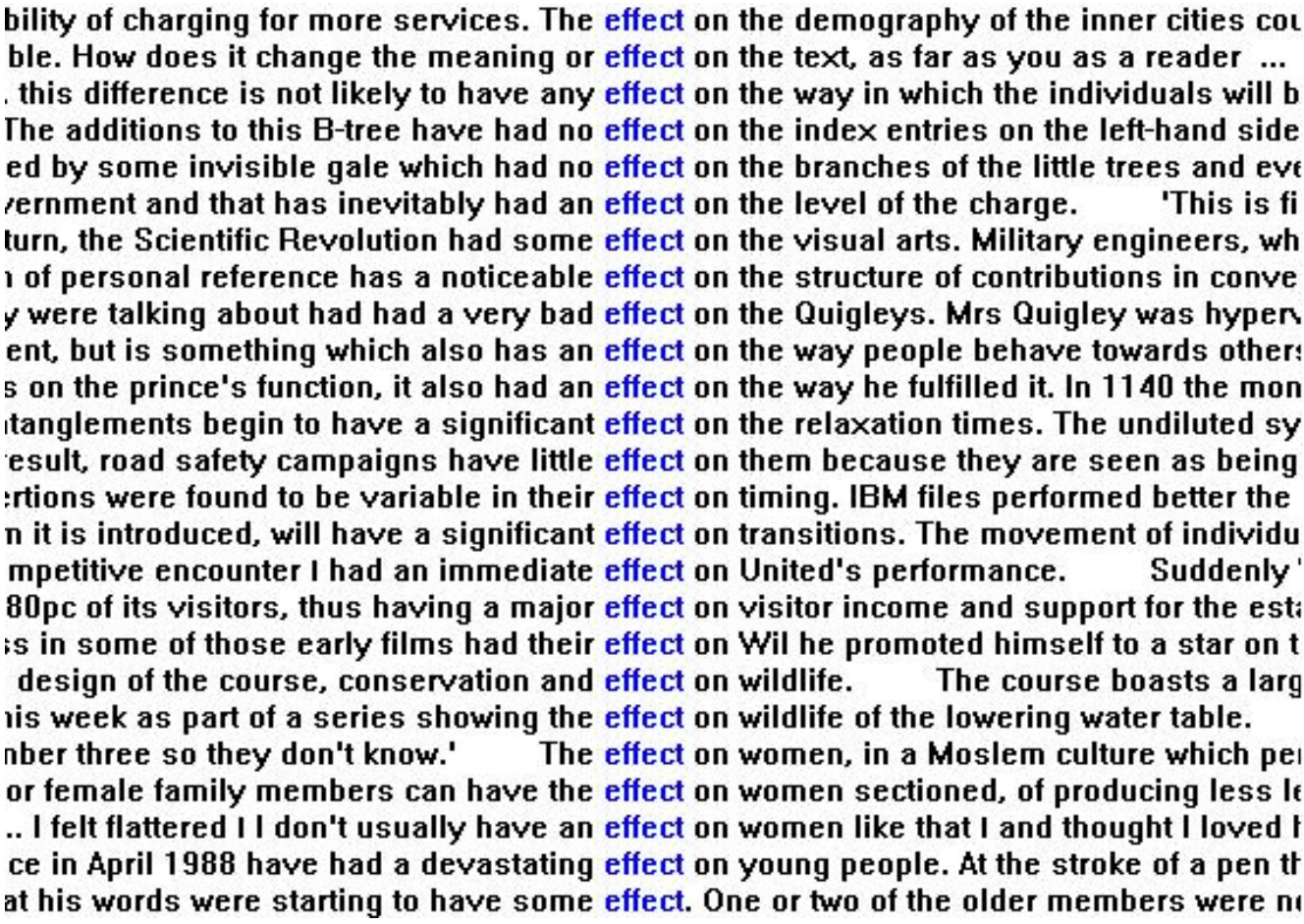


Figure 6: concordance sorted by first right then second right

Figure 6 shows a screenful of concordance lines for 'effect' from BNC-Baby, sorted by first word to the right, then second word to the right.

6.3. Analysis with this function

Sorting is often necessary to reveal the patterns of words surrounding the search term. These patterns can only be seen by the analyst when repeated occurrences of relevant features are grouped together on the screen. Sorting can help to bring these examples together. In this way, simply sorting on various positions and viewing the lines can reveal hitherto unseen patterns. It may also be necessary to vary the number and selection of examples in order to spot the interesting patterns.

6.4. Technical Requirements

A concordance program should provide the functionality to sort concordance lines. There may be variation in the number of criteria allowed, which criteria may be applied, and the range of co-text over which they may be applied.

7. Searching in concordances

Much of the work of searching and concordancing is about finding out which words tend to occur in the vicinity of other words, and searching in the co-text surrounding the search term in concordance lines may be useful if the analyst is looking for examples of a particular word or pattern.

The linguist can start to find the words which occur with

the search term by sorting the concordance lines and by computing collocations. When some potentially interesting words are suggested the list of collocates, or the examination of the concordance lines, the linguist will want to search for the lines in which the word occurs.

Not all concordance programs provide the functionality to search within the concordance lines for a particular word or pattern. It may be necessary to use sorting to find the word, although this can be difficult if it is appearing in many different positions around the search term in the KWIC concordance. Words that occur above a certain frequency will appear in a list of collocates, and some programs will allow the user to switch from the collocates list to show a concordance of all the lines in which the collocates appear with the search term.

8. Thinning

8.1. Description

Thinning concordance lines is a function which reduces the number of lines in a concordance, by selecting a subset of the lines based on some criterion. This may be done in order to reduce the number, if there are too many to analyse, or because the analyst is only interested in a particular subset.

Ways of thinning concordance lines include reducing to the set to every nth occurrence, to n per text, or to the first n examples (where n is any positive integer). A set of concordance lines may also be thinned on the basis of user annotations (see section 9 below).

Searching in results to produce a reduced number of concordance lines (see section 7 above) can be one way of thinning the concordance lines. Some programs allow the user to search for a string in the concordance lines, and then thin the set of concordances to only those which contain the search string.

8.2. Example

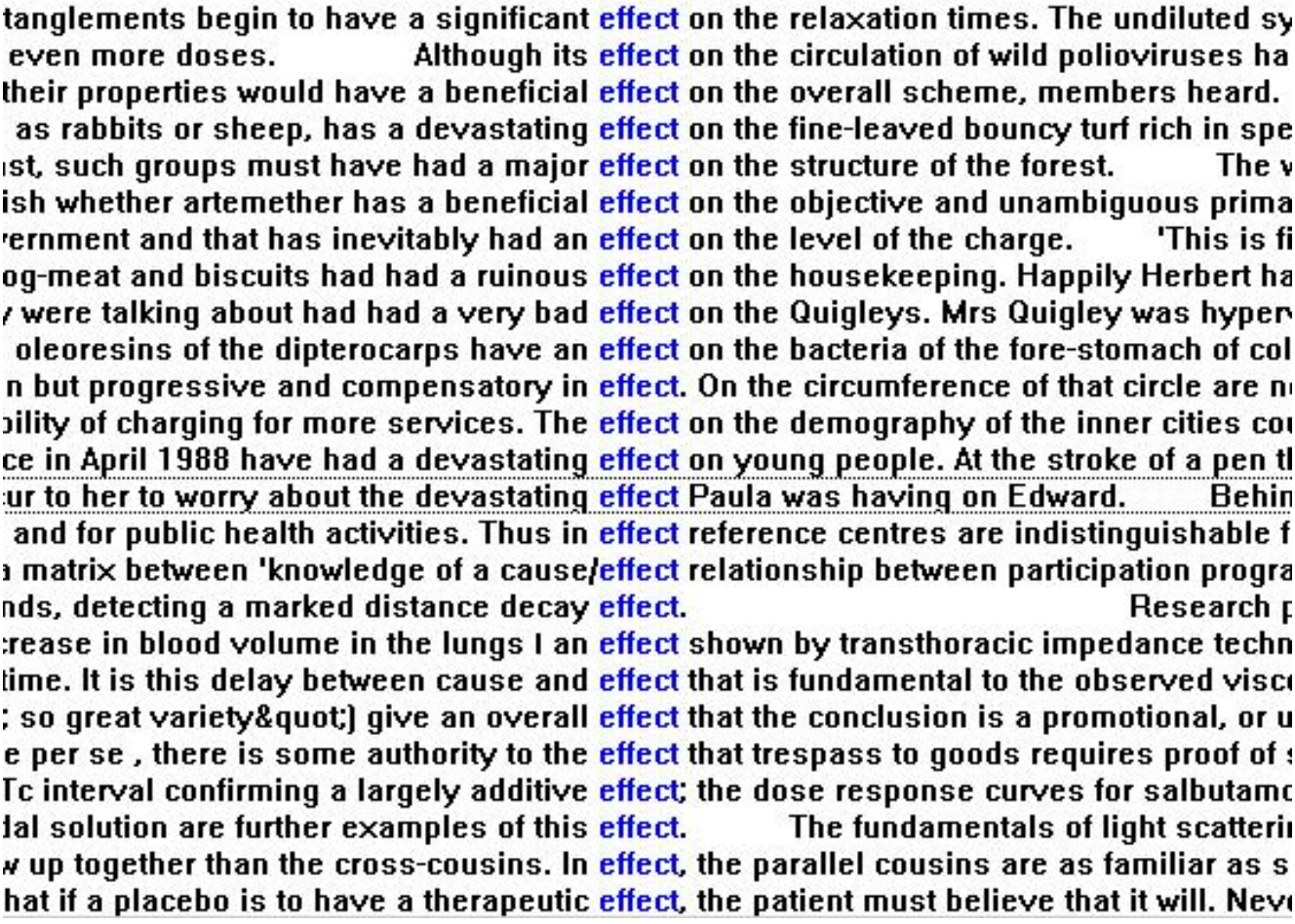


Figure 8: the concordance from figure 6 (sorted on right co-text), thinned to display only every 5th occurrence.

8.3. Analysis with this function

Thinning lines is often part of the heuristic process of focussing the analysis on a particular area of usage in the corpus. A corpus-driven enquiry will typically start with a search for a particular form, followed by analysis of its meaning and contexts, and then searching for a longer phrase.

Thinning concordance lines is used chiefly for providing an appropriate number of examples for a human analyst to be able to view. This may be done for use in the classroom, so as not to swamp or intimidate the student with two many examples. Manually thinning lines is also possible, and may be useful for illustrative or pedagogic purposes, but there is a danger of making a biased selection, and it is important that the person reading the concordance knows that the lines have been manually selected.

If it is intended to generalize from the analysis of the sample, then it is necessary to be aware of the way in which the corpus is structured, and to decide whether the sample is likely to be representative of all the examples. In a similar fashion, if the intention is to generalise about the

language on the basis of a corpus, the linguist must also always bear in mind the way in which the texts in the corpus itself have been sampled from the overall population of texts. Analysing only a limited number of the concordance lines may be necessary from a practical point of view, but the analyst must bear in mind that the analysis is based on a sample of a sample.

It is also possible, at least in principle, to apply automatic procedures to thin concordance lines by selecting one or two examples which exemplify typical patterns of usage. This is an attempt to automate the work of finding typical patterns of usage in concordance lines, and may be useful for pedagogical, or for lexicographic applications. Concordance output thinned in this way may be able to show something of the variety of different usages, but will not show patterns of repeated usage in and around the search term. Such a concordance must be read differently. The analyst should not look for repeated occurrences as evidence of typicality, because lines displaying some similarity will have been deleted, and a single typical example allowed to stand for them.

8.4. Technical Requirements

A concordance program may be able to thin results, or the same result may be possible by re-running the query with a different search term, or with more filters, for example by searching for a phrase, or by limiting the results to

every nth occurrence, as described in section 2 above.

Automatically thinning a concordance to produce typical examples, as discussed above, requires software to implement complex algorithms to interpret the patterns in the co-text and to select typical examples.

9. Categorising

9.1. Description

This is a function which can be applied to concordance lines from a corpus. It is sometimes useful for the analyst to be able to manually categorise the concordance lines, for example to classify different senses of a word which the analyst is able to assign by reading the concordance. Categorising concordance lines can also be used as a way of manually thinning the concordance.

9.2. Example

In the example in Figure 9, the analyst has assigned letter codes ('i', 'j', 'r' and 'v') to each of the 21 concordance lines for 'fast' (every 15th example sampled from BNC-baby).

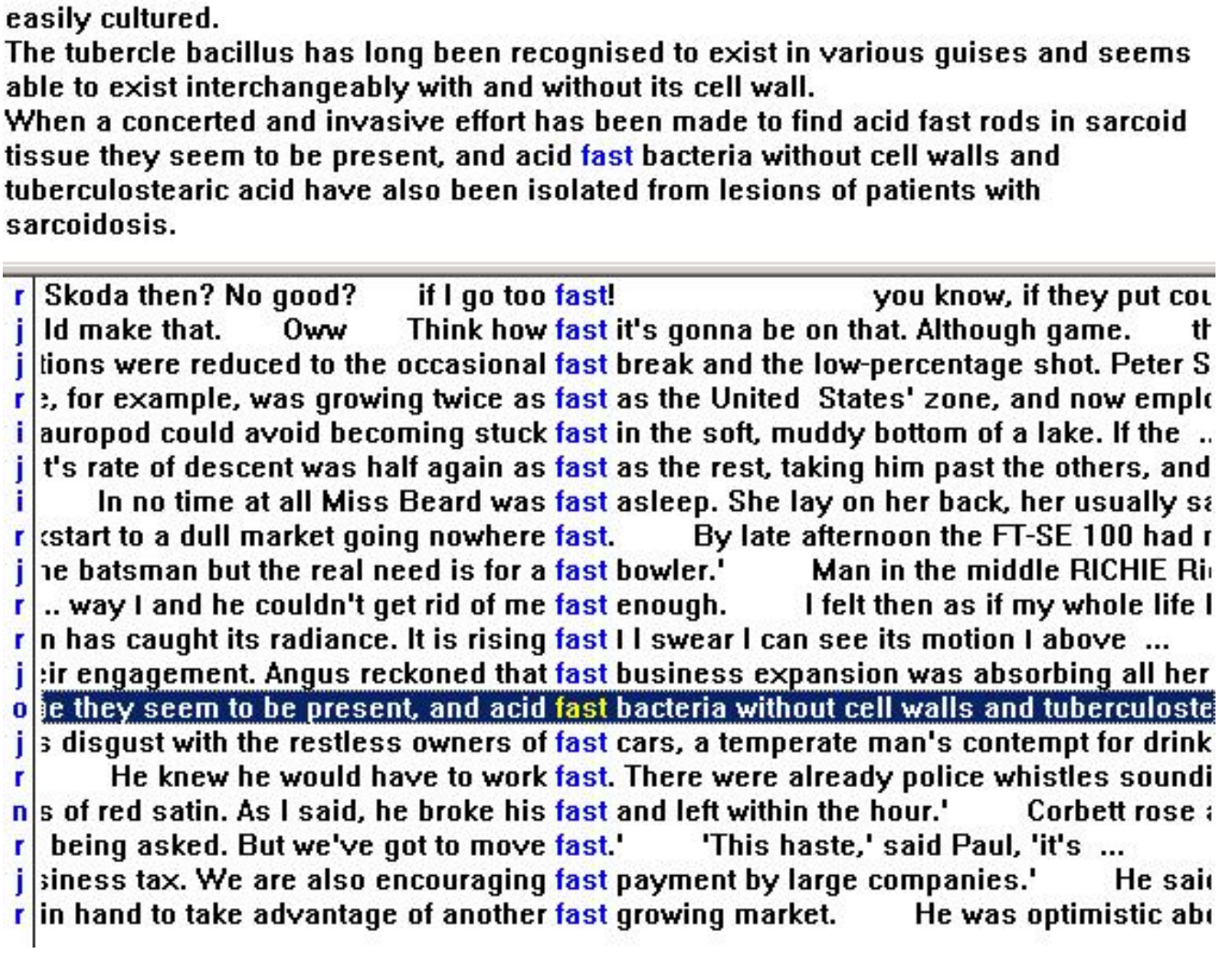


Figure 9: Categorisation codes assigned to concordance lines (using the Monoconc program).

The categorisation has been done as follows: 'r' indicates 'fast' is an adverb, meaning quickly; 'j' indicates that 'fast' is an adjective, meaning quick; 'n' indicates that 'fast' is a noun, meaning to go without food, and 'i' indicates that 'fast' is part of an idiomatic expression, partially or fully de-lexicalised. One line has also been tagged 'o', for 'other', and it is often useful to have such a category for problematic examples. Examining more concordances would probably should yield more evidence, making it possible to categorise this and other difficult examples, and would involve increasing the number of categories.

9.3. Analysis with this function

Categorising lines manually is necessary where it cannot be done by specifying formal criteria in the searching or thinning stages, either because the functionality is not available, or the necessary level of annotation is not present, or, most likely, because the desired categorisation requires human intervention and analysis. This type of categorisation is therefore be seen as often a type of research where the concordance is a tool to help manual, qualitative linguistic analysis.

9.4. Technical Requirements

This type of manual annotation of concordance lines is often done on concordance printouts with a pen. Software which allows the annotation to be done on the electronic concordance data makes it possible to sort on the basis of the annotations, and to thin the concordance to leave only those lines with or without a certain manual categorisation.

10. Hiding the node word

10.1. Description

A simple but powerful pedagogic exercise can be created

by hiding the search term (or node word) in a KWIC concordance. A human subject can then be shown the concordance lines with the node word invisible, and they must try to guess what the word is. An alternative, or additional, task is to ask the student to identify the wordclass.

10.2. Example

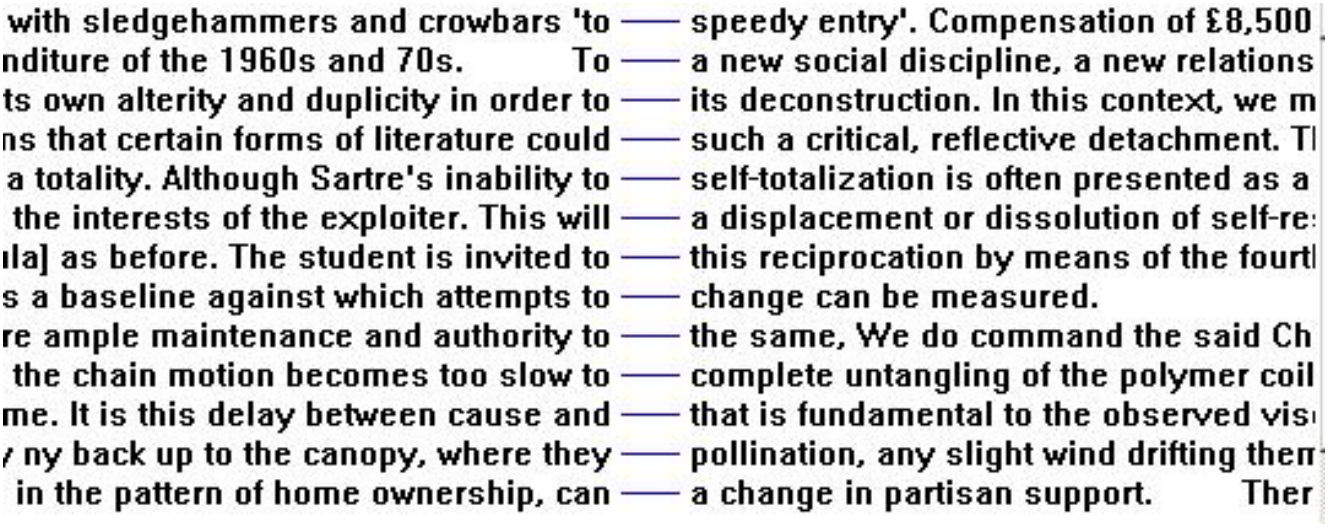


Figure 10: Concordance with node word concealed.

10.3. Analysis with this function

The use of this technique is usually pedagogically motivated. It can be used as a language awareness training exercise for native or non-native speakers.

10.4. Technical requirements

It is useful if the software can do this and print or save the lines with the node word hidden. Otherwise the user can save the concordance and then edit it in another

application (such as a text editor or word processor), or even simply print it out and black out the node word with ink.

A related technique that could be useful for teaching purposes would be to conceal the significant collocates where they occur in the concordance lines.

11. Showing collocates

Collocates are words which tend to occur frequently in the vicinity of the search term. Some concordance software applications can silently compute the significant collocates of the search term in the corpus, and represent these words in a particular way in the concordance view, for example by colouring them. In figure 11, the collocates are shown in bold and italic type to differentiate them from other words in the co-text.

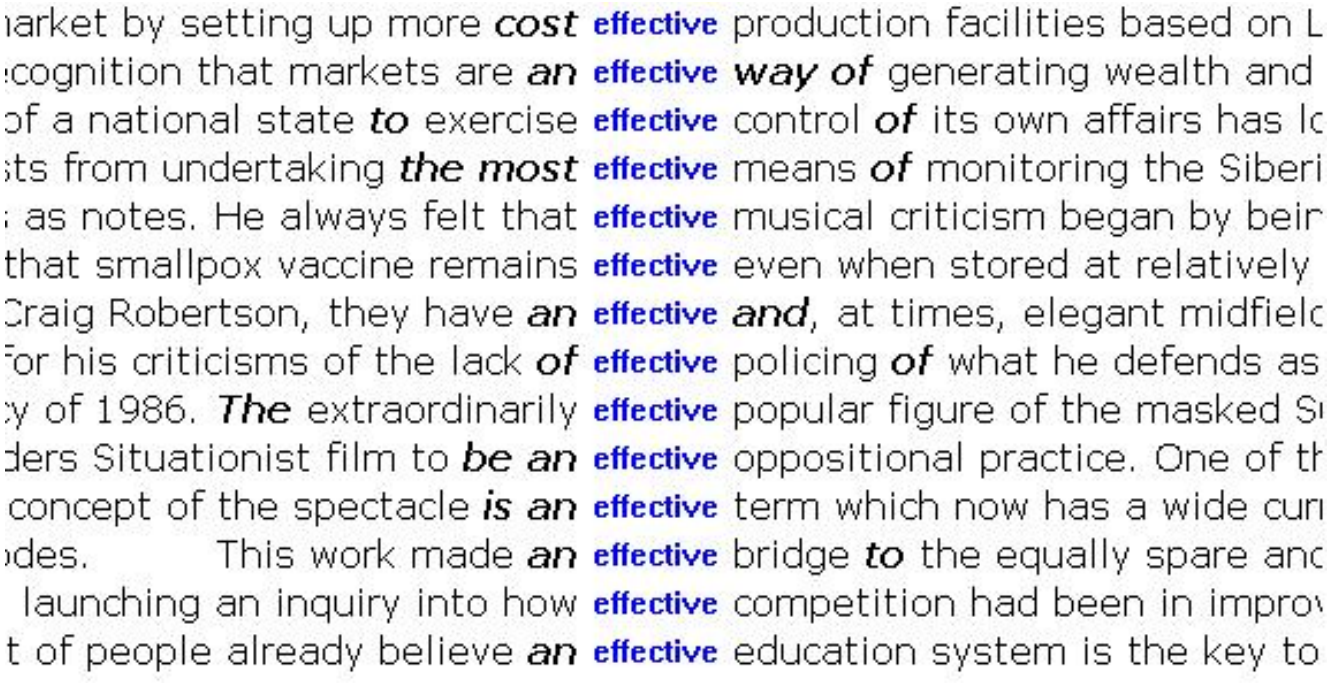


Figure 11: collocates of the node word

This can help to identify patterns of co-occurrence in the concordance lines, particularly where there are too many examples to see in one screenful, or where the position of the collocate is variable. This is a useful function, because while the linguist may be able see repeated co-occurrences of words and structures, it is not possible to assess the statistical significance of these features simply by looking at them.

This method of silently computing and displaying the collocates does risk obscuring the process of calculation from the user. The linguist should remember that there are various ways to calculate collocates, and choices need to be made regarding, among other things, the collocation window, the basis for establishing what is the expected frequency of co-occurrence, the metric for assessing significance and the thresholds for frequencies and significance. Showing collocates in the concordance window should be seen as only a quick or preliminary indication of potential collocates, which are likely to require more focussed investigation and verification.

Concordance programs will also typically be able to generate lists of significant collocates, sometimes lists of positional collocates, showing which words tend to co-occur in particular positions to the left and right of the

node word. Such lists can be invaluable for suggesting further searches to produce concordances and examine patterns of usage. Investigating collocation is a very important part of the corpus linguistics basic toolkit, and is covered in Chapter 57 in this volume.

12. Displaying annotations

12.1. Description

A corpus may include various tags, which may encode descriptions of the texts constituting the corpus, elements of the text structure (e.g. paragraphs), or linguistic annotations (e.g. wordclass tags) (see section 4 above). Concordance software sometimes has the option to hide or display markup.

One possibility is for the concordance software to colour the different parts of speech, so that nouns are red, verbs blue, for example. This is likely to be easier to read than viewing the concordances with the tags displayed inline with the text. The analyst can see the wordclass categorisation without interruption to the stream of words.

The analyst may wish to be shown tags associated not with the individual words or lines in the concordance, but rather the information associated with the whole text from which the particular concordance line is derived. For

example, the name of the file, or title of the text may be useful. Another option would be to indicate, again perhaps by the use of colour contrasts, the category of text from which the example comes. For example, in a corpus such as BNC-Baby which is designed to compare and contrast newspapers, fiction, academic writing and spoken conversation, examples from each of these four sub-corpora could be coloured differently.

12.2. Example

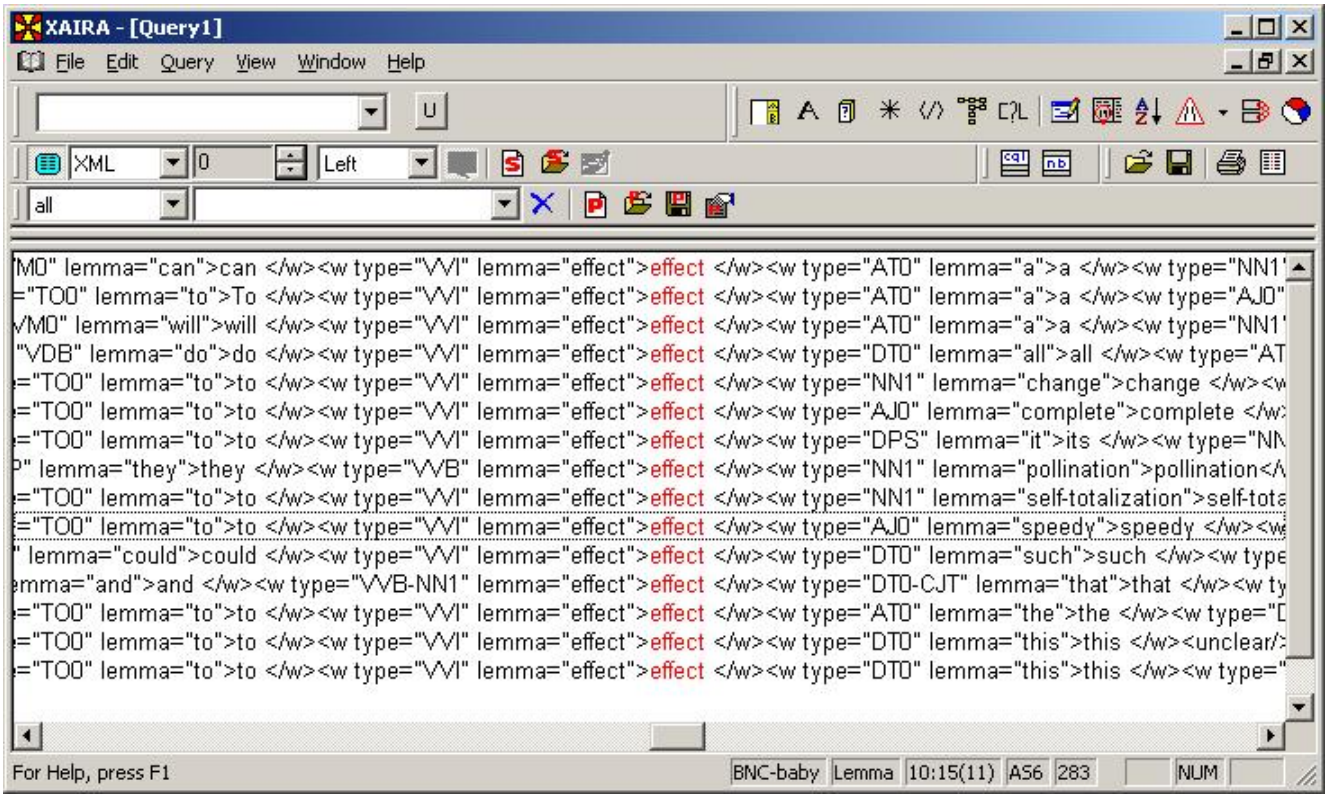


Figure 12: displaying XML annotations in the BNC.

Figure 12 shows a concordance from the BNC-Baby corpus with the Xaira corpus analysis program. Xaira is an XML-aware program, which means that it can recognise and manipulate the metadata and annotations in the corpus in sophisticated ways because it knows how XML

works. XML is an international open standard for marking up documents. Since the annotations in BNC-Baby are XML tags, XML-aware programs can selectively display or hide the tags.

12.3. Analysis with this function

Viewing the markup associated with concordance lines may sometimes be useful in order to help interpret some of the concordance lines, or to make more patterns visible. It is essential for checking the results of searches using annotations (see section 4 above). When unexpected results are obtained from searching for a particular wordclass tag, for example, it may be necessary to read the tags to find out whether they have been incorrectly assigned, or at least to try to understand the ways in which the wordclass tags have been assigned. There is little consensus in linguistics about how wordclasses should be categorised, and therefore there is a lot of variation in the ways in which different analysts or different programs will assign tags.

12.4. Technical requirements

Applying this function usually requires that the corpus text has been annotated. As described above (see section 4 above), the possibility of implementing this function, and of it being useful to the analyst, depends on the manner in which the markup has been encoded and documented. It

is possible in principle that markup could be applied by the software on the fly, but relying on automatic tagging is likely to involve problems of accuracy and consistency as well as additional computational processing.

Displaying information about the source text is usually useful. Many programs will show the file or text name alongside each concordance line. Others will allow the user to select a line and then view the metadata associated with the text, such as the title, author, date of publication, etc..

The possibilities for selecting elements of the markup and using them to display words or lines differently will become increasingly possible as standards for the encoding of corpora in XML are developed and stylesheet functionality is incorporated into XML corpus analysis tools, to enable the user to adjust and control the display of the output.

13. Using metadata

13.1. Description

With certain corpora and analysis tools, it is possible to restrict the scope of searches to texts (or elements of texts) with particular characteristics. For example, some corpora contain texts which originate in both written and

spoken modes. If a corpus is marked-up in such a way that the component texts are clearly marked up as written and spoken, then searches can be restricted to one section or the other, and frequencies compared.

13.2. Example

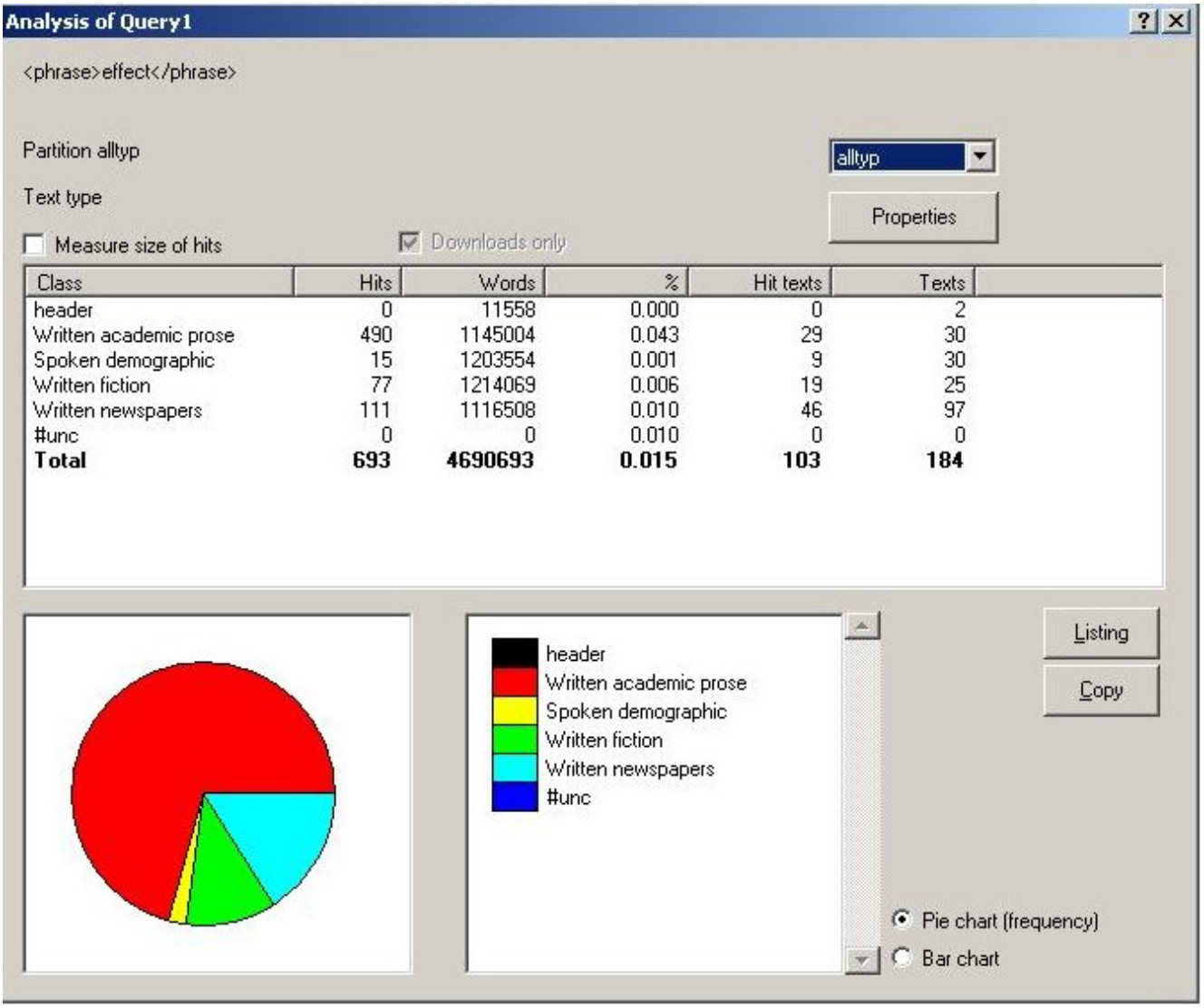


Figure 13: distribution of 'effect'

Figure 13 shows a pie chart from the Xaira program which displays graphically the distribution of the word 'effect' in the BNC-Baby corpus. BNC-Baby is divided in to four subsections, each of approximately 1 million words,

representing spoken conversation, fictional prose, academic writing and newspapers.

13.3. Analysis with this function

Using metadata to search in particular texts or elements of texts can be particularly useful for research which aims to exploit differences in register, genre, mode and text type among the texts in a corpus.

It should, however, be borne in mind that the design criteria of a corpus may have aimed to sample certain categorisations in a balanced and representative way, while others may simply be indicated and were not part of the corpus design criteria. It could be a mistake to conclude from a metadata-sensitive search that Sinclair (2005) argues that only elements which are designed to be balanced and representative should be contrasted in research using a corpus.

Careful attention should always be paid to the documentation of the design and implementation of the corpus metadata. For example, the BNC Handbook indicates what categories were design criteria for the corpus. It also indicates that, for the spoken conversational part of the corpus, there is a metadata category which could be used to indicate the sex of the speaker, but which is often only recorded for the main respondent (the person carrying the recording device),

and not for the other interlocutors. It is therefore difficult to carry out research to compare the speech of men and women using the BNC, although not impossible (e.g. Leech, Rayson and Hodges 1998).

13.4. Technical requirements

In common with other functions which exploit the markup in a text, application of this function requires access to software which is capable of interoperating with the markup. Successfully doing this is rarer than successful processing of structural markup such as paragraph and page breaks, and annotations associated with individual words, such as wordclass tags. This is because it is necessary for software to recognise and process annotations with a wider, and more complex scope. There may be descriptive metadata which applies to the whole corpus, to sections of it, to subsections, to individual texts, to groupings of texts in different sections, or to certain passages within various texts. For this reason, software which can work with descriptive markup in a sophisticated way will often be tied to a particular corpus, or corpus markup framework.

If the limitations of the corpus markup or the corpus analysis software mean that it is not possible to restrict searches on the basis of metadata categories, then an alternative approach is to put files representing texts from different text types in different folders. Similarly, file

naming conventions can be used, so that in effect, codes representing metadata values are encoded in the file name. With these methods, the operating system's method of storing files is exploited to allow the analysis of certain corpus components. While certain types of analysis can be done this way, there are various limitations and potential problems.

These problems include the following:

- the user has to load and analyse different subcorpora and find a way of comparing results;
- in transferring between computers, it may be difficult to preserve the file hierarchy and the file names, and for this reason this solution is not suitable for long term preservation of the corpus metadata;
- it can be difficult, and sometimes impossible to apply multiple metadata categories to define sets of texts - the user may be restricted to one pre-defined categorisation which is fixed in the arrangement of files into folders;
- some operating systems and some concordance programs restrict allowable file names and suffixes, and the number of files in a folder, so a solution may not work in all computing environments.

14. Wordlists

14.1. Description

A wordlist (sometimes written 'word list') is a list of all the different words in a text or corpus, usually accompanied by the number of times each word occurs. While users may simply press a button in a software application to make a wordlist, they should be aware that making a wordlist depends on having a working definition of what a word is, and of the what counts as an occurrence of the same word, and that this will vary from language to language.

It is also possible to make lemma lists, where inflected forms of words are counted together as examples of occurrence of a head word, or lemma. Other phenomena in a text, such tags, or word and tag pairs, may also be counted and listed.

14.2. Example

Rank	Frequency	Lemma	Wordclass
1	6187267	the	determiner
2	4239632	be	verb
3	3093444	of	preposition
4	2687863	and	conjunction
5	2186369	a	determiner
6	1924315	in	preposition
7	1620850	to	infinitive-marker
8	1375636	have	verb
9	1090186	it	pronoun
10	1039323	to	preposition
11	887877	for	preposition
12	884599	i	pronoun
13	760399	that	Conjunction
14	695498	you	pronoun
15	681255	he	pronoun
16	680739	on	preposition
17	675027	with	preposition
18	559596	do	verb
19	534162	at	preposition
20	517171	by	preposition

Figure 14: 20 most frequently occurring words in a lemmatised frequency list for the BNC

14.3. Analysis with this function

Examining the wordlist from a large general reference corpus, as in Table 2 above, can be useful for finding out about the words which occur most frequently in a language. Examining the wordlist from a text or a specialised corpus can be a useful starting point for examining the lexis of a particular text or text type. It can be useful to compare a wordlist from a text or corpus with

the wordlist from a large reference corpus. It may be interesting, for example, if the most frequent words differ from the norm. The lower than usual occurrence of 'of', for example, may be a reflection of a lower number of post-modification of noun phrases, which may be a sign of a style which is simpler, or less formal in some way. For a more systematic way to find words that occur more or less frequently in a text compared to a reference corpus, the 'keywords' function can be used (see below).

When the user wants to look beyond the most frequently occurring grammatical words in order to see which are the â€˜content' words which are used most often, a stop list may be used. This is a list of words which the program omits from its searches. Such lists typically contains grammatical, or closed class, words, or simply the most common words in a corpus. If the user is interested in the grammatical words, then it may be necessary to make sure that a stop list is not being used by the program by default. Stop lists are sometimes used by corpus analysis programs to prevent users from searching for the most common words, as this would be too big a task for the software, and would produce too many results to analyse manually.

Wordlists derived from different components within a corpus may also be compared. Rayson et al (1997) were able to compare wordlists derived from various sets of speakers in the spoken component of the BNC, and

conduct a quantitative analysis of speakers categorized by such factors as sex, age, social group and geographical region.

Looking at frequently occurring words may also tell you something about the themes or topics in the texts in a corpus (see more in the section on Keywords below).

Obtaining lists of the most frequent words in a corpus can also be useful for pedagogic research. Knowing which words occur most frequently can help to indicate which words a learner is most likely to encounter. The top of a wordlist can perhaps be seen as the core vocabulary of a language. The selection of words for learner dictionaries, grammars and other teaching materials is often informed by wordlists obtained from corpora.

14.4. Technical requirements

A wordlist may be made using a simple script or program, or can be done by a specialist program. A utility to make wordlists is usually part of a text or corpus analysis software package. Although it may be invisible to the user, compiling a wordlist depends on tokenising the text. Being able to count words depends on being able to recognise what a word is, and so some variation in word frequency counts will result form the differing tokenisation algorithms which are employed by different programs.

It is not necessary to use a corpus analysis software package to obtain wordlists from a text or corpus. Various scripts and programs are available, and a simple program can be written in a variety of scripting or programming languages to produce a wordlist. Typically, what such a program will do is to send the contents of a file (or a collection of files) through a series of processing steps to (i) remove punctuation characters and other non-word elements, (ii) identify word boundaries, (iii) sorting the words into alphabetical order, (iv) identify and counting identical words which are adjacent in the sorted list, and (v) sort the wordlist into descending order of frequency.

It is possible to obtain the result by various means, and to make different decisions about what to include or exclude, and how to identify word boundaries, etc.. Corpus analysis programs usually include functions to make a wordlist, with varying levels of control of these options.

15. Keywords

15.1. Description

The keywords of a text, in the sense intended here, are words which can be shown to occur in the text with a frequency greater than the expected frequency (using some relevant measure), to an extent which is statistically significant.

Confusingly, the term 'keyword' is used in more than one way in corpus linguistics. It is also used to mean the search term, or node word, as in Keyword in Context (KWIC) .The meaning in this section is an important, or 'key', word in a text. This latter usage is derived from Raymond Williams (1976), who uses it to signify culturally significant words in the discourse of a society. Mike Scott introduced the term to corpus linguistics and implemented a method of computing keywords in his Wordsmith Tools software.

It is usually most relevant to compute keywords for a text or a set of related texts in comparison to a reference corpus. It is also possible to compare a specialised corpus with a reference corpus to try to obtain an indication of characteristic lexis in the specialised domain.

An interesting development of the notion of keywords, is key key-words. Scott (2006) noted that while words are often computed as key in a particular text, they may not be significant across a number of texts of the same type. Those that are key across a number of texts in a corpus are called key key-words.

15.2. Example

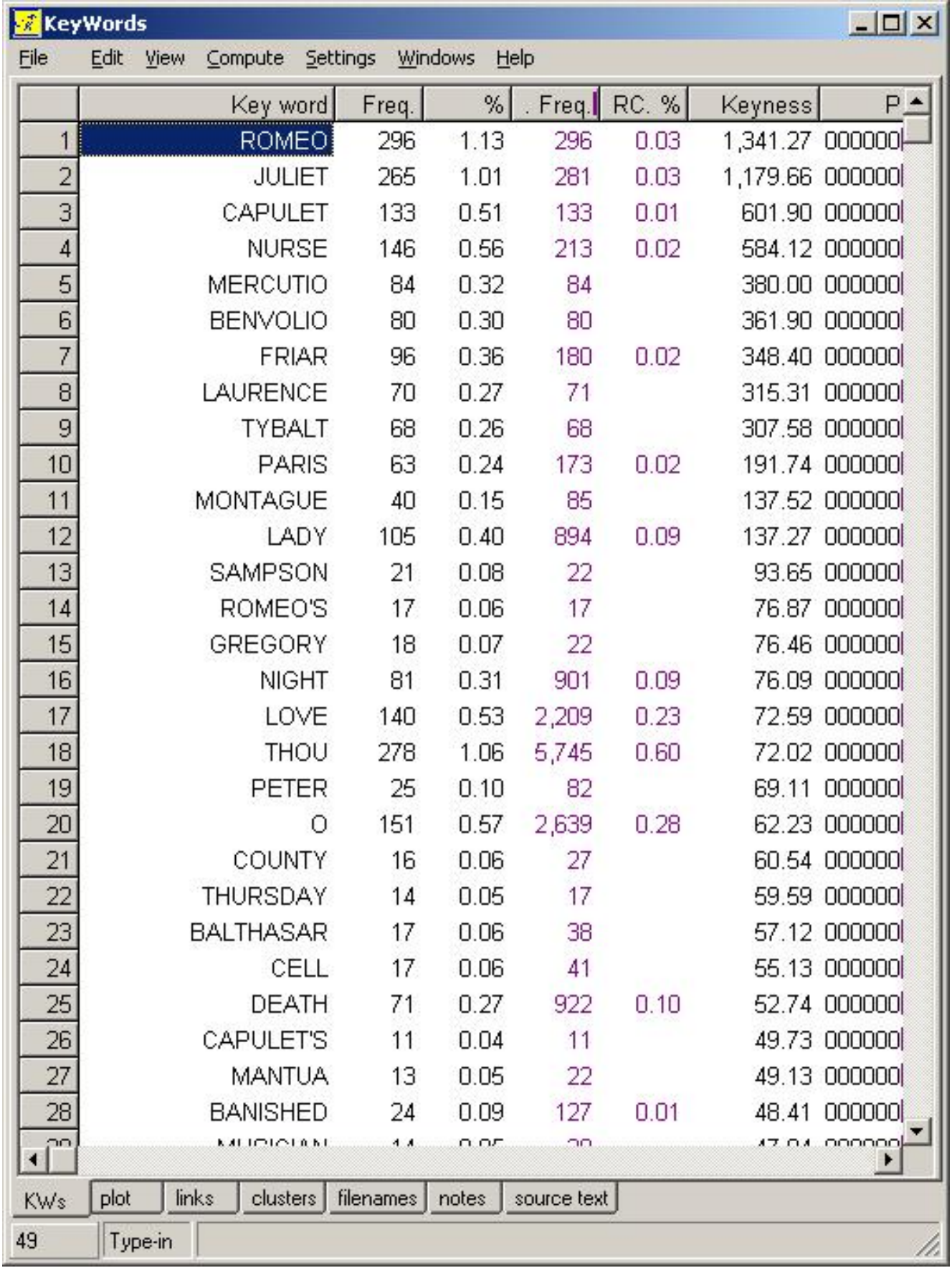
Figure 15a shows a list of keywords generated by Wordsmith Tools by comparing A Connecticut Yankee in

King Arthur's Court by Mark Twain with the written component of the British National Corpus, ranked in descending order of their significance, or keyness. (The usefulness of such a comparison is discussed in 15.3 below.)

I, AND, SIR, KING, YE, IT, MY,
LAUNCELOT, ME, WAS, KNIGHTS, MERLIN,
KNIGHT, ARMOR, CLARENCE, THING,
SANDY, HIM, MARHAUS, THAT, UPON,
TOWARD, MORDRED, GAWAINE, CAMELOT,
SAGRAMOR, SO, DOWLEY, YES, COULDN'T,
MILRAYS, THEN, BUT, THEY, HUNDRED,
PRESENTLY, KING'S, ARTHUR'S, WOULD,
MAN, HAD, WE, ALL, YONDER, THOU,
SLAVE, MIRACLE, OUT, ARTHUR, GOOD,
UNTO, COULD, AH, HATH, MYSELF,
ERRANTRY, LET, SMOTE, ALONG, WELL,
MAGICIAN, NOBLE, HIS, GOT, WHEREFORE,
SWORD, HE, EVERYBODY, THEE, SPEAR,
YOU, ABBOT, PERADVENTURE, OFFENSE,
HERMIT, THEM, PROCESSION,
STRAIGHTWAY, A, YET, MONKS, KAY,
EVER, GUENEVER

Figure 15a: keywords in A Connecticut Yankee

In the example given in Figure 15b below, the keywords from Shakespeare's Romeo and Juliet are given, as calculated in comparison to the rest of Shakespeare's dramatic works. In this way, words which appear relatively more frequently in this play than in the others should appear at the top of the list. The Wordsmith Tools program was used for this example, and the display includes frequency of the words in the text (Romeo and Juliet), and in the corpus (the rest of Shakespeare's drama), as well as the frequencies as a percentage of the total number of words in the text or corpus and a calculation of the â€˜keyness' of the words.


KeyWords

File Edit View Compute Settings Windows Help

	Key word	Freq.	%	. Freq.	RC. %	Keyness	P
1	ROMEO	296	1.13	296	0.03	1,341.27	000000
2	JULIET	265	1.01	281	0.03	1,179.66	000000
3	CAPULET	133	0.51	133	0.01	601.90	000000
4	NURSE	146	0.56	213	0.02	584.12	000000
5	MERCUTIO	84	0.32	84		380.00	000000
6	BENVOLIO	80	0.30	80		361.90	000000
7	FRIAR	96	0.36	180	0.02	348.40	000000
8	LAURENCE	70	0.27	71		315.31	000000
9	TYBALT	68	0.26	68		307.58	000000
10	PARIS	63	0.24	173	0.02	191.74	000000
11	MONTAGUE	40	0.15	85		137.52	000000
12	LADY	105	0.40	894	0.09	137.27	000000
13	SAMPSON	21	0.08	22		93.65	000000
14	ROMEO'S	17	0.06	17		76.87	000000
15	GREGORY	18	0.07	22		76.46	000000
16	NIGHT	81	0.31	901	0.09	76.09	000000
17	LOVE	140	0.53	2,209	0.23	72.59	000000
18	THOU	278	1.06	5,745	0.60	72.02	000000
19	PETER	25	0.10	82		69.11	000000
20	O	151	0.57	2,639	0.28	62.23	000000
21	COUNTY	16	0.06	27		60.54	000000
22	THURSDAY	14	0.05	17		59.59	000000
23	BALTHASAR	17	0.06	38		57.12	000000
24	CELL	17	0.06	41		55.13	000000
25	DEATH	71	0.27	922	0.10	52.74	000000
26	CAPULET'S	11	0.04	11		49.73	000000
27	MANTUA	13	0.05	22		49.13	000000
28	BANISHED	24	0.09	127	0.01	48.41	000000

KWs plot links clusters filenames notes source text

49 Type-in

Figure 15b: keywords from Romeo and Juliet

15.3. Analysis with this function

Keywords are an attempt to characterise the topic,

themes or style of a text or corpus. As such, compared to some other forms of analysis using a corpus, keywords analysis tends to focus on the ways in which texts function, rather than on overall characterisations of a corpus, or focussing on isolated linguistic elements in the corpus. For this reason, it is a technique which is popular in various forms of discourse and stylistic analysis (e.g. Culpeper 2002).

It is possible to obtain a lexical characterisation of a text using keywords analysis. The analyst needs to be careful how to interpret results: words can show up as keywords because they are related to the topic, and this is especially likely with proper nouns. Another potential problem is that the value of keywords analysis depends on the relevance of the reference corpus. Comparing a US nineteenth century novel with the BNC should produce some keywords typical of the topic and style of the novel, but will also show up words which are more typical of US English, words with are more typical of nineteenth century English, and words which are more typical of prose fiction than the wide range of texts sampled in the BNC. If the aim of the analysis is to identify typical stylistic features of the author, then the novel should more usefully be compared to prose fiction in English of his US contemporaries. If the aim is to identify features typical of only the one novel, it could be compared to the rest of the author's oeuvre.

A final problem relates to frequency and salience. The words which are perceived by the reader as the most significant in a text are not necessarily only those which occur more frequently than the reader would expect. There are other textual devices which can give a particular importance to a word in a text. The fact that a character in a play does not mention his wife's name can be striking and important, for example. Keywords can suggest ways to start to understand the topics or style of a text, and provide statistical evidence for certain textual phenomena, but cannot provide a list of all the interesting words, reveal all stylistic devices, and cannot explain a text.

15.4. Technical requirements

Keywords are calculated by comparing word frequency lists. One interesting aspect of the technique is that it is not necessary to have access to the full text or corpora used, only to the word frequency list. This has the advantage that researchers who don't have access to the corpus for whatever reason may still be able to access the wordlist and thus calculate keywords. It also means that if the researcher is employing a very large reference corpus, only the wordlist needs to be stored on the computer. So this technique can provide a means of using a large reference corpus where restrictions arise due to size, cost or legal issues. The researcher can conduct their analysis with access to only the wordlists, and keywords can be

identified and assessed for their significance purely on the basis of the comparison of lists of out-of-context words. However convenient this may be, it can also be a hindrance to thorough analysis of the actual usage of words in texts. Once a word has appeared in a list of keywords, it is likely to be useful, usually necessary, to look at the concordance lines from the corpus to understand more about whether it is part of a larger unit of words, whether it is occurring only in particular texts, and whether it is playing a particular role in the discourse.

16. Searching for larger units

16.1. Description

Up until now this chapter has focussed on searching for words, or for patterns, which are entered as the search term. Researchers are also interested in using computational methods to find out which sequences of more than one word occur frequently in a corpus. This section deals with using the computer to generate lists of the most frequently occurring sequences of words in a corpus.

16.2. Example

The following is a list of the 20 most frequently occurring sequences of four words in the British National Corpus

(BNC), as computed by the online resource Phrases in English (http://pie.usna.edu/explore.html). Note that # represents a number, and that certain sequences count as two words (e.g. â€œdon'tâ€) and others as one (e.g. â€œper centâ€) according to the BNC tokenisation.

I don't know
the end of the
at the end of
at the same time
I don't think
for the first time
on the other hand
between # and #
the rest of the
as a result of
in the case of
one of the most
per cent of the
the Secretary of State
by the end of
from # to #
is one of the
don't want to
to be able to
I don't want

Figure 16: 4-grams from the BNC

16.3. Analysis with this function

Research into multi-word expressions of various types is possible with these methods. Various types of recurring sequence of words are referred to as multi-word expressions, multi-word units, pre-fabricated units (also known as "pre-fabs"), n-grams, idioms, phrases and fixed and semi-fixed expressions. Generating lists of such sequences can start to give an insight into the recurrent phraseology of texts.

The analyst will often find it useful to search for patterns where one of the words is a wildcard, or specified by a part of speech, or a lemma, rather than a literal word form. For example â€˜preposition the noun of the' proves to be a very common pattern in English. Examining concordances of these variable multi-world expressions will help to show the varieties of lexical forms which are produced within and around them, and give some insights into how they are used.

16.4. Technical requirements

Calculating frequencies of multi-word expressions is a computationally complex task. Corpora and tools can be optimised for searching for multi-word expressions. Storing all sequences of words of a given length (e.g. all

two-, three- and four-word sequences) in a database can give a large improvement to access times and allows computation of the most frequent and significant sequences. It is also possible to make use of lemmatisation and wordclass tagging to find sequences with variable slots.

Such work requires considerable pre-processing of the corpus, and then the use of tools designed to access the indexed data.

There is also a complication with identifying the scope and granularity of multi-word expressions. Many patterns appear to be built up of smaller units, with some variable and some fixed elements. For example, some of the entries in the list above overlap, so that for example "I don't want" and "don't want to" appear as separate entries, but many if not most of these will be part of the longer sequence "I don't want to".

17. Searching and concordancing beyond the monolingual text corpus

This chapter has concentrated on the functions and tools which are commonly used with an English monolingual text corpus. Other types of corpus exist and are widely used. There are corpora of languages, and with different types of writing systems. With some languages it is useful

to focus on different forms of analysis from the ones suggested here. There are multilingual corpora, where the focus of research is on comparing languages; similarly there may be corpora which have different versions of a text in one language (e.g. different translations) which the user wishes to compare. And there are corpora which encode and store different modes, in the form of digital audio or video. Some corpora have streams which need to be aligned, such as audio, a phonetic transcription and an orthographic transcription. Similar forms of alignment are necessary for various forms of parallel corpus, such as translation corpora.

Many of the functions described can be used on corpora of other languages; some of the functions reflect basic techniques which have to implemented in different ways for other types of language; some are Also, some quite different techniques are necessary for other types of corpus.

Parallel corpora require structural markup to indicate the alignment of equivalents units. In the translation corpus, this alignment is typically done at the paragraph or sentence level. Specialist concordance programs exist for displaying both versions to the user, and may have further functionality, such as suggesting potential translation equivalents for the search term.

It is becoming increasingly possible for multimedia

corpora which capture in digital form the audio (and sometimes also video) of language events. Current technology for sound or image-based retrieval (e.g. "find me something which sounds like this...") is rarely successfully implemented in language analysis tools, and corpora still generally require the use of the text transcription or markup for retrieval. A user may search in the orthographic or phonological transcription for occurrences of a particular word (or other unit) and then listen to the linked audio stream as well for each of the concordance lines. The techniques for analysis of spoken data are likely to be subject to significant advances in coming years as more resources and tools become available.

While it is hoped that some of the principles outlined above will continue to be of relevance, the corpora of the future are likely to reflect a multilingual and multimedia environment in which distributed online access to resources and to analysis tools is increasingly the norm.

References

Barlow, M. (2004), Software for corpus access and analysis. In: Sinclair, J. (ed.), How to use Corpora in Language Teaching. Amsterdam: John Benjamins, 204-221.

Bennett, T./Grossberg, L./Morris, M. (eds)(2005), New Keywords: A Revised Vocabulary of Culture and Society. Oxford: Blackwell.

Bernot, E and Alarcón, E, Web Edition of the Index Thomisticus by Roberto Busa SJ and associates [Online]. Available from: http://www.corpusthomisticum.org/it/index.age [Accessed 2006-10-02]

Chomsky, N. (2002), Syntactic structures. Berlin : Mouton de Gruyter. [First published 1957.]

Culpeper, J. (2002), Computers, language and characterisation: An Analysis of six characters in Romeo and Juliet. In: Melander-Marttala, U./Ostman, C. and Kytö, M. (eds.), Conversation in Life and in Literature: Papers from the ASLA Symposium 15. Uppsala: Universitetstryckeriet, 11-30.

Hoey, M. (2005), Lexical priming: a new theory of words and language. London: Routledge.

Jakobson, R. (1960), â€˜Concluding Statement: Linguistics and Poetics'. In: Sebeok, T., Style in Language. Cambridge, Ma.: The M.I.T. Press, 350-377.

Lamy, M-N., and Klarskov Mortensen, H. J., Using concordance programs in the Modern Foreign Languages

classroom. In: Davies G. (ed.) Information and Communications Technology for Language Teachers (ICT4LT), Slough, Thames Valley University [Online]. Available from: http://www.ict4lt.org/en/en_mod2-4.htm [Accessed 2005-10-02]

Louw, W. (1993/2004). 'Irony in the text or insincerity in the writer? The diagnostic potential of semantic prosodies' in Baker, M./Francis, G./Tognini-Bonelli, E. (eds.),.Text and technology. Amsterdam: John Benjamins, 157-176. [Reprinted in Sampson, G./McCarthy, D. (eds.) (2004), Corpus linguistics: readings in a widening discipline. London: Continuum, 229-241.]

Rayson, P./Leech, G./Hodges, M. (1997), Social differentiation in the use of English vocabulary: some analyses of the conversational component of the British National Corpus. In: International Journal of Corpus Linguistics 2(1), 133-152.

Renouf, A./Kehoe, A./Banerjee, J. (in print), 'WebCorp: an integrated system for web text search'. In: Nesselhauf, C./Hundt, M. (eds.), Corpus Linguistics and the Web. Amsterdam: Rodopi.

de Saussure, F., (1983), Course in general linguistics. London : Duckworth [First published 1922] .

Scott, M./Tribble, C., (2006), Textual Patterns: keyword

and corpus analysis in language education. Amsterdam: John Benjamins.

Sinclair, J. (1991), Corpus, Concordance, Collocation. Oxford: Oxford University Press.

Sinclair, J. (2004), Trust the Text, London: Routledge.

Sinclair, J. 2005. The text and the corpus: basic principles. In: Wynne (2005), 1-16.

Tognini-Bonelli, E. (2001), Corpus Linguistics at Work. Amsterdam: John Benjamins.

Williams, R. (1988), Keywords. London: Fontana [first published 1976].

Wynne, Martin (ed.) (2005), Developing Linguistic Corpora. Oxford: Oxbow Books. Available from http://www.ahds.ac.uk/linguistic-corpora/ [Accessed 2006-10-02].

Acknowledgements

The following resources and tools were used in producing the examples for this chapter, or were examined to help with the taxonomy of functions: British National Corpus (BNC), Bank of English and the â€˜lookup' tool, Concapp,

Concordance, IMS Corpus Workbench, Monoconc, Paraconc, Phrases in English (PIE), Variation in English Words and Phrases (VIEW), Sara, Wordsmith Tools and Xaira.

There are many other very useful tools available. The fact that they are not included here should not be seen as a reflection on their potential usefulness.