ParaConc: A Concordancer for Parallel Texts
(Draft 3/03)

Michael Barlow
ParaConc is a new tool designed for linguists and translators who wish to work with translated texts. It can also be used by language teachers, cross-language retrieval experts, and other researchers interested in the analysis of multilingual texts.

The following group of people, who are all involved in translation studies in one way or another, have been using and commenting on various aspects of the program as it has been developed:

Mona Baker, Centre for Translation Studies, UMIST, Manchester
Pernilla Danielsson, Dept of English, University of Birmingham
Anna Mauranen, English/Philology, University of Tampere, Finland
Jennifer Pearson, Dublin City University, Eire
Raf Salkie, English, University of Brighton, England
Kim Wallmach, Dept. of Linguistics, University of South Africa

The first version of ParaConc was a rather rudimentary Macintosh program (based on Hypertalk) created in 1995. A few enthusiasts such as Pernilla Danielsson and Raf Salkie made some use of that program due to the fact that there was not much else available at that time. In the following year I started to design a Windows version, which led to the first beta version produced in 1996. These older Macintosh and Windows versions of the program can be downloaded from my webpage at www.ruf.rice.edu/~barlow. The present version is based not on the 1996 program, but on the code used in a monolingual concordance program (MonoConc Pro). An earlier version (9/00) of this latest incarnation is available at the website listed above.

This manual is based on Build 260 of ParaConc. The version you are using may be slightly different---the version number is provided in ABOUT PARACONC in INFO menu.
CONTENTS

1. Introduction  9
2. Installation  11
   2.1 Requirements  11
      2.1.1 Disk space  11
      2.1.2 Compatible versions of Windows  11
      2.1.3 RAM  11
3. Aligning files  13
4. Starting ParaConc  15
   4.1 Selecting a corpus  17
   4.2 Choosing a language  18
   4.3 Adding files  19
   4.4 Aligning the corpora  20
   4.5 Loading the corpus  24
   4.6 Displaying the corpus files  25
   4.7 Changing the corpus  26
5. Performing a basic concordance search  27
   5.1 Results in two languages  30
   5.2 Sorting the results  31
   5.3 Suggesting translations  32
   5.4 KWIC results II  33
   5.5 Hot words  35
      5.5.1 Paradigm option  37
   5.6 Context window  38
   5.7 Using wildcards  38
   5.8 Saving the concordance lines  39
   5.9 Printing the concordance results  40
6. Corpus frequency information  41
   6.1 Frequency options  41
   6.2 Stop list  42
   6.3 Saving and printing the frequency results  38
7 Collocates and Collocations 45
  7.1 Collocate frequencies 45
  7.2 Highlighting collocations 46
  7.3 Advanced collocations 47
  7.3 Saving and printing 48
8. Working with a corpus 49
  8.1 Tag settings 49
  8.2 Tags 52
  8.3 Part of Speech tags 54
  8.4 Meta-tags 54
  8.5 Using workspaces 55
  8.6 Saving a workspace 55
  8.7 Opening a workspace 56
9. Performing an advanced or parallel concordance search 57
  9.1 Text search 59
  9.2 Regular expression search 59
  9.3 Tag search 63
    9.3.1 Searching for words and tags 64
    9.3.2 Searching POS tags 65
    9.3.3 Searching for words 65
    9.3.4 Meta-tags 65
    9.3.1 Searching for punctuation 66
  9.4 Parallel search 66
10. Performing additional kinds of searches 69
  10.1 Context search I 69
  10.2 Context search II 69
  10.3 Heading search 71
  10.4 Append search 72
11. General search control and search options 73
  11.1 Ignore case of characters 73
  11.2 Skipping and equal characters 74
  11.3 Sentence mode 74
  11.4 Maximum number of hits 74
  11.5 Frequency of hits 74
  11.6 Changing context type and size 75
  11.7 What is a word? 75
11.8 Changing the wildcard characters 76
11.9 Headings 77
12. Displaying the results 79
  12.1 Navigation 79
  12.2 Change font and highlight font 79
  12.3 Word wrap 79
  12.4 Suppress words/tags 79
  12.5 Distribution 79
  12.6 Locating hits 81
  12.7 Changing the display: KWIC, Sentence, 81
13. The power of sorting 83
  13.1 Sorting 83
  13.2 Advanced sort 83
  13.3 User-defined sort 84
  13.4 Sorting tracked tags 85
15. Exiting the program 87
Index 89
1. INTRODUCTION

Translating from one language to another is like trying to get into someone else’s clothes. Sometimes the new clothes fit fairly well and at other times a considerable amount of retailoring must be done in order to make the person look presentable. A translator must continually push here and squeeze there and add a stitch or two to successfully complete the required transformation. When faced with the prospect of translating even simple phrases, the translator must try to match word choice, syntax, discourse, and register, along with subtle connotative aspects of meaning. There is no completely satisfactory translation—only the best compromise overall once the importance given to competing considerations is weighed.

Analyses of the choices made by individual translators practising their art is of interest in the field of translation studies (including translator training). Linguists (and also translators) tend to look at the general picture that emerges from the accumulation of choices made by individual translators, the focus being the comparison and contrast of two (or more) languages rather than the translation process itself. The analysis of translations provides the best evidence we have of the relationships holding between languages, a trove of information that has been neglected by linguists in the past, such comparisons will perhaps lead to a new, corpus-based approach to contrastive analysis.

Assuming that the data is potentially of great interest, there remains the problem of how to approach the analysis of translations. ParaConc is a simple software program that makes it easy to analyse translated texts. It is designed primarily as a search tool designed to work with translated (i.e., parallel) texts. The program when combined with a suitable set of texts can be thought of a full context, bilingual dictionary. There are differences, of course, in that the information is fairly raw, with the advantages and disadvantages that rawness entails.
2. INSTALLATION

Summary: Copy the file ParaConc.exe to hard disk.
There are currently no help files. Simply copy the ParaConc.exe file to a ParaConc folder created on the hard disk. It is also useful to also create folders for Texts, Results, and Workspaces.

2.1 Requirements
2.1.1 Disk space
The software files constituting ParaConc require a little over one megabyte of disk space on a hard disk drive. In the course of its operation, ParaConc also writes some small, temporary files to disk and creates or updates an ini file, which stores information such as settings and the search query histories. ParaConc also allows the user to save a workspace, that is, to save the current corpus links, alignment information and other settings. This workspace is saved as a file, which is typically around 10k in size, but the associated workspace folder may be anything from 2-20MB or more, depending on the extent of the tags, etc. and the number of options that are saved.

2.1.2 Compatible versions of Windows
ParaConc is a 32-bit program and hence must be run under Windows 95 or higher. Any of the higher Windows versions, including Windows 2000/Me/NT/XP, are acceptable platforms. ParaConc is not optimised for any particular system and is designed to run under a wide range of hardware/software configurations.

2.1.3 RAM
It is recommended that a computer with Windows 95 installed have 16 MB of RAM and a Windows 2000 (NT) or XP system should have a minimum of 32 MB of RAM. As always, the more RAM the better, but it is possible to run the program on a 100 MHz Windows 95 system with only 8 MB of RAM. The program is designed to work with whatever memory is available.
3. ALIGNING FILES

Summary: A first version of an alignment utility is included with ParaConc. (If you aim to align a number of texts, then you should contact Athelstan to obtain the latest version of the software.)

The successful searching and analysis of parallel texts depends on the presence of aligned text segments in each language text. The alignment, an indication of equivalent text segments in the two languages, typically uses the sentence unit as the basic alignment segment, although naturally such an alignment is not one in which each sentence of Language A is always aligned with a sentence of Language B since occasionally a sentence in Language A may be equivalent to two sentences in Language B, or perhaps absent from Language B altogether. The size of the aligned segments is not set by the software. It would be possible to work with paragraphs as the alignment unit, but then the results of a search will be more cumbersome because the translation of a word or phrase will be embedded within a large amount of text, which is especially difficult in cases in which the language is not well-known.

The alignment process is crucial for the successful operation of ParaConc. When the program searches through the source text, the only information the program has about the links between the different languages is the alignment. No use is made of bilingual dictionaries or of any kind of language-particular information.

If the parallel texts are pre-aligned, then it will simply be necessary to indicate the way in which the alignment is marked, as described in Section 4.3.

If the translated texts are not aligned, then the alignment will be carried out by finding equivalent segments first at the level of headings (perhaps with just a single heading for the whole file), then at the paragraph level, and finally at the sentence level. Details of the alignment process are given in Section 4.4.
4. STARTING PARACONC

Summary: Double-click on ParaConc to open the program. HELP will eventually be available in the INFO menu.

To start the program, double-click on the file ParaConc. Once the program is open, a simple screen appears, as shown below in Figure 1. This initial screen looks rather bare, containing only a blank window and two menu items: FILE and INFO.

![Initial screen](image)

Figure 1: Initial screen

Note that the information field in the lower left corner states: “No files loaded.”

QUICK LOOK

User-friendliness is an important aspect of the design of ParaConc and one measure of a well-designed program is the ease with which a user can
perform a simple search. Because we are dealing with multiple languages and multiple windows, some complexity is unavoidable, but it is nevertheless possible to load and search corpus files quickly and easily. To illustrate this and to get an overview of the operation of ParaConc, you can follow the instructions below to carry out a simple parallel text search. Performing a parallel search is somewhat complex and if you would rather follow a more methodical, step-by-step introduction, then you should skip to the next part of this section on page 16.

1. If the LOAD CORPUS FILES dialogue box is not open, choose LOAD CORPUS FILE(S) from the FILE menu. Set the number of parallel texts to 2. Select the language of each set of texts by clicking on the language textbox.

2. Select one or more text files for each language using the ADD (and REMOVE?) buttons.

3. Click on OPEN. When the files are being loaded, progress bars will appear as each file is processed.

4. Choose SEARCH from the SEARCH menu.

5. Choose a language to work with and enter a search string (a word or phrase). Click OK (or press return).

6. Press return (or click Cancel) if you wish to halt the search at any time.

7. Examine the results in the concordance window. The hits will be easier to view if you maximise the window. Click on a line that is of interest and you will see that the corresponding line or segment in the second half of the window will be highlighted.

8. Try sorting the concordance lines, using commands in the SORT menu to look for different patterns in the data.

9. Try different display options by using CONTEXT TYPE and SHOW LINE NUMBERS. Delete any unwanted examples by selecting the lines and pressing Ctrl-D.

10. Possible translations (in the lower text window) can be identified (or highlighted) in a couple of ways. You can position the cursor in the lower window, click on the right mouse button, and select SEARCH QUERY from the pop-up menu. Enter the possible translation and press return. Choose CONTEXT TYPE from the DISPLAY menu and choose WORDS. You will then get a KWIC format in the lower window, which can be sorted via the SORT menu. Thus the display/sort menus control whichever text is selected. This is important: the menu commands are sensitive to whichever text or language is currently selected.

**no hits?**

If, contrary to your expectations, the search returned no finds or “hits,” then it is likely to be either due to a problem with the specification of the search string because the word or phrase you looked for is not present in the text.

**FILE and INFO**

Having given a brief overview, let us return to the beginning and work more methodically through the different aspects of the program.

Once the program is open, a simple screen appears which contains a blank window and two menus: FILE and INFO. Let us deal with INFO first. This menu
is always present; it provides access to HELP (not yet implemented) and to some basic information about ParaConc, as well as some contact information for Athelstan.

Selecting the FILE menu reveals several commands, one of which is the LOAD CORPUS FILE(s) option (see Figure 2), but before making a corpus available for searching you may want to look at the discussion of TAG SETTINGS in Section 8.1.

4.1 Selecting a corpus

Let us examine the options for loading a corpus, that is, making one or more text files available for processing by ParaConc. If the files and configuration were saved as a workspace from a previous session, then choose OPEN WORKSPACE from the FILE menu. Otherwise the corpus can be made accessible by choosing LOAD CORPUS FILE(s) from the FILE menu (Figure 2) or by issuing the CTRL-L command.

![Figure 2. File menu](image)

When the LOAD CORPUS FILE(s) command is given, a dialogue box as shown in Figure 3 appears, enabling particular parallel files to be loaded. The heading PARALLEL TEXTS at the top of the dialogue box is followed by a number in the range 2-4. We will assume for now that we are dealing with texts in two languages and will choose 2—although the procedure is essentially the same if you are dealing with four languages.
ParaConc does not know anything about the special text formatting conventions of the different word-processing programs, and so files produced using a word-processor must be saved as text-only ANSI files, rather than as the usual application-specific file type (such as a Word document), which contain boldface and other formatting information tied to the particular word-processor. Whether or not your files actually contain formatting (boldface etc.), they must be saved as text-only files before they can be loaded and searched by the program.

It is a good idea to make a copy of your files before creating the text-only versions. The details of producing the correct kind of file can be found in the manual for your word-processing program under the heading of ANSI or text-only files.

If you are working with English texts, you can use ASCII files, but if you have non-English ASCII files containing accents, etc., then attempts to load them into ParaConc will cause the accented characters to appear in garbled form. You must transform the files into ANSI text format by opening them using a Windows-based word-processor and then saving a copy of the files as Windows (ANSI) text.

4.2 Choosing a Language

It is necessary to choose an appropriate language for each set of texts. The available languages can be seen by clicking on the drop-down arrow to the right of the language name. Apart from providing a label for each set of
texts, choosing a language allows special character and accents to be entered easily, and defines the appropriate alphabetical (sorting) order. (The range of languages displayed will depend on the version and configuration of Windows installed on your machine.) If the language you want is not present in the list, you should simply select the font that you want to use.

4.3 Adding Files

When the ADD button is selected, the typical Windows dialogue box appears in which the contents of the current directory are displayed. Alternative directories/drives can be selected via the text box at the top of the dialogue box.

Once the appropriate file name appears, use the mouse to click on the filename to select it. To select several files, use the shift and cursor keys, or CTRL-A (or to select non-adjacent files, use Ctrl and shift and click). The result should be similar to the example in Figure 4.

![Figure 4. English and French files are added](image)

The files for each language should be listed in appropriate (parallel) order. If they are out of order, it is possible to “click and drag” them so that they are in their proper place.

The text segments in these files are already aligned and so all that is needed is to select the appropriate alignment indicator. The text segments in these files are delimited by means of a carriage return (paragraph mark) and so we choose the NEW LINE DELIMITED SEGMENTS setting for ALIGN FORMAT.
Symbols other than a carriage return may be used and in this case the appropriate option is **DELPITED SEGMENTS**. To indicate the alignment symbol used, click on options and enter the symbol in the text box.

**Odd results**

If you find that after carrying out a search that the lower window contains a single sentence repeated over and over, then it is likely to be a problem arising out of the incorrect setting of the alignment scheme or alignment delimiter.

The different alignment options are shown in Figure 5.

![Alignment options](image)

**Figure 5. Alignment options**

An alternative scheme available in the current version is based on the use of start/stop tags. Thus aligned segments may be indicated by the segment start and end tags `<seg>` and `</seg>`. Alternatively, a COCOA format may be used in which the start tag contains alignment information internally in the form of an attribute, as in `<seg ID=245>`.

The particulars of this type of configuration can be specified once the **SEGMENT START/STOP MARKERS** alignment option is selected.

### 4.4 Aligning the corpora

If the corpus files are not pre-aligned, select the **NOT ALIGNED** option. If the files use headings or indicate paragraph boundaries in a particular way, this information will be needed for the alignment. In order to enter this formatting information, select **FORMAT** (for each language). The **FORMAT** buttons can be seen in Figs 4 and 5.

The content of the format dialogue box is shown in Figure 6. The information in the dialogue box is divided into three sections: Headings, Paragraphs, and Sentences, allowing the formatting of each type of text unit
to be specified. Thus, in the paragraph section the user can choose from the following:

- New Line delimited
- HTML/SGML Markers
- Two New Lines Delimited
- Indented

![Text format](image)

**Figure 6: Specifying the format of headings, paragraphs and sentences**

**Headings**

The formatting options for Headings are more complex because the range of options is greater. Some texts may use HTML/SGML tags to mark headings and in this case the form of the tags can be entered. If both the Start tag and End tag are empty the whole text is taken to consist of a single heading. The sequence `%d` used to specify HTML/SGML Headings matches any number and so if H%d and /H%d are entered for the Start and End tags, then the program will recognize `<H1> ... </H1>`, `<H2> .... </H2>` etc.

Another option is Pattern. The program looks for this specified pattern at the beginning of every paragraph and if it is found, then a new heading is started. The sequence `%d` means any number; `%r` means any roman number.

The third option is Regular expression: The program looks for the regular expression at the beginning of every paragraph and, again, if it is found then a new heading is started.

**Sentences**

The two options that the user can specify for the recognition of Sentences are HTML/SGML or Automatic Recognition.

Note that alignment can only be carried out on two files at a time.

If the alignment algorithm is successful, then the files will be processed and the menu bar will be expanded to include SEARCH and FREQUENCY commands. However, it is a good idea to check the alignment, whether or not the
alignment process seems to be successful. To do this, choose \texttt{VIEW CORPUS FILES} from the \texttt{FILE} menu, select a pair of files and click on \texttt{ALIGNMENT}, as shown in Figure 7.

![Select Files to View](image1.png)

\textit{Figure 7: Viewing alignment}

The alignment of headings or paragraphs or sentences can be viewed by selecting the appropriate command from the \texttt{ALIGNMENT} menu. (See Figure 8.)

![Alignment menu and display of aligned units](image2.png)

\textit{Figure 8: Alignment menu and display of aligned units}
In order to make the alignment as clear as possible, the alignment units are indicated by horizontal lines and the sentences within a larger unit follow a colour sequence, as can be seen (perhaps) from Figure 9.

![Figure 9: Merging and splitting alignment units](image)

The left mouse button can be used to insert a marker and clicking on the right mouse button brings up a menu of options which allow the splitting or merger of alignment units.

Because some items such as dates or cognates will usually occur in equivalent alignment segments in translated texts, this fact can be used to highlight potential misalignments. The information is not used in the alignment algorithms themselves; it is only used to help the user check the alignment. These equivalences are entered as ALIGNMENT MARKERS SETTINGS, as seen in Figure 10.

![Figure 10: Specifying equivalences: dates or cognates](image)
**Important note:** If the texts are not pre-aligned and alignment is done within the program, then that alignment information must be saved. This can be accomplished by saving the current files/settings as a workspace. To do this, use the `SAVE WORKSPACE` command.

### 4.5 Loading the corpus

Clicking on `OPEN` (or pressing Return) loads the selected files into *ParaConc*, making them available for searching. As discussed below in Section 8.1, the program scans the files looking for information about tags and alignment. When the scanning process is completed, the word count for the two corpora will appear at the lower right of the window. The processing of the files may take some time and it is advisable to use the workspace option (described in Section 8.5) to minimise the number of times that this operation has to be performed.

There is no real limit to the size of the corpus loaded. The corpus files appear to be loaded in the program ready for searching, but in reality *ParaConc* does not load the whole text, but switches chunks of text in and out of memory as needed, which means that the program should be able to handle any size of text.

Once a corpus is loaded, some new menu items related to the analysis and display of the text appear on the menu bar. These are `FILE`, `SEARCH`, `FREQUENCY`, and `INFO`. In addition, looking at the screen in Figure 5 we see information in the lower left corner relating to the number of the files loaded and in the lower right corner a word count for the two corpora. If some settings, such as word delimiters are changed, it will be necessary to run `SCAN CORPUS` in the `FILE` menu to re-process the files.
Before initiating a text search, we should note that despite the fact that we have one or more files loaded and a window showing the text of the first of these files, the choices made so far can be changed very easily.

### 4.6 Displaying the corpus files

The loaded corpora are not displayed on screen, but any of the corpus files can be viewed if desired. Selecting **LOAD CORPUS FILE(s)** allows the individual files to be viewed.

The main method viewing files is to select the command **VIEW CORPUS FILE(s)** (Ctrl-V) from the **FILE** menu. A dialogue box containing the file system is displayed. This option allows you to examine parallel files. If the **ALIGNMENT** button is clicked, then the alignment menu becomes available, as described above. If the **SHOW** button is selected, then the parallel files are displayed side by side and a new menu **CORPUS TEXT** appears. The commands in **CORPUS TEXT** are **CHANGE FONT**, **SUPPRESS**, **WORD WRAP**, **EQUAL WIDTHS** and **SYNCHRONIZE TEXTS**. The first three commands control the display of the text. The vertical bar separating the files can be adjusted to allow more space for the display of one of the languages and the **EQUAL WIDTHS** command simply readjusts the display to give equal horizontal space to each language. If **SYNCHRONIZE TEXTS** is selected, scrolling through one file will cause the parallel files to scroll.
4.7 Changing the corpus

unload  In addition to manipulating the form of the corpus files, it is possible to alter the actual composition of the corpus. For instance, all the files may be removed by choosing UNLOAD CORPUS from the FILE menu. This returns the program to its initial state, with only the FILE and INFO menus available. All other windows and menus are closed.

remove file The selective removal of one or more files is accomplished by selecting LOAD CORPUS FILE(s) from the FILE menu, selecting one or more files and clicking on the REMOVE button. Additional files can be added by selecting LOAD CORPUS FILE(s) and clicking the ADD button.
5. PERFORMING A BASIC CONCORDANCE SEARCH

Summary: Select SEARCH from the SEARCH menu and enter a search string. The parameters controlling the search are listed in SEARCH OPTIONS. The ADVANCED SEARCH and PARALLEL SEARCH commands are covered in Chapter 9.

A monolingual concordancer such as ParaConc’s sister program MonoConc is basically a search program that looks for patterns in texts based on a search query. Simple as this sounds, it can lead to sophisticated analyses of lexical, grammatical and textual structure by allowing the researcher to easily find rare instances of words or strings; find strings in the context of other strings, e.g., the instances of economy occurring after <title> and before </title>; and generally trawl through texts looking for particular language patterns.

Of course, here we are looking at a multilingual concordancer, which adds an important new dimension to the search process. Let us illustrate this with a simple search for the English word head in an English-French parallel corpus. To initiate the search, we select SEARCH from the SEARCH menu, or enter CTRL-S, (shown in Figure 12).

![Figure 12: Search menu](image)
search

In the text box at the top of the dialogue box that appears (Figure 13) type in the search term **head** and click on OK (or press enter). Some example search queries that can be found below the search box serving as reminders of the format of simple text searches and the use of wildcards. Notice too that a search can be performed on either language. This is controlled by the choice under **LANGUAGE** at the top of the search dialogue box.

![Text Search](image)

**Figure 13: Performing a simple text search**

Note: The parameters of the search are determined by the settings in force in **GENERAL SEARCH CONTROL** in the **ADVANCED SEARCH** dialogue box and in **SEARCH OPTIONS** as illustrated in Figure 14. These latter settings cover such things as whether a hyphen or an apostrophe or brackets are treated as word boundaries. (See Chapter 11 for a complete description.)
Once the search query has been entered, the program starts to work though the text files looking for instances of the search string. If a search does not behave in the way that you expect, then this is probably because (i) the corpus is not quite what you thought it was; (ii) the search term was mistyped; (iii) some setting (such as the list of word delimiters) is not what you expected; or (iv) you are searching under the wrong language.

Let us now look in some detail at what happens during the search process. In our example, the program works its way through the text looking for the word head. With such a common word, the results should come in at a fairly rapid clip. Two progress bars track the progression of the search routine as it scans (i) individual files and (ii) the corpus as a whole.

To stop a search, but retain the hits already found, simply press return or click Cancel. (If pressing return has no effect, simply click on the progress box to reactivate the window.)

The progress bars remain visible as long as the search is still in progress. The results of the search appear in a backgrounded window, which we will refer to as the concordance results window (Figure 15). This window might seem a little daunting at first due to the arrangement of blocks of text which differ markedly from the normal text layout found in books and documents. Note that the text results in this example are divided into two (separated by an adjustable bar). In the top part of the window, each instance of the search word that the program finds is copied along with a
preceding and following context. Typically, the search word is centred and highlighted so that the instances of the search word line up, as shown in Figure 15. This format is commonly referred to as a KWIC (Key Word In Context) format.

![KWIC Format Image]

Words surrounding the keyword may also be highlighted. This highlighting indicates potential collocates of the search word. See Section 7.2 for a discussion of this feature.

5.1 Results in Two Languages

The lower part of the window contains the French sentences (or, more strictly, text segments) that are aligned with the hits displayed in the top window. This process is not at all magical; it is a consequence of the alignment process. Thus if the first instance of head occurred in segment 342 of the English text, then the program simply throws segment 342 of the French text into the lower window, and this process is repeated for all instances of head. It is worth thinking about this for a moment in order to fully understand how the parallel concordancer works and how important it is to interpret the results with caution. Thus we are working with (assumed) equivalence at the sentence or segment level, and within this equivalence, we can use the program to help determine word or phrase equivalents, but there is no guarantee that an English word highlighted in the KWIC display will be translated in the equivalent French sentence. The program is simply indicating the sentence or text segment that is equivalent on the basis of alignment to the English sentence that contains the keyword.
5.2 Sorting the Results

Let’s follow this example further. We could simply click on a particular instance of *head* and examine the equivalent French segment to check on the translation. Typically, the next step is to sort the results so that similar instances line up together. Thus if we are interested in the translation of *shower head*, we can sort the results in a way that brings all instances of *shower head* together. We need to sort the instances so that they are in alphabetical order of the word preceding the search term. The advantage of performing this ‘left sort’ is that the modifiers (adjectives) of *head* that are the same will occur together. The easiest way to achieve this ordering is to select 1ST LEFT, 1ST RIGHT, from the SORT menu (Figure 16).

![Figure 16: The Sort menu](image)

The program then immediately rearranges the concordance lines in a way that typically groups the results in revealing ways. If you are interested in one particular example of *head* in English, we can click on the appropriate line and both the English and French lines will be highlighted, as shown in Figure 17. (Notice also the sorted lines in the upper text window.) You can glance at the words directly preceding *head* to check that the intended re-ordering has taken place. Alternatively, you can check the description of the current sort that appears at the bottom of the results window, to the right of the information on the number of hits. Having ordered the concordance lines in this way, it is very easy to see which words occur with *head* and with what frequency. In ParaConc, specifying a sorting order typically involves selecting a primary sort position (e.g., 1st Left) and a secondary sort position (e.g., 1st Right). See Chapter 13 for a detailed explanation of sorting.
If you scroll fairly quickly through the concordance results, you will discover that the visual patterning created by several identical words surrounding the search word will be striking enough to catch your eye. It is not necessary to focus on the results line by line; you can scan the output quite rapidly. You can also click examples of interest and examine the corresponding sentence in the French text. Conversely, it is also possible to click on a French sentence and examine the English sentence. Notice that as you alternately click on the lower and upper windows, both the language indicator and the filename in the lower left of the screen change. Thus it is possible to make either the English text active or the French text active. This is important since, as we will see, we will want to use the menu items to manipulate the French results as well as the English results.

5.3 Suggesting Translations
It can be difficult to locate the position of possible French translations of head within each French segment. To alleviate this, we can highlight suggested translations for English head by positioning the cursor in the lower French results window and clicking on the right mouse button. The menu shown in Figure 18 pops up and the user can select SEARCH QUERY which gives access to the usual search commands. You can enter a possible translation such as tête.
The program then simply highlights all instances of tête in the French results window, as illustrated in Figure 19.

**Figure 19: Highlighting possible translations**

### 5.4 KWIC Results II

**KWIC**

We can now change the context for the French results so that the results in the lower window are transformed into a KWIC layout (at least for those segments containing tête.) First, we make sure that the lower window is
active. (Look at the lower left of the window to check that a French file is selected or simply click within the French text window.) Next we choose CONTEXT TYPE from the DISPLAY menu and select WORDS. Finally, we rearrange the lines to bring those segments containing tête together at the top of the French results window. To achieve this, choose SORT and sort the lines by search word, and 1st left. The sorting procedure will now rearrange the results in lower window since the sort commands are applied to whichever window is active. The two text windows then appear as shown in Figure 20. Because this series of actions is a little cumbersome the same results can be accomplished using a shortcut. Simply click on the right mouse button when the cursor is within the French results window and select KWIC from the menu.

Naturally, only those words in the French text that have been selected and highlighted can be displayed in this way. By sorting on the search word, all the KWIC lines are grouped together at the top of the text window; the residue, which may be quite large, can be found by scrolling through towards the bottom of the window.

This is a revealing display, but we have to avoid being misled by the dual KWIC displays. There is no guarantee that for any particular line, the instance of tête is the translation of head. It could be accidental that tête is found in the French segment that corresponds to the English segment containing head.

The idea behind this feature of ParaConc is to let the user move from English to French and back again, massaging the results to get a sense of the connections between the two languages.

Let me remind you once more that the DISPLAY and SORT menus are sensitive to which of the windows is active. The information panels at the bottom of the screen will provide information on the current state of the program.
5.5 Hot Words

In the previous section, we used SEARCH QUERY to locate possible translations in the second window. In this section we will look at a utility in which possible translations and other associated words (collocates) are suggested by the program itself. We will call these words: hot words. First we position the cursor in the lower (French) half of the results window and click using the right mouse button. If we used SEARCH QUERY earlier, we need to select CLEAR SEARCH QUERY and then we can choose HOT WORDS, which brings up a dialogue box containing a ranked list of hot words. The ranked list of candidates for hot words are displayed as shown in Figure 21. The figures can be taken as an approximate guide to the relative strength (hotness) of the different words.
Some or all the words can be selected. (Use CTRL click to select non-contiguous words.) When the list of selected words is complete, click on OK. The chosen words will be highlighted in the results and can again be sorted. A new sorted list is shown in Figure 22.

Figure 22: Highlighted and Sorted Hot Words
5.5.1 Paradigm Option

One problem of using calculations based on word comparison to provide a list of hot words arises from the fact that words which are very similar, such as tête and têtes, are treated as completely different words. We need some way to boost the ranking of words whose forms are similar. To do this we select OPTIONS in Hot Words and check Paradigm option, as shown in Figure 23. There are various settings that control the calculations of Hot Words. The settings in Global Options relate to the minimum number of hits required for a word to qualify as a Hot Word and the MAX CANDIDATES figure refers to the number of word clusters. Thus tête and têtes count as one candidate. The Language Options control the range of words considered and the strictness of the matching. Some experimentation should lead to the optimal settings for a particular language.

![Figure 23: Paradigm Option in Hot Words](image)

The results of choosing the paradigm option may compensate to some extent, in some languages, for the lack of a morphological analyser. The results of selecting the paradigm option in Figure 24 can be compared with those in Figure 21 above.
5.6 Context window
We have seen that the results of a search are displayed in KWIC format in the concordance results window. To see a larger context (i.e., a section of preceding and following text) for any concordance line, simply select the line in the results window by double-clicking on it. This larger context is then displayed in a new context window.

5.7 Using wildcards
Let us consider again our search for head. To broaden the search, we can make use of wildcard characters. (We will discuss more precise and sophisticated alternatives in Section 9.2.)

* wildcard The first wildcard character is the asterisk *. This wildcard in a search string will match zero or more characters and so we can use it to formulate a more complex search string with a wider range of potential matches than would be possible using a fully specified search string. Thus a search for head* will find head, heads, header, heading, etc.

delete hits Searching for the head* string produces numerous hits. Since we are not sure what words are found by this search, we need to sort the concordance lines with respect to the search word. This means that the lines are sorted according to the alphabetical order of all the words that fit the template head*. To accomplish this, choose SEARCH TERM, 1ST RIGHT from the SORT menu. Once we have done this, we can scroll through the concordance window, checking the search words. We may notice words that are not of
interest and we can press CTRL-D to delete those particular lines (or select DELETE ITEM from the DISPLAY menu).

? wildcard

The wildcard character ? stands for a single alphanumeric character. In the unlikely event that we wanted to search for head or heed, we could find both words by searching for he?d.

% wildcard

The special character % represents zero or one character. Thus the search string head% would match head and heads.

* alone

It is a feature of ParaConc that it does not require there to be an alphanumeric character within the search string. You might like to think about the results of a search for * by itself.

If you want to search for question mark (?) as a literal character, you will have to substitute a different character for the wildcard in the SEARCH OPTIONS dialogue box. (See Section 11.8)

Below is a summary of the wildcard characters relevant for a TEXT SEARCH:

* matches zero or more alphanumeric characters

% matches zero or one character

? matches exactly one character

5.8 Saving the concordance lines

To save the contents of the concordance window, select SAVE AS FILE from the CONCORDANCE menu. A dialogue box appears and the name of the results file can be entered.

show names

The default option is to save the concordance results in the file. However, there are a variety of options concerning other information that can be saved along with the concordance lines. Checking the box FILE NAMES ON EACH LINE causes a preceding filename to be saved with each concordance line. The filename indicates the file in the corpus that the concordance line is associated with. Other options are SEQUENTIAL NUMBER IN EACH LINE, CONCEAL HITS, LINE NUMBERS, PAGE NUMBER, NORMAL TAGS, CONCEAL COLLOCATES, SEGMENTS, RELATIVE LINES, DEFINED SORT LABELS. (If PAGE NUMBER is selected, then RELATIVE LINES may be selected if you want to see the line number of the hit relative to the page rather than the line number in the file as a whole.)

The SUPPRESS option allows you to suppress words or tags.

The SAVE AS HTML option allows the results to be saved in HTML format for display on the web.

To save the concordance lines, enter a suitably descriptive file name and press the enter key (or click on OK).

For further manipulation of the text, you may need to load the results file into a word processing program such as Word.
If only a few examples are needed, it may be more convenient to use CTRL-C to copy the required text directly from within ParaConc to the clipboard and then paste it into your document.

5.9 Printing the concordance results
The concordance lines can be printed by entering CTRL-P or selecting PRINT from the CONCORDANCE menu. The appropriate concordance window must be active for printing to occur.
Summary: To create a word list, select FREQUENCY ORDER or ALPHABETICAL ORDER from the FREQUENCY menu. The parameters controlling the results are listed in FREQUENCY OPTIONS.

It is sometimes instructive to destroy the linear structure of a text completely by breaking it down into individual words which can be counted. The result is a word list, which is usually accompanied by a frequency count for each word (and the frequency expressed as a percentage of the whole corpus).

To find out the distribution of words in the entire corpus, choose FREQUENCY ORDER or ALPHABETICAL ORDER from the FREQUENCY menu. You should also indicate whether you want to obtain the frequency information for one particular language corpus or for all the language corpora. See Figure 25.

Figure 25: Corpus frequency commands

6.1 Frequency options
Choosing FREQUENCY OPTIONS allows the frequency list to be tailored to fit particular requirements. It is possible to limit the data presented in three main ways, as can be seen from the FREQUENCY AND COLLOCATION OPTIONS dialogue box, shown in Figure 26. First, we can set the lower frequency boundary for the corpus frequency list (MINIMUM FREQUENCY). And, if
desired, a maximum frequency limit can also be selected (MAXIMUM FREQUENCY); this leads to the exclusion of words occurring more often than the set threshold. (The setting of 0 is used to indicate no upper boundary.). A second parameter is the maximum number of lines in the frequency list (MAXIMUM LINES), which means that it is a simple matter to find, for example, the twenty most frequent words or the hundred most frequent words in a corpus.

With these parameters, it is possible to examine various frequency lists for a corpus. The most common uses are to list the most frequent words or, alternatively, to display those words that occur just once. The two checkboxes in the dialogue box (IGNORE CASE OF LETTERS and SKIP TAGS) control case-sensitivity and tag-sensitivity in frequency calculations.

6.2 Stop list
Because grammatical or function words such as the are both of high frequency and generally not very interesting, it is often desirable to omit them and concentrate on words of interest which might otherwise be masked by the more frequent forms.

There are two ways to exclude a set of words from consideration. One option is to create and load a text-only file containing the words to be
excluded from the frequency listings—a stop list. The second is to add words to the stop list from within the program. Both these functions are accessed via FREQUENCY OPTIONS. To exclude a particular set of words from the frequency counts, select CONTENT WORDS ONLY, then choose EDIT, and type in the words to be omitted. Alternatively, you can load a file containing the words you wish to exclude from the frequency counts. (The words should be listed one to a line in the file.)

The various frequency settings can be individually tailored for each of the languages selected.

6.3 Saving and printing the frequency results
The output from the corpus frequency analyses can be saved to a file by choosing SAVE AS FILE from the FREQUENCY menu. The frequency file that is active will be saved to a text file. The nature of the saved file is controlled by the options selected in the dialogue box. See Figure 28.

![Figure 28: Save options](image)

The frequency results can be printed using CTRL-P or PRINT.
7. COLLOCATES AND COLLOCATIONS

Summary: The collocates of a search word can be calculated by the program. It is also possible to use the ADVANCED COLLOCATION option to produce a frequency list of collocations larger than two words.

7.1 Collocate frequencies

ParaConc furnishes a variety of frequency statistics, but the two main kinds are corpus frequency and collocate frequency. The corpus frequency commands create a word list for the whole corpus, as described in the previous chapter. Choosing COLLOCATE FREQUENCY DATA from the FREQUENCY menu displays the collocates of the search term ranked in terms of frequency. A more subtle rendering of collocate frequency information is the highlighting of collocates in the results window with the brightness of the colour indicating frequency information.

The collocates of a word are its frequent neighbouring words. In ParaConc, the collocate frequency calculations are tied to a particular search word and so the frequency menu only appears once a search has been performed. Typically, the collocation data produced by the COLLOCATE FREQUENCY DATA command (Figure 29) is organised in four columns, with one column for each position surrounding the keyword: 2nd left, 1st left, 1st right and 2nd right. (Thus 1st left refers to the word before the search term and 1st right refers to the word following the search term.) The columns show the collocates in descending order of frequency. This describes the collocate frequencies for a span 2L-2R. The size of the span and other information is controlled by the settings in FREQUENCY OPTIONS, as shown above in Figure 27. The collocate frequency span can range from 1L-1R to 4L-4R.

Returning to the earlier example of a search for head in English (and an associated search for tête in French), we can see the words commonly occurring with the search word by scrolling through the concordance lines. The program then calculates, for the language chosen the frequency of collocates surrounding the search term for a particular span.

Note: The COLLOCATE FREQUENCY DATA command can only be chosen when a concordance results window is active. Note also that if you want to force a recalculation of the collocate frequencies, you must first close the COLLOCATE FREQUENCY DATA window. Thus, you must be careful about choosing the COLLOCATE FREQUENCY DATA command after having deleted some concordance lines. If no collocate window tied to the concordance is open, then the collocates will be (re-)calculated and everything will be fine, but if a collocate window already exists, then the COLLOCATE FREQUENCY DATA
command will simply invoke the existing collocates window, the contents of which will not reflect the current state of the concordance results.

Figure 29: Collocate frequency data for different languages

7.2 Highlighting collocations
The software highlights potential collocates of the search word within the concordance results window.

Figure 30: Collocates of head highlighted in red

The span of the highlighted collocates is set in FREQUENCY OPTIONS (Figure 27), as is the depth of ranking of collocates that are to be highlighted. For example, the 5 most frequent collocates for each position in the span might
be highlighted. It is also possible to use the stoplist in FREQUENCY OPTIONS to exclude certain words.

### 7.3 Advanced collocations

One disadvantage of the simple collocate frequency table is that it is not possible to gauge the frequency of collocations consisting of three or more words. To calculate the frequency of three word collocations, it is necessary to choose ADVANCED COLLOCATION from the FREQUENCY menu and select one or more languages. The ADVANCED COLLOCATION dialogue box (like ADVANCED SORT) is divided into two sections. The top part of the dialogue box associated with ADVANCED COLLOCATION allows the user to choose from up to three word positions.

![Advanced Collocation - English (United States)](image)

*Figure 31: Settings in Advanced Collocation*

In Figure 31 the settings are configured to have the program calculate the most frequent collocations based on the positions 2L, 1L, Search word. The ADVANCED COLLOCATION option can also be used to count the different components making up the search term. The main collocate frequency table always ignores the search term, which means that if you use wildcard characters or tags in the search query and capture a variety of words as hits, then you need a way to count the different words. You can do this by using ADVANCED COLLOCATION and selecting SEARCH TERM in one column and setting the other two columns to NONE. The ADVANCED COLLOCATION routine will then calculate the frequency of each of the search term forms.

When checked, the Customised Collocation option allows the user to enter word positions (1L,0,1R) defining the collocation into a text box. The user can list positions separated by commas or enter a range of words such as 2L-2R. Note that if this option is checked, it is not possible to select the drop-down menus used to define collocations by position.
The final component of **ADVANCED COLLOCATION** is **SPAN**, as illustrated in the dialogue box in Figure 32.

![Figure 32: Counting 5-word Collocations](image)

The span option will produce a frequency-ordered list of 5-word collocations containing the search term.

The calculation and display of collocations is guided by the settings in **FREQUENCY AND COLLOCATION** options. (See Figure 27.) This dialogue box is invoked by selecting **FREQUENCY OPTIONS** from the **FREQUENCY** menu.

### 7.4 Saving and printing the frequency results

The output from the frequency analyses can be saved to a file by choosing **SAVE AS FILE** from the **FREQUENCY** menu. The frequency file that is active—the collocate frequency or the corpus frequency—will be saved to a text file. The frequency results can be printed using **CTRL-P** or **PRINT**. Once again, it is the frequency file that is active that will be printed.
8. WORKING WITH A CORPUS

Summary: Check the configurations in TAG SETTINGS in the FILE menu and then choose LOAD CORPUS FILE(s). Saving a workspace avoids the need to reload (and realign) corpus files or re-process the corpus.

In order for the software to take advantage of the information encoded in texts it is often preferable to enter descriptions of the form of annotations before initiating the loading of files. The following description might seem to be somewhat involved, but once the settings are entered, then the processing and display of the corpora and the search results will be much clearer. In addition, the use of “workspaces,” which are described in Section 8.5, is a good way to ensure that the information about the annotations used in the corpus only has to be entered once.

8.1 Tag Settings

More often than not, text files contain, in addition to the text or content itself, a set of tags or annotations or mark-up that provides extra explicit information about the components of the text, such as the division into title and body, or about the source of the text. In some cases the mark-up is minimal, as is the case with the demo text files included with ParaConc.

In such corpora there is a simple but very important distinction between the strings of letters and symbols that identify the speaker and the strings of letters that represent the words actually spoken. If this distinction is made in the text, then it is best if the program treats the text per se and the tags or mark-up as separate kinds of objects. There are some complexities involved here but to give a simple example it is clear that generally the calculation of frequency information should be based solely on the contents of the text should be used, not the words comprising the tags. (Note: if you wanted to include the words in tags, you could do it by purposely not indicating the structure of tags in TAG SETTINGS or by deselecting SKIP TAGS in FREQUENCY OPTIONS.)

Increasingly, corpora are highly tagged with information representing the source and structure of texts. For example, the BNC Sampler CD-ROM contains texts that look like the following.

```
tagging <text decls="CN004 HN001 QN000 SN002" complete=Y org=COMPO> <div1 n="08-NOV-89 edition, page " complete=Y org=SEQ>
```
Lebanon leader builds cabinet
By Reuter in Beirut

LEBANON’S new President, Mr Rene Muawad, yesterday worked to weld old militia foes into a cabinet to govern his divided country while fellow Christians, demonstrating against him, shut half of Beirut.

Alternatively, the normal tags and words can be suppressed so that we get a view of the corpus as though it consisted only of part-of-speech tags.
These suppress/display options (shown in Figure 33) provide alternative views of the texts (and search results), but whatever view is adopted, all the words and tag information in the corpus can be part of the search query. Thus all aspects of the corpus are searchable and yet there is control over the components that are displayed.

The other options controlling the display of the corpus are CHANGE FONT, WORD WRAP, EQUAL WIDTHS and SYNCHRONIZE TEXTS. The latter command, if unchecked, allows the user to scroll through each corpus independently.

Figure 33: Suppressing tags

To fully exploit the information in a corpus it is often necessary to work with tags; however, it is possible to use ParaConc without knowing about all the details related to tagging that are covered in the remainder of this section.
8.2 Tags
To set the form of tags, we select TAG SETTINGS from the FILE menu. The most important components that need to be identified are the TAG START and TAG STOP symbols, which are often < and >. A typical set of tag settings is shown in Figure 34 below.

Figure 34: Defining normal tags and tracked tags
Once this information is entered and SCAN CORPUS has been run, ParaConc will distinguish between mark-up and text to the extent shown in the following table.

<table>
<thead>
<tr>
<th>Action</th>
<th>Text/Tag Distinction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Search</td>
<td>No</td>
</tr>
<tr>
<td>Regex Search</td>
<td>No</td>
</tr>
<tr>
<td>Sorting</td>
<td>Yes</td>
</tr>
<tr>
<td>Collocate Frequency</td>
<td>Yes</td>
</tr>
<tr>
<td>Corpus Frequency</td>
<td>Yes/No</td>
</tr>
<tr>
<td>Display/Suppress</td>
<td>Yes</td>
</tr>
<tr>
<td>Word Count</td>
<td>Yes/No</td>
</tr>
</tbody>
</table>
Let us briefly examine a couple of options presented in the table. The simple text search does not distinguish text and mark-up, which means that a search for `name` will find `name` occurring in the tag `<name>` and instances of `name` in the text itself. Similarly, a (case insensitive) search for `at` will find all the instances of both the word `at` and the tag `AT` assuming that they both occur in the corpus. In contrast, the `TAG SEARCH`, which is described in detail in Section 9.3, allows searches that distinguish words and/or part-of-speech tags and consequently is marked in the table as being sensitive to the text/tag distinction. A search for `AT&` will find the word `AT`, and `&AT` will find words tagged as `AT`, i.e., articles.

**SKIP TAGS**

The corpus frequency command will either include tagnames in the count or ignore them, according to the setting of `SKIP TAGS` in `FREQUENCY OPTIONS`. If `SKIP TAGS` is selected, tag names are excluded from the corpus frequency count.

**page/para #**

It is possible to locate the line number and in some cases the page or paragraph number of a particular hit. To make the program track line numbers, go to `NORMAL TAGS` in `TAG SETTINGS` and select `DISPLAY LINE NUMBERS`. If your corpus indicates pages by means of mark-up such as `<p>`, then enter `<p>` in the page delimiter text box and select `DISPLAY PAGE NUMBERS`. The program will simply count these tags. If there are 234 instances of `<p>` occurring in the corpus before a particular search item, displays the information as follows: Page 234.

There is a further box, labelled `RELATIVE LINES`. Select this box if you want the line numbers to be shown for a particular page such that the location of a particular hit is given as, for instance, page 6 line 2 rather the absolute line number (line 182).

**tracking**

It is clearly useful to be able to track the location of hits (by filename and page/line number), but if the structure of a corpus is encoded by the use of tags, then it will also be possible to inspect the links between individual concordance lines and particular tags.

The `TRACKED TAGS` feature allows us to identify the value of a tag for any of the lines in the concordance results. This means that even if the tag is not visible in the concordance line, the tag value can be identified by simply clicking on the appropriate line. Selecting a line in this way causes the filename, line number and any `TRACKED TAGS` to appear in the lower left of the `ParaConc` window.

**display**

If there is too much information to display in this way, an alternative is to click on a concordance line using the right mouse button. This action evokes a pop-up window and the user can then select the item `DISPLAY INFO` and all the tracking information for that line will be shown. (This information can optionally be included in saved and printed versions of the concordance results.)

To specify one or more `TRACKED TAGS`, go to `TAG SETTINGS` in the `FILE` menu and select `NORMAL TAGS`. Choose `ADD` in the `TRACKED TAGS` section of the resulting dialogue box, enter a name for the tag and enter the tag start and tag stop information, for example, either `SP` and `/SP` or `<SP>` and `</SP>`.
For a COCOA style tag, such as <A Dickens>, select internal information and enter A in TAG START, and leave TAG END empty. Note: it is necessary to re-run SCAN CORPUS if any major changes are made in TAG SETTINGS.

8.3 Part of Speech Tags

speech_NNS To take advantage of a corpus that is tagged for part-of-speech information, it is necessary to indicate the distinction between part-of-speech tags and other tags. Once this is accomplished, the suppress/display options will work and, more importantly, we can perform a word/tag search. Again we select TAG SETTINGS and this time choose SPECIAL TAGS. If the POS tags take a form such as the_AT, we select EMBEDDED IN WORD and enter _ in the DELIMITER CHARACTER box. This might be described as “attached to word.” The tag is connected directly to the word by a special symbol such as _ or ^, and the end of the tag is typically a space or other word-delimiter character. If the tags are specified in a SGML style, as in <w AT>, then we select OUTSIDE WORD and enter <w and in TAG START and > in TAG END. See Figure 35. These tags, which may or may not be attached to the word, have an explicit tag end (although it would be possible to use # meaning “space” as the tag end, but this would not cover line returns, etc.) If the tag precedes the word it classifies, it is necessary to select the BEFORE WORD box.

8.4 Meta-tags

user tags It is possible for the user to define POS meta-tags to search more easily for certain classes of words and to add the possibility of creating something like a user-defined set of tags that is built on top of the tagset provided in the corpus. Let us give a simple example. Say that we want to define a meta-tag that covers the POS tag AT (the) and AT1 (a). In the META-TAGS area of the part of speech tag settings, we choose ADD and in the resulting dialogue box.

Figure 35: Entering POS Tag Settings
we add a name for the meta-tag, ART, and in the CONTENTS box, we simply enter AT, AT1 and click Okay. This meta-tag can then be used in a tag search just as if it were a tag occurring in the corpus. To initiate a tag search, choose ADVANCED SEARCH and then under SEARCH SYNTAX select TAG SEARCH. The syntax of the search is (word)&(POS tag) such as &ART *s&NNS, as described in detail in Section 9.3.

8.5 Using Workspaces

As is clear from the previous section, loading and processing a parallel corpus can take some time. Since the same sets of corpus files are often loaded each time ParaConc is started, it makes sense to freeze the current state of the program, at will, and return to that state at any time. This is the idea behind a workspace. A workspace is saved as a special ParaConc Workspace file (.pws), which can then be opened at any time to restore ParaConc to its previous state, with the corpus loaded ready for searching. The alignment information is also saved as part of the workspace and so it is crucial to use workspaces in order to maintain any alignment information that has been created using the alignment options within ParaConc. Searches and frequency data are not included in the saved workspace. (Only the search histories are saved.)

8.6 Saving a workspace

A workspace—the current corpus and settings of ParaConc—can be saved at any time by selecting the command SAVE WORKSPACE or SAVE WORKSPACE AS from the FILE menu. The usual dialogue box appears and the name and location of the workspace file can be specified in the normal way. Once a (memorable, descriptive) filename for the saved workspace has been entered, the user is asked to choose some different workspace options. The line/page and the tag info can be saved as part of the workspace, along with frequency information and formatting/alignment features. (The saved workspace consists of a saved pws file and a bin file.)
It is advisable to create a folder (called Workspaces) to hold all the workspace files (and their associated folders). This will make it easy to locate any given workspace file and avoid mixing of workspace files and corpus files. The first few lines of the workspace file are similar to the following:

```
[GeneralConfiguration]
ConcordancePath=
FreqPath=
CorpusFreqPath=
BigContextSize=10
KeepLinesPages=1
KeepWatchedTags=1
KeepFrequency=1
KeepSegmentIndexes=1
ColFirst=None
ColFirstReverse=0
ColSecond=None
ColSecondReverse=0
ColThird=None
ColThirdReverse=0
ColCustom=0
ColCustomText=
ColMaxItems=0
LanguagesCount=2
```

If something like this appears in your loaded corpus, then you know that you have loaded a workspace file as a corpus file by mistake.

Once a workspace has been opened, then the user has a choice of two commands: SAVE WORKSPACE AS or simply SAVE WORKSPACE.

In addition, the SAVE ON EXIT command can be selected in order to save the current workspace when the user quits the program.

### 8.7 Opening a workspace

It is possible to choose OPEN WORKSPACE from the FILE menu (or select CTRL-O) in order to load a saved workspace file. If OPEN WORKSPACE is activated when corpus files have been loaded, then those files are unloaded and replaced by the corpus specified in the workspace file.

**Shortcut** Hint: A quick way to open ParaConc with the corpus loaded is to double-click on a saved workspace file. (Not implemented.)
9. PERFORMING AN ADVANCED OR PARALLEL CONCORDANCE SEARCH

Summary: Select **ADVANCED SEARCH** from the **SEARCH** menu, choose **TEXT SEARCH** or **REGULAR EXPRESSION** or **TAG SEARCH** and enter the search string appropriate for the type of search selected. To create a search in which the search terms for each language are specified, choose **PARALLEL SEARCH**. The parameters controlling the search are listed in **SEARCH OPTIONS**. The **ADVANCED SEARCH command** contains other refinements **APPEND SEARCH**, and **HEADINGS/CONTEXT search** covered in the following chapter.

The simple searches described in Chapter 5 will suffice for many purposes and are especially useful for exploratory searches. The basic **TEXT SEARCH** is also very useful when used in conjunction with a sort-and-delete strategy. Particular sort configurations can be chosen to cluster unwanted examples (words preceded by *a* and *the* perhaps), which can then be selected and deleted. For more complex searches, however, we need to use the **ADVANCED SEARCH command**. This command brings up a more intricate dialogue box (displayed in Figure 37), which at the top contains the language of the text query and under that a text box in which the search query is entered.
The most important part of the ADVANCED SEARCH dialogue box is labelled SEARCH SYNTAX. The three radio buttons allow users to specify the kind of search they wish to perform. The first, TEXT SEARCH, refers to the basic searches described in the Chapter 5. The REGULAR EXPRESSION search allows for search queries containing boolean operators (AND, OR and NOT). For example, a regular expression to capture the speak lemma might be given as sp[eo]a?k. This expression will match the string sp followed by e or o, an optional a and finally k. As we will see in the next section, this is still not precise enough to yield unwanted matches, but it is more accurate than the search query based on wildcard characters described above. (Attentive readers will already have noticed that the special character ? differs in meaning in the REGULAR EXPRESSION search and the basic TEXT SEARCH.) The third option in the advanced search dialogue box is TAG SEARCH, which allows the user to specify a search query consisting of a combination of words and tags, with the special symbol & being used to separate words from tags in the search query. For instance, the search string that&DD finds examples of that tagged as a demonstrative pronoun and &JJ of& finds all instances of adjectives followed by the word of. (Naturally, a tagged search can only be used if the corpus contains POS tags.) These three types of search will be described in the following sections.
9.1 Text search

The TEXT SEARCH option in ADVANCED SEARCH is the same as the simple SEARCH described in Chapter 5. One advantage of launching a simple search query from the ADVANCED SEARCH dialogue box is the fact that other options such as APPEND SEARCH, and IGNORE CASE OF LETTERS are accessible.

9.2 Regular Expression Search

You will be able to perform reasonable searches using the simple search option, but if you are working with non-tagged texts, you should, over time, explore the use of regular expressions since they provide a level of control that will allow you to formulate complex, precise searches.

The full complexity of regular expressions, a well-defined set of string operators, can be overwhelming at first, but it is certainly possible to make immediate use of the simpler aspects of regular expressions and to build up complex searches step-by-step. It is useful to refer to previous regular expression searches saved in the search history list, and if you create a particularly useful regular expression, you might want to use store the expression a file so that search strings can be retrieved rather than recreated.

To initiate a regular expression (regex) search, select the radio button labelled REGULAR EXPRESSION located on the left side of the ADVANCED SEARCH dialogue box. Note that three examples of regular expression searches appear under the search text box, as an aid to remembering the appropriate form of regex search queries.

Regex searches are string searches, which means that the computer searches for the string anywhere in the text, regardless of word boundaries or other surrounding text. Thus, if let is specified as the search term, then letter, toilet and complete are all potential matches for this search term.

Because regex searches are string searches, word boundaries must be indicated explicitly and there are a variety of ways to do this. The basic method is to use symbols for the left and right boundaries, \< and \>, as in \<let\>. Or, if you wish to find words starting with the letter z, you could search for \<z. If you try this, you will see that only the initial z of the word is highlighted and to order the z words in alphabetical order you must choose a 1st right sort.

Similarly, if you wish to find words ending in z, you can search for z\>. A more general and slightly easier way of specifying a word boundary is by using the meta-character \b. Thus a search for let can be specified as \blet\b. Meta-characters such as \b typically come in complementary pairs and in this case the alternative symbol is \B, which stands for a non-word boundary. Thus a search for \Bz\B will find all those words which contain z surrounded by non-word boundaries—that is, by other letters or numbers.

How can a search for the phrase spoke to be specified? One way to capture this is by means of the search string \bspoke\b \bto\b, which includes the space character between the two word boundary symbols. One alternative
to entering the space character is to use the whitespace meta-character \s, which covers both space and tab. However, whitespace does not cover all word delimiters and therefore in order to avoid missing some instances of the phrase *spoke to* it is better to use the meta-character \W, which stands for any symbol that is a word delimiter; in other words, it covers all the characters listed as word delimiters in *SEARCH OPTIONS*, along with space, return, etc. Thus the alternative search string for *spoke to* is \bspoke\Wto\b (or \Wspoke\Wto\W), which is based on non-word characters rather than the word boundary. (It is not necessary to specify word boundaries and search for \bspoke\b\Wbto\b; the simpler query above yields the same results.)

You might think about how likely it is that \bspoke\Wto will give the same results. (Certainly, this search string will find all instances of *spoke to*, but it may also snare other strings if the corpus contains phrases such as *spoke together*.)

A solid session reading about regular expressions is not much fun, and here, in particular, time reading this text should be interspersed with generous periods of experimentation. In general, it is best to try out different regex searches, starting with simple searches and increasing the complexity of the expression symbol by symbol. It is also true that regexes created one day look opaque, even to their creator, the next day. Useful regexes should be saved in a file with a descriptive label.

While there is a generally agreed core of regex forms, each software program that includes them has its own idiosyncrasies due to the general context within which the regular expressions operate. Even if you are familiar with regex syntax, you will need to experiment with a variety of options to see how regular expression searches are implemented in *ParaConc*.

[aeiou] Square brackets are used to indicate choices, as in [eo], which matches one character, e or o. In addition, it is possible to specify a choice of a set of characters, as in [0123456789], which can be also be given as the range [0-9]. Letters and numbers can be specified as [0-9a-zA-Z], but remember that despite the length of this range only one alphanumeric character encountered in the text is matched by this search term; square brackets are always associated with just a single character in the text.

\d \w \D \W Metacharacters are commonly used in place of the standard ranges. Thus \d is any digit, equivalent to [0-9]. The metacharacter \w is equivalent to any alphanumeric character, but in addition to digits and numbers, it includes characters not listed as word delimiters in *SEARCH OPTIONS*. Conversely, \D is any non-digit and, as discussed above, \W is any non-alphanumeric.

\| OR Let us return to the question of how regular expressions can be used in a lemma search. One possibility is to use \| to indicate logical “or” as in \bspeak\b|\bspeaks\b|\bspoke\b|\bspoken\b etc. In general, parentheses are used to indicate the scope of a disjunction. In this case, however, they are not necessary since strings take precedence over disjunction. (In other words, the first part of the search query specified above is not interpreted as a search for *speak* followed by \b or \b.)
Another way to perform this lemma search is to specify the query `speak? | spoken?` which includes two kinds of disjunction, `|` and `?`. The question mark is a rather specialised form of disjunction and is typically classified as a counter, which means zero or one instance of the symbol specified. The search string `s?` or `[s]?` stands for `s` or zero (nothing). Other similar forms of counters are `s+` and `s*`, which are equivalent to one or more `s` and zero or more `s`. (Note the difference in the use of `*` in a simple text search where it is equivalent to zero or more characters and here where it means zero or more instances of the preceding expression—`s` in this case.)

In some situations, searches involving the use of `+` and `*` will match a larger than expected chunk of text. This is because these operators match the maximum string possible: a feature known as greediness.

An alternative search string we could enter is `sp[eo]a?k[se]?n?`, which finds words containing `sp` followed by `e` or `o`, and an optional `a`, followed by `k`, an optional `s` or `e`, and finally, an optional `n`. Since we did not specify a word boundary, the search will also capture `outspokenly` and perhaps `exotica` (in English texts) such as words like `prospekt`.

The search string `\bsp[eo]a?k[sei]?n?g?` does a good job at capturing the SPEAK lemma, including `speaking`. Since these strings are fairly complex you should definitely take advantage of the fact that, as mentioned above, the previous search strings are available from the drop-down box at the end of the search text box. A separate search term history is maintained for each of the three types of advanced search.

If we wish to search for `speak` followed by either `to` or `with`, and we try the string `speak \Wto \b | \bwith \b`, we will find that we have specified a search for either `with` or `speak to`. To search for `speak to` or `speak with`, we need to specify the scope of the disjunction and use the search term `\bspeak \W\Wto \b | \b\b\W\Wwith \b`.

You might well be overwhelmed by instances of `New York`, even in a British newspaper text. One way to avoid these instances is exclude the word `New`. To accomplish this, we can take advantage of the `^` symbol and use the search string `[^N][^e][^w] W\WYork \b` or simply, `[^w] W\WYork \b`. If, to continue with our example, you also wanted to exclude instances of the `Duchess of York`, and `Archbishop of York`, then you might disallow both `f` and `w`, as in `[^fw] W\WYork \b`, although this would also exclude `City of York`.

Turning to another example, perhaps we would like to find examples of `an` followed by a word that does not start with a vowel. To do this, we search for `\ban \W[^aeiouAEIOU]`.

The occurrence of `^` as the first item after `\` means `NOT`. In other positions outside of square brackets, it will be interpreted as the beginning of a line. The dollar sign `$` is the anchor for the end of a line.

Let us look at a rather different search query: `\W(is | are) \W1 \W`. The `\1` option is essentially a way of repeating whatever is specified in the preceding parentheses. (If there is more than one set of parentheses, then `\1, \2, can be used to reference the corresponding groupings.) In this
particular example, the expression grouped by parentheses contains a disjunction and the query will match instances of is is and are are. A simple text search (or two) could replicate this search, but there are searches that would be much more difficult to carry out using a simple text search. For example, if we wish to locate three-word alliterations, that is, sequences of three words starting with the same letter, you could use the simple text query a* a* a*, then b* b* b*, etc. Or you could use a single regex search: \W\w*\W\1\w*\W\1\w*\W.

Does English allow a string of sss or ssss in a word? Or are all such occurrences likely to be typos and spelling mistakes? We could simply search for sss, or we can use a counter—a number enclosed by curly brackets—and search for s{3}. If you do this, you might find quite a few typos such as embarrassssing, and also some acceptable expressions such as Yesss! and Psssst! Other formulations of similar search queries are s{3,5} to capture sss, ssss and sssss. And to look for three or more s, we can search for s{3}.

Here is a question. If the word Psssst! occurs in a text and you search for s{3,4}, will the word show up once, twice or three times in the hits? You might think that the hits would be Psssst! Psssst! and Psssst!. In fact, that there will only be one hit, equivalent to Psssst!. This is another example of greediness in action; the search term will only match the maximum string.

If we wish to search for the English perfect in an untagged corpus, then we know that we have to search for something like has or have followed by an -en/-ed form. This can be accomplished by the following regex search query: \bhas\[vs\]e?\W\w\{4,\}e\[nd\]\b. The part of the search query that we hope will match the participle is “padded” with alphanumeric characters (\w) to eliminate shorter words ending in -en or -ed such as ten and bed. However, some good hits such as seen will also be omitted by this search query and in cases like this it is up to the user to formulate the search query in such a way as to get the right balance between a good retrieval rate and a high percentage of desired forms in the concordance results. The specification of a minimum of four letters in the participle in the search query above has the effect of increasing the percentage of good “hits” in the results, at the cost of missing some instances of the present perfect that occur in the corpus.

Let us continue with this example and make a modification to the search query to capture cases in which a word intervenes between the auxiliary and participle. Thus, to capture those strings in which an adverb (or not or been) occurs before the participle, we can add a specification for a word (at least three characters long) \W\w\w+w+\W or \W\w\{3,\}\W. The revised search query then becomes \bhas\[vs\]e?\W\w\{3,\}\W\w\{4,\}e\[nd\]\b. And, finally, to make the adverb optional we group the “word” characters comprising the intervening word and add a counter: \has\[vs\]e?\(\W\w\{3,\}\)[0,1]\W\w\{4,\}e\[nd\]\b.

You might try out a broader search query based on the following: \bhas\[vs\]e?\(\W\w+\)[0,2]\W\w+e\[nd\]\b. How do the results compare with those produced by the previous search query?
If you have followed these examples, you will have a sense of the use and power of regular expression searches. I would encourage you to experiment in building up different regex search queries. The language is very powerful and repays some attention to the details of the query syntax. You will need some time to work out the interaction between regexes and other parts of the program. For example, if you are making use of upper case in regex searches, you should make sure that IGNORE CASE OF LETTERS is not checked in the search dialogue box.

For those unsatisfied with the extent of the coverage here, there are several books devoted solely to the elucidation of regular expressions.

The table below summarises regular expression syntax.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>any character</td>
</tr>
<tr>
<td>[a-z]</td>
<td>any lower case letter</td>
</tr>
<tr>
<td>[0-9]</td>
<td>any number</td>
</tr>
<tr>
<td>[aeuio]</td>
<td>any vowel (in English)</td>
</tr>
<tr>
<td>[:^ir]</td>
<td>not l or r</td>
</tr>
<tr>
<td>\b[a-z]+\b</td>
<td>lower case word</td>
</tr>
<tr>
<td>\b[A-Z][a-z]*\b</td>
<td>word with upper case initial letter</td>
</tr>
<tr>
<td>\b[0-9]+\b</td>
<td>number</td>
</tr>
<tr>
<td>\b</td>
<td>word boundary</td>
</tr>
<tr>
<td>\B</td>
<td>word non-boundary</td>
</tr>
<tr>
<td>\d</td>
<td>any digit</td>
</tr>
<tr>
<td>\D</td>
<td>any non-digit</td>
</tr>
<tr>
<td>\w</td>
<td>any word character (= alphanumeric)</td>
</tr>
<tr>
<td>\W</td>
<td>any non-word character (defined by word delimiters)</td>
</tr>
<tr>
<td>\s</td>
<td>whitespace</td>
</tr>
<tr>
<td>\S</td>
<td>non-whitespace</td>
</tr>
<tr>
<td>*</td>
<td>zero or more</td>
</tr>
<tr>
<td>+</td>
<td>one or more</td>
</tr>
<tr>
<td>{n}</td>
<td>n instances of previous expression</td>
</tr>
<tr>
<td>{n,m}</td>
<td>from n to m</td>
</tr>
<tr>
<td>{n,}</td>
<td>at least n</td>
</tr>
</tbody>
</table>

**9.3 Tag Search**

A corpus in which each word is annotated with part-of-speech tags provides a richer, more structured database allowing targeted searches and opening up the possibility of performing syntactic analyses.

It is possible to investigate a tagged corpus using a simple **TEXT SEARCH** or a **REGULAR EXPRESSION** search and in this case the search queries simply treat
both the words and their tags as searchable strings of symbols. There may well be good reasons for carrying out such a search, but here we will focus on searching tagged corpora using a *TAG SEARCH*.

Tagging in parallel corpora is rare and for illustrative purposes, let us take as an example of a tagged corpus the *BNC Sampler*, available from British National Corpus at Oxford University. (See [http://info.ox.ac.uk/bnc/](http://info.ox.ac.uk/bnc/).) A few lines from the corpus were shown above on page 49 and if you look at that sample, you will see that the POS tags, which have the form `<w TAG>`, are attached and precede the words they categorise. Thus, in this corpus, the adjective *old* is rendered as `<w JJ>old`. (Punctuation is indicated by tags of the form `<c PUNC>`.)

Before initiating a *TAG SEARCH*, information on the format of the different tags must be entered in the SPECIAL TAGS option in *TAG SETTINGS* and the SCAN CORPUS routine must have been run. For details, see Section 8.1.

To perform a *TAG SEARCH*, select ADVANCED SEARCH from the SEARCH menu. Once the ADVANCED SEARCH dialogue box appears, select the *TAG SEARCH* radio button in the SEARCH SYNTAX section on the left of the dialogue box, as shown in Figure 22.

The & is used as a special symbol to distinguish specifications of words from specifications of tags. If an alternative symbol such as $ is preferred, then simply substitute $ for & in the TAG SEARCH SEPARATOR text box in SEARCH OPTIONS.

### 9.3.1 Searching for words and tags

Having a tagged corpus opens up a lot of possibilities. We can look for different constructions, as described below, and in searching for words, we can pinpoint the forms we are interested in.

In English, in particular, there is a considerable amount of zero derivation, i.e., a lack of morphology, which means that many words have the same form whether they are verbs or nouns. Thus, while *hold* most frequently occurs as a verb, it is also a noun, as in *ship’s hold*. Having a tagged corpus makes it much easier to locate the desired POS, and this becomes particularly important when we want to search for the rarer of the two forms. For instance, we can search for the noun *hold* by entering `hold&NN1` as a search term. If we want to find the plural noun form *holds*, we can use the search query `holds&NN2`.

What if we want to search for both singular and plural noun forms? In this case, we can make use of the simple wildcard characters % ? * and enter the search query `hold%&NN?` or `hold%&NN*`. 

wildcards

old&JJ
Note that if IGNORE CASE OF LETTERS is left unchecked, then the search for tags will be case sensitive.

So far I have emphasised the positive aspects of working with a tagged corpus. However, once the search is completed, a disadvantage of a tagged corpus is only too clearly evident in the cluttered appearance of the search results. Now that we have used the tags to refine our search, we will probably want to hide the POS tags, and perhaps the normal tags too, before examining the results in detail. To achieve this, select SUPPRESS in the DISPLAY menu and choose first SPECIAL TAGS and then reselect SUPPRESS and select NORMAL TAGS. All the mark-up will disappear and you will then be left with only the words. The idea is to have the best of both worlds: to exploit the tags in the search process and then suppress them when the results are obtained.

If we want to allow for the possibility of “intervening words” occurring in a phrasal search, we can make use of the ‘range’ wildcard character @.

### 9.3.2 Searching for part-of-speech tags

To create a search query that locates complex noun-noun compounds consisting of five nouns, we can search for &NN? &NN? &NN? &NN? &NN?.

### 9.3.3 Searching for words

It is simple to search for words in a tagged corpus. One way to do this is to enter the search words and &, as in take& part&. By default, however, you can, in fact, simply search for take part, without specifying the word-tag boundary explicitly. Thus, if you have a tagged corpus and you want to simply ignore the presence of tags, you just enter the words or phrase you wish to locate. (Note: this will only work in the TAG SEARCH option.)

### 9.3.4 Meta-tags

In the above searches, we made use of wildcard characters to capture a range of tags in the search. For example, we used the search term NN? to capture the tags for both singular and plural nouns, NN1 and NN2. However, the search query NN? also matches the tag NNB, which is used for titles such as Mr., Lady, and General. What we need is a way to specify a tag representation that covers singular and plural nouns but not titles.

To do this, we can create a meta-tag CNOUN, which we define in the meta-tags section. We choose TAG SETTINGS from the file menu, select PART OF SPEECH TAGS and in the META-TAGS part of the dialogue box, choose ADD. Enter the name of the new meta-tag (i.e., CNOUN) and enter all the constituent tags (i.e, NN1, NN2), separated by a comma in the TAGS box.

Note: In defining the tags making up a meta-tag, it is permissible to include wildcard characters in order to capture more than one tag name. Meta-meta-tags are not allowed, however; you can only use actual tags in the definition of a meta-tag.
The use of meta-tags allows the user to build on the tagset of the corpus to define a new tagset to be used in search queries. Say, for instance, that we wish to define a tag complementizer that includes *who*, *whom*, *which* and *that*, which in the Sampler are tagged as, respectively PNQS, PNQO, DDQ, and CST. It is very easy to define a new tag COMP that covers all four original tags. Once this is done, searches can be carried out using the overarching category COMP or a specific (original) category such as PNQS.

### 9.3.5 Searching for punctuation

The *BNC Sampler* corpus contains tags for punctuation as well as tags for words. The punctuation tags have the form `<c PUNC>` and in the settings for SPECIAL TAGS we entered both `<w` and `<c as valid tag starts. Based on these punctuation tags, we could look for three-word sentences using the query `&YSTP * * * &YSTP`.

Note: Since the full stops (periods) will belong to different sentences, we need to ensure that SENTENCE MODE in the ADVANCED SEARCH dialogue box is not selected. In addition, even though we are searching for tags indicating full stops, we need to remove the full stop from the set of word delimiters specified in SEARCH_OPTIONS.

### 9.4 Parallel Search

One kind of search tailored for use with parallel texts is a parallel search, which is an option in the SEARCH menu. This command allows you to constrain your search based on occurrences in the different parallel texts. The search dialogue box is shown in Figure 38.

![Figure 38: Parallel search](image_url)
Clicking on the Pattern box under Language: English brings up the normal advanced search dialogue box and a search query can be entered. In this case, I typed in **head**. We can then move to Language: French and again click on Pattern and enter another search string such as **chef**. If we click OK, the search routine will look for examples in which *head* occurs in the English and *chef* is also found in the corresponding French segment. If the **NOT** box is selected, then the search routine will display *head* only if *chef* does not occur in the equivalent French segment.
10. PERFORMING ADDITIONAL KINDS OF SEARCHES

Summary: Search possibilities are CONTEXT SEARCH, HEADING SEARCH and APPEND SEARCH.

10.1 Context search

@ range

It is possible to search for `speak` followed by `mind` two words to the right by using a simple TEXT SEARCH and searching for `speak * mind`. This is an example of a context search, which is more restrictive than a regular search. We are searching for `speak` with a right context of “a word followed by `mind`.” The obvious limitation here is that the context must be stated in terms of a set or known number of words to the right. What we want to be able to specify is a range of intervening words. This is accomplished by using the wildcard @, which we have already seen in action above. The range matched by @ is set within the SEARCH OPTIONS dialogue box and can be anything from 0 (no intervening words) to 9 (nine intervening words). Thus we can search for strings using the format `speak @ mind` (or `speak@mind`), with the value of @ set to a range such as 1-3, 2-4, etc. If it is necessary to search for @, i.e., treat it as a literal symbol, then any alternative symbol, such as #, can be entered as the new range wildcard in SEARCH OPTIONS.

Some additional points: It does not make any sense to start or end a search query with @, and an error message will occur if you try to do this. On the other hand, there is no limit to the number of times @ can be used in a search, and as we have seen, the wildcard @ can be used in both simple text searches and in tag searches. This range symbol cannot be used in regular expression searches, however.

10.2 Context search II

A left and a right context can be specified by checking the HEADINGS/CONTEXTS box under ADDITIONAL SEARCH CONTROL in the ADVANCED SEARCH dialogue box. Selecting EDIT brings up a dialogue box in which contexts restricting a search can be specified. The left and right contexts may be tags or mark-up such as `<TITLE>` and `</TITLE>` or plain words and phrases.

Note: Setting the left context as > and the right context as < is one way to ignore the html tags when searching a downloaded web page.
Let us look at a made up “fruit” text to see how this works. The potential
targets of the search are the variations on word (in boldface), and the
underlined fruit words are potential contexts, used here for illustrative
purposes.

word mmm apple mmmmm worda mm m mmm banana mmmmm m mm
mm wordb mmmmm mmm mm mm wordd mm mm plum mmmmm mmm mm.
mmm worde mm mm m mmm apple mmmmm wordf mm mm banana mmm wordg
mmm apple mmmmm m mm banana mmm plum mmmmm wordh.

First of all, as a baseline, note that if we perform a simple (non-context)
search for word% (where % is a wildcard character), we will find all nine
instances. On the other hand, if we perform a context search in which the
left context is specified as banana, then the search will pick up wordb to
wordh — that is, all the instances except the first two, which both occur before
banana.

Adding plum as a right context changes the procedure as follows: The
search algorithm works through the text until it finds the left context
(banana) and then starts searching for all instances of word% until it
encounters the right context (plum). Once the right context word is found,
the program continues to run through the text looking for the left context
again before registering any further instances of word%. Thus specifying a
left context of banana and a right context of plum, the program will locate
wordb, wordc, wordd, and wordg.

If you try a sample search like this and don’t get the expected results, you
should make sure that (i) the HEADINGS/CONTEXT box is selected and (ii) the
appropriate search settings are selected (such as IGNORE CASE OF LETTERS).

Is this clear so far? If not, you should step through the above searches using
the “fruit” text to convince yourself that the search results will indeed turn
out as stated. You might work through the fruit text and write ON when
the left context is reached and OFF when the right context is located, then
look for the search terms located between ON and OFF. Alternatively, you
should create your own toy text and try out various possibilities.

We can make the left context more complex and use the “phrase” mm
banana, keeping plum as the right context. What will the program find
now? Only one instance of word%, namely wordg.

What if we want to allow discontinuous elements in the left context, say
banana followed somewhere in the text by apple. To achieve this, we use the
extended range character @@. With the left context specified as
banana@@apple and the right context as plum, a search for word% would
finds wordf and wordg. We can also use @@ in the right context. We can
illustrate this using our dummy text once more. Imagine that we set the left
context to apple and the right context to banana@@plum. The first instance
of word% to be found is worda. Next the beginning of the right context
(banana) is encountered and the search for word% is suspended as the
program ploughs through the text looking for *plum*. Once *plum* is located, the program searches for the left context *apple* again and after processing the whole text the end result is that two items *worda* and *wordf* are found.

A minor point: The results of searches in which the left context is *X* and the right context is *Y@@Z* are very similar to searches in which the left context is *X* and the right context is simply *Y*. Differences occur when an *X* occurs between *Y* and *Z*. Such specifications may not give the result expected by the user because a left context may be ignored because it is in the scope of @@ in the right context.

There is nothing to prevent the use of the same word as both a context and a search term, except, of course, rampant confusion in the mind of the user.

Having worked through the basic mechanisms using the “fruit” text, you might try a context search in a real text before reading further details about context-constrained searching.

The above explanation of context searches is based on the use of simple text searches, but it is also possible to combine a context search with a REGULAR EXPRESSION search. Selecting this search option means that **both the search query and the context will be interpreted as regular expressions**. Taking advantage of the flexibility that regular expression syntax allows us, we can use disjunctions (among other possibilities) in identifying the context strings. For example, we can specify the left context as *apple || banana* (or *app || ban*).

It will now come as no surprise to learn that it is also possible to add context restrictions to a TAG SEARCH so that specifications using the word&tag format can be entered as left and right contexts. It is also possible to use the long range wildcard @@, described above, in the context specification.

With all these searches the main thing to remember is that the syntax appropriate for the specification of the left and right contexts is the same as the syntax of the search query, with the proviso that @@ is only meaningful in the specifications of contexts and cannot be used in the search query. Finally, we should note that it is permissible to specify only a left context with no right context, but not to set a right context with no left context.

**10.3 Heading search --- NOT IMPLEMENTED**

Some corpora (such as the *British National Corpus*) are structured in such a way that they are divided into a series of headings and texts. The heading, which contains information about a text, is followed by the text itself. The HEADING SEARCH enables the user to select which texts are to be searched, based on the presence of particular specifications in the heading. Thus if the headings contain an annotation about the sex of the authors, then it should be possible to search only those texts written by female authors. For example, the pattern entered in the heading search might be *author=fem*.

To perform a heading search, you first check the **HEADINGS/CONTEXTS** box (towards the bottom of the ADVANCED SEARCH dialogue box) and then choose **EDIT**. In the **HEADINGS/CONTEXTS** search dialogue box that arises,
choose **EDIT** again. A **HEADINGS CONFIGURATION** dialogue box appears in which you can **ADD** (i) a heading name, (ii) the begin and end tags that delimit the heading, and (iii) the tags that indicate the beginning and ending of the text itself.

Once the **HEADINGS CONFIGURATION** has been specified, the **PATTERN** to be searched for in the heading (e.g., **author=fem**) can be entered in the **HEADINGS/CONTEXTS search dialogue box**. All the heading information is now in place and so it just remains to enter the search query itself.

This is another kind of context search. It allows us to search for a word (say *victory*) used in texts written by female authors. Once the headings have been specified as above, a search can be initiated. The program locates the heading and looks for the pattern specified (e.g., **author=fem**). If the pattern is found, then the text following the heading is searched for the string specified in the search text box (e.g., *victory*). If the pattern **author=fem** is not found in the header, then the program ignores the corresponding body of text and checks the next header for the presence of author=fem, and so on until the entire corpus has been searched.

### 10.4 Append search – NOT IMPLEMENTED

You may sometimes wish to combine the results of two searches. For example, if you wish to approximate a lemma search for GO, you might search for **go%%**, and then perform an **APPEND SEARCH** for **going**, and then another **APPEND SEARCH** for **went**.

To do this, you perform the first search, then check **APPEND SEARCH** in the **ADVANCED SEARCH** dialogue box. This leads to the concordance results from the second search being appended to the end of the concordance lines from the first search.

As noted above, a **REGULAR EXPRESSION** search can used to achieve the same results much more efficiently, but the **APPEND SEARCH** option is conceptually simpler.
11. General Search Control and Search Options

Summary: Settings related to searches are located in SEARCH OPTIONS and in GENERAL SEARCH CONTROL in the ADVANCED SEARCH dialogue box.

The parameters controlling searches are located in two places: GENERAL SEARCH CONTROL in the ADVANCED SEARCH dialogue box and in SEARCH OPTIONS. The features under GENERAL SEARCH CONTROL (visible in Figure 37 for example) are checkboxes for:

- IGNORE CASE OF CHARACTERS
- USE SKIPPING AND EQUAL CHARACTERS
- SENTENCE MODE

The features under SEARCH OPTIONS are:

- MAX SEARCH HITS
- FREQUENCY OF HITS
- CONTEXT TYPE
  Special characters
  - Range character
  - TAG SEARCH SEPARATOR
  - CHARACTERS TO TREAT AS (word) DELIMITERS
  - SKIPPING CHARACTERS
  - EQUAL CHARACTERS

  Wildcard characters
  - HEADINGS

11.1 Ignore case of letters

The “factory” default is set such that searches that are insensitive to case, that is, a search for let will find Let and LET. To change this setting, simply check (or uncheck) the box labelled IGNORE CASE OF LETTERS in GENERAL SEARCH CONTROL.

This setting can have wider ramifications than you might expect. For example, a REGULAR EXPRESSION search for [A-Z] will (surprisingly) match a lower case letter if IGNORE CASE OF LETTERS is selected. In addition, the specification in the left and right contexts of context searches is sensitive to this setting, as is the specification of tags in TAG SEARCH.
11.2 Skipping and equal characters

Selecting the USE SKIPPING AND EQUAL CHARACTERS option by checking the appropriate box in the ADVANCED SEARCH dialogue box brings into force the settings for these two parameters as entered in SEARCH OPTIONS.

The skipping characters are useful for avoiding potential problems with mark-up symbols. It is often the case that words in spoken corpora contain non-alphanumeric symbols that are used to indicate prosodic information. Thus, you may come across different forms of the same word in a spoken corpus: speaking and speak^ing, for example. This causes a fundamental problem for word searches since a search for speaking will miss speak^ing. The answer, not surprisingly, is skipping characters. We simply enter ^ in the text box for skipping characters in the SEARCH OPTIONS dialogue box. The result is that a search for speaking will find both speaking and speak^ing (or even s^p^e^a^k^i^n^g). Basically, the occurrence of ^ is ignored.

d=t

The EQUAL CHARACTERS option is useful for finding alternative spellings, e.g., making d equivalent to t (d=t) or for ignoring certain distinctions. Thus if you wanted to disregard the particulars of vowels in a romanised version of Arabic, you could enter a=u=i in the EQUAL CHARACTERS box in SEARCH OPTIONS. If two sets of options are required, they should be separated by a semicolon: d=t; a=u=i.

11.3 Sentence mode

A search for a phrase, particularly a search using the range wildcard character @, may catch words spread across a sentence boundary with the result that the beginning of the string is in one sentence and the end is in the following sentence. Sometimes this is fine, but nine times out of ten, the user will simply eliminate these examples because they do not fit the desired target pattern. Checking the SENTENCE MODE option in the ADVANCED SEARCH dialogue box restricts the search such that hits are only displayed if all the components of the search string occur within a single sentence.

11.4 Maximum number of hits

The maximum number of concordance lines or hits is constrained by various program limitations. The maximum number of concordance lines that can be displayed can be set to whatever high number you choose. However, you may occasionally wish to fix the number of hits. For example, if you are performing several searches, you might want to limit each search to 100 hits, for example.

The factory default setting for the maximum number of hits is 500.

11.5 Frequency of hits

The default value for FREQUENCY OF HITS is 1, which means that every instance of the search term found in the text is displayed in the concordance results window. This is usually what you want, but if you search for a very common word, you may well prefer to gather examples from all the files in
your corpus, and not just from the first file that is searched. To allow the sampling of hits from a range of text files, you need to change the value of FREQUENCY OF HITS to a larger number. For instance, entering 5 in FREQUENCY OF HITS in the SEARCH OPTIONS dialog box will result in every 5th hit being displayed.

11.6 Changing context type and size
Generally, the context is set so that each context line fills the width of the screen, allowing the results to be viewed easily. This is the traditional KWIC display. However, there are, in total, five context types: characters, words, lines, sentences and segments. In addition, for all the options apart from sentence, an appropriate size of context can be specified in the relevant units: 40 characters, 8 words, 2 lines etc.

In fact, there are actually two ways in which the context can be set. First of all, the default context type, which controls the way in which the results of each search are initially displayed is set in SEARCH OPTIONS. The same range of settings also appears in CONTEXT TYPE in the DISPLAY menu. Changing the options in the DISPLAY menu only alters the format of the current results window; it does not change the default setting.

Typically, the default context is based on a number of characters or words. This provides a KWIC display, which makes it easy to visually scan the results. The default context for the “non-search” language is segment. Once the appropriate results have been identified, users may want to switch to a sentence context so that the examples can be saved to a file in the form of a series of sentences.

11.7 What is a word?
It is clear that a prerequisite for a word searching program is a definition of what a word is. This is not a particularly difficult issue in written texts—the first definition of a word that comes to mind is a string of letters (and perhaps numbers) surrounded by spaces. And with a little further thought, we would realise that we need to include punctuation symbols, in addition to spaces, as possible delimiters of words. Hence, we can define a word as a string of characters bounded by either spaces or punctuation (plus special computer characters such as the carriage return).

Let’s examine a couple of situations in order to illustrate the subtleties that you are likely to encounter after a little experimentation with concordance searches. To exemplify the complications you might run into, we can consider the first word in the previous sentence. But what is the first word? According to our preliminary definition it is let, and the second word is s. Similarly, we can ask whether committee’s should be treated as one word or two. And the same question can be asked concerning mid-day, and so on. ParaConc is initially configured so that by default the apostrophe is taken to be part of a word. This means that a search for let will not find let’s because...
the latter would count as a 5-character word, whereas the search was for the 3-character word let. (To find both let’s and let, we would need to specify the search term let%%. ) If, on the other hand, the apostrophe counted as a word delimiter, then searching for let would find let’s or at least the first part of let’s.

The advantage of having the apostrophe be part of a word and not a punctuation character (or word delimiter) is that you can then search for the apostrophe. Thus the search *s would find all possessives, as well as contractions of is and has. (Note: If you are really interested in possessives in English and want to avoid forms such as he’s and it’s, you can ensure that the string before the possessive is at least three characters long by using the search query ???’s.)

If you decide that you want to change the search parameters so that let’s and let would both be found by a search for let, then you simply select SEARCH OPTIONS and add the apostrophe to the characters that define a word boundary. These characters, the word delimiters, are listed in a text box, and characters can be added or removed from the list by simply typing in or deleting the appropriate characters. In this case, we need to click in the text box and type in the apostrophe character.

Having made this change, we can now search for let and find let’s. On the other hand, a search for ???’s will now cause an error message to be displayed, because it is not possible include a delimiter character in a search. (Space is an exception to this rule. It is, of course, possible to include a space in the search string and space will then match any delimiter.)

Let’s run through a couple more examples. If you wish to search for sentence boundaries, you can remove “.” from the list of delimiters and search for * not . alone, which would only work if there were a space between the last word in the sentence and the full stop. Along similar lines, you might be interested in examining sentence-initial occurrences of a word such as first. In this case, you can search for *. First (assuming that there are no tags indicating a sentence boundary). And, a sentence-initial active participle may be captured with *. *ing.

Clicking on the drop down arrow at the end of the delimiter box will reveal a list of previous settings for word delimiters. Any of these previous settings can be restored if necessary. It is also possible to restore the original “factory” setting for the list of word delimiters by selecting the button REVERT TO DEFAULT in the SEARCH OPTIONS dialogue box.

11.8 Changing the wildcard characters

In discussing search options above, we saw the usefulness of % as a wildcard character. But what if we want to search for % as a literal in 100%? To allow a search for %, it is necessary to first replace % as a wildcard symbol, perhaps with $, so that %, as %, can be searched for. To do this, select SEARCH OPTIONS and simply substitute $ in the ‘zero or one character box’ in the SEARCH OPTIONS dialogue box. We can then search for % with the
character then being treated literally as a percent sign rather than as a wildcard character.
The other wildcard characters ? (exactly 1 character) and * (0 or more characters) and the range character @ can be replaced in the same way. It is also possible to change the word-tag delimiter (&) used in tag searches.

11.9 Headings – NOT IMPLEMENTED
Before a heading search can be performed, it is necessary to specify the tags that define the heading and the body. The dialogue box to do this can be accessed from SEARCH OPTIONS or via EDIT in the HEADINGS/CONTEXTS SEARCH dialogue box.
12. DISPLAYING THE RESULTS

Summary: Moving from one window to another is accomplished by selecting the desired location from the list in the WINDOW menu or by clicking on the appropriate window. The format of concordance lines can be changed by (i) hiding the search word; (ii) displaying frequent collocates; (iii) hiding the collocates; (iv) suppressing the words/tags/POS tags in the concordance window, and (v) changing the context type. It is also possible to create a graph of the distribution of hits across the corpus.

12.1 Navigation
Changing from one window to another is accomplished by selecting the desired window from the list in the WINDOW menu. If the CASCADE option in the WINDOW menu is selected, you can simply click on the visible portion of the window you wish to bring to the foreground.

12.2 Changing Fonts
To change the font, font style, font size, or font colour for each language, select CHANGE FONT, CHANGE HIGHLIGHT FONT, or CHANGE COLLOCATE FONT from the DISPLAY menu. It is also possible to change the Hot Words font.

12.3 Word Wrap
The display of text in the corpus file and in the concordance results (if the context type line or sentence is selected) can be controlled by the WORD WRAP command in the CORPUS TEXT menu and in the DISPLAY menu.

12.4 Suppress words/tags
The concordance lines (or corpus) can be “cleaned up” by selecting SUPPRESS from the DISPLAY menu. The choices for suppression are NORMAL TAGS, SPECIAL TAGS and WORDS. Selecting one of these will suppress/display the appearance of the corresponding data structure. Any two of the three data structures can be suppressed at a time.
The form of the tags is specified in TAG SETTINGS in the FILE menu and the program uses this information to hide the tags.

12.5 Distribution
It is often useful to obtain a sense of the distribution of hits through a corpus. You might want to check for major skewing in the distribution of hits that might be due to an idiosyncratic file or to the different sub-corpora that are present.

Selecting **DISTRIBUTION** in the **DISPLAY** menu leads to a graphical display of hits through the corpus or through an individual file.

Warning: If the concordance results window is not active, then the **DISTRIBUTION** command will not appear.

Also of considerable interest is tracking the distribution of a translation equivalent though the corpus. The diagram below shows the distribution of *head* (in the English corpus) and *tête* (in the French corpus).

![Distribution of head and tête](image)

*Figure 39: Distribution of head and tête*

The x axis indicates the number of hits and the y axis shows the position in the corpus (or in a particular file). If the **PERCENT** checkbox is deselected, the y axis displays line numbers rather than percentages to indicate position. It is also possible to display individual file names on the x axis.

The box labelled **SCOPE** at the top of the screen allows the user to toggle between a display of hits in the entire corpus and the hits in a single file.

In the graph shown, the number of hits are calculated for each 10% of the corpus. Other divisions of the corpus (or file), segments comprising 2%, 5%, 20%, 25%, 33% and 50%, can be set (in **STEP** at the top of the window).

To print a copy of the table, click on the **PRINT** button at the top right of the window.
12.6 Locating hits

Selecting a line in the concordance results window invokes the display of information about the source of the hit in the lower left of the screen. The information given includes the file name and—if specified in TAG SETTINGS—page number, line number and tracked tags. (See Section 8.2.) This information can also be seen by clicking on the line with the right mouse button and choosing DISPLAY INFO.

A window comes into view that provides information on the source file/page/line and on the value of tracked tags. The specification of tags to be tracked is given in TAG SETTINGS (NORMAL TAGS) in the FILE menu.

12.7 Changing the display: KWIC, Sentence, ...

The form of the context—words, characters, etc.—for the concordance results can be changed via CONTEXT TYPE in the DISPLAY menu.
13. THE POWER OF SORTING

Summary: The concordance lines can be sorted to reveal patterns in the data. The basic sorting option involves choosing a primary and secondary sort order. More extensive and more complex sorting options are available in ADVANCED SORT.

13.1 Sorting

As mentioned above, the search terms in the concordance window are displayed initially in what is referred to as text order. However, much of the utility of a concordance program derives from its ability to re-sort the items in such a way that similar forms line up together. This re-sorting reveals recurrent patterns in the text, as illustrated earlier with the search for *speak*. Once the concordance lines are sorted in an appropriate manner, the user can scroll through the results to visually identify common patterns.

Aggregating similar patterns is accomplished by ordering the concordance lines so that they are alphabetised according to, for example, the word to the right of the search term. This is a 1ST RIGHT sort. Other similar sorting options are 2ND RIGHT, 1ST LEFT, 2ND LEFT, ORIGINAL TEXT ORDER, or SEARCH TERM. Choosing one of these sort orders specifies the primary sort. Once one item is selected from the SORT menu, the mouse can be moved horizontally to select a second sort, which specifies the secondary sort order. The primary and secondary sort orders 1ST RIGHT, 1ST LEFT and 1ST LEFT, 1ST RIGHT are commonly used and hence are listed separately in the SORT menu.

A 1ST RIGHT, 1ST LEFT sort means that the concordance lines are sorted alphabetically according to the word following the search term, and then, if there are lines in which the same word occurs after the search term, these lines are further sorted according to the alphabetical order of the word preceding the search term.

The particular meaning of alphabetical order is determined by Windows (not by the software) and depends on the language setting in place.

13.2 Advanced sort

An ADVANCED SORT option is divided into two parts. The first, upper component of the dialogue box invoked by the selection of ADVANCED SORT allows the user to choose a primary, secondary and tertiary sort order (labelled as FIRST SORT, SECOND SORT and THIRD SORT).
In addition, for any sort position a reverse sort can be selected. A reverse sort orders the lines in alphabetical order of the endings of words with the result, that *panda*, for instance, comes before *bear*. (This feature may be especially useful if you are working with Arabic or Hebrew.) The second component is a customised sort, which is selected by a checkbox. This provides even more flexibility since the user can enter a range for the sort, such as 3L-3R. This range will sort the lines 3L, 2L, 1L, SW, 1R, 2R and 3R! Alternatively, the user can specify a sequence of sort domains separated by a comma. The terms permitted in this listing are as follows:

- 3L, 2L, 1L, 1R, 2R, 3R
- 0 = Search term
- DEF = Defined sort
- ORIG = Original order
- 1Lr = 1st Left, reverse sort

Note: If the customised sort option is checked, it is impossible to select items such as FIRST SORT, SECOND SORT, etc. in the upper portion of the dialogue box.

13.3 User-defined sort

A further option available under ADVANCED SORT is a DEFINED option. This allows the categorisation and grouping (i.e., sorting) of concordance lines according to user-defined categories. Thus if you have five categories or classifications that are appropriate for a set of concordance results, you can
assign the letter \( a, b, c, d, \) or \( e \), as appropriate, to each line. To do this, select (click on) a concordance line, click on the right mouse button and assign a letter to that line. The assigned letter is then displayed to the left of each concordance line. Choosing the DEFINED option in ADVANCED SORT orders the lines according to the letters assigned by the user, which will group the members of each type together.

### 13.4 Sorting Tracked Tags

It is possible to sort the results based on the value of tracked tags. Thus, if the tracked tags code the speech of individual speakers, it will be possible to sort the results so that the instances of the search word from each speaker are clustered together.
14. EXITING THE PROGRAM

To exit ParaConc, choose EXIT (CTRL-Q) from the FILE menu.
INDEX

$, 61
\%, 39, 76
\%d, 21
\%, 21
&, 64, 77
*, 77
*, regex, 63
., 63
.pws, 55
?, 39, 61, 77
@, 65, 69, 77
@, usage, 69
@@, 70
[^lr], 63
[a-z], 63
^ anchor, 61
^n\{, 62, 63
{n,m}, 62, 63
l, 60, 61
\l, 61
\d, 60
\s, 60
\W, 60
+, 63
<, 52
<c PUNC>, 66
<p>, 53
<w, 54
<w TAG>, 64
>, 52
1st left, 31
1st left, 83
1st right, 83
2ND LEFT, 83
2nd right, 83
500 hits, 74

absent commands, 80
advanced collocation, 47
ADVANCED SEARCH, 57, 58
ADVANCED SORT, 83
align format, 19
alignment, 11, 13, 19
alignment markers, settings, 23
alignment menu, 22
alignment, saving, 55
alphabetical order, 41

alphanumeric, 60
annotations, 49, 50
any character, 63
any lower case letter, 63
any number, 63
apostrophe, 75, 76
APPEND SEARCH, 72
Arabic, 84
ascii, 18
asterisk, 38
attribute, 20
author=fem, 71
Automatic Recognition, 21

British National Corpus, 64, 71
case, 73
change collocate font, 79
change font, 79
change font of corpus, 51
change highlight font, 79
changing the display, 81
CLEAR SEARCH QUERY, 35
clipboard, 40
COCOA, 54
collocate frequency, 45
collocate highlighting, 30
collocates, highlight using colour, 45
collocation frequency, 45
collocation, customised, 47
COMP, 66
complex noun-noun compound, 65
concordance window, 39
content words only, 43
context, 69
context search, 69, 71, 72
context search, REGULAR EXPRESSION, 71
context search, context search, 34, 73, 75
corpus size limit, 24
corpus, tagged, 49
copy, 40
count search term, 47
CTRL click, 36
CTRL-A, 19
CTRL-O, 56
customised collocation, 47
defined sort, 84
delete, 39
delete item, 39
DELIMITED SEGMENTS, 20
delimiter character in a search, 76
delimiters, 73, 75
disjunction, 71
disk space, 11
DISPLAY INFO, 53, 81
DISPLAY LINE NUMBERS, 53
DISPLAY PAGE NUMBERS, 53
DISPLAY, active window, 34
distribution, 80
drop down arrow, 76
edit heading, 71
end-of-the tag, 54
English perfect, 62
EQUAL, 73
EQUAL CHARACTERS, 73, 74
EQUAL WIDTHS, 25, 51
equivalence, 30
error message, 76
extended range character, 70
FILE menu, 17
file, 15, 16
FIRST SORT, 83
format, 20
frequency data, forcing a recalculation, 45
frequency information, saving in workspace, 55
frequency list, 41
frequency list, printing, 43
frequency list, saving, 43
frequency list, settings, 42
frequency of collocates, 45
FREQUENCY OF HITS, 73, 75
frequency options, 41, 48
frequency order, 41
frequency statistics, 45
frequency, link to concordance window, 45
function words, excluding, 42
GENERAL SEARCH CONTROL, 28, 73
grammatical words, excluding, 42
greediness, 61, 62

headings, 77
HEADING SEARCH, 71, 77
Headings, 20, 21, 73
Hebrew, 84
help, 17
hide tags, 65
hits, 16, 38
hits, none, 16
hot words, 35, 37
Hot Words, change font, 79
Hot Words, options, 37
Hot Words, paradigm, 37
hot words, ranked list, 35
html, avoiding, 69
HTML/SGML, 21
HTML/SGML Markers, 21
IGNORE CASE, 73
ignore case of letters, 63
IGNORE CASE OF LETTERS, FREQUENCY CALCULATIONS, 42
IGNORE CASE, AND REGEX, 73
Indented, 21
info, 15, 16
ini, 11
installation, 11
KWIC, 30, 75
KWIC COMMAND, 34
KWIC, second language, 34

LANGUAGE, 28
left context only, 71
left/right context, 70
left/right context, phrase, 70
lemma, 61
line number, 53
line/page information, 55
LOAD CORPUS FILE, 16, 17
logical or, 60
mark-up, 49
mark-up versus text, 52
MAX CANDIDATES, 37
MAX SEARCH HITS, 73
maximum, 74
maximum frequency, 42
maximum lines, 42
maximum number of hits, 74
maximum number of lines, 42
menu bar, 24
merger, 23
metacharacters, 60
meta-meta-tags, 65
meta-tag, 54, 55, 65
minimum frequency, 41
navigation, 79
New Line delimited, 21
NEW LINE DELIMITED SEGMENTS, 19
NORMAL TAGS, 52
not aligned, 20
NOT, in parallel search, 67
nouns, 65
number of the files loaded, 24

open ParaConc from workspace file, 56
OPEN WORKSPACE, 17, 56
ORIGINAL TEXT ORDER, 83

paradigm option, 37
Paragraphs, 20
parallel corpus, 27
parallel search, 66
PARALLEL TEXTS, 17
parentheses, 60
part-of-speech, 64
part-of-speech tags, 54
Pattern, 21
PERCENT, 80
possible translations, 35
primary sort, 31
primary/secondary sort order, 83
print results, 40
progress bars, 29
punctuation, 66
punctuation, searching, 39

question, search, 39

RAM, 11
range, 60, 73
regex, 59, 60
Regular expression, 21, 58
regular expression search, 59
RELATIVE LINES, 53
remove, 26
results, interpreting, 30
reverse, 84
reverse sort, 84
revert to default, 76

save as file, 39, 43, 48
SAVE AS HTML, 39
SAVE ON EXIT, 56
SAVE WORKSPACE, 55
save, CONCEAL COLLOCATES, 39
save, conceal hits, 39
save, DEFINED SORT LABELS, 39
save, file names on each line, 39
SAVE, LINE NUMBERS, 39
save, normal tags, 39
SAVE, PAGE NUMBERS, 39
save, RELATIVE LINES, 39
save, SEGMENTS, 39
save, SEQUENTIAL NUMBER IN EACH LINE, 39
save, suppress tags, 39
save, 39
saved workspace files, 55
saved, 48
scan corpus, 24
scan corpus, consequences, 52
SCOPE, 80
scope of disjunction, 60, 61
screen, initial, 15
SEARCH, 16
SEARCH OPTIONS, 28, 39, 73, 74
search options, 76
search parameters, 76
SEARCH QUERY, 32
SEARCH SYNTAX, 58
search term, 38
SEARCH TERM, 83
search, exclude, 61
search, stop, 16
search, 27
secondary sort, 31
segment start/stop markers, 20
selecting several files, 19
selective removal, 26
SENTENCE MODE, 66, 73, 74
Sentences, 20, 21
SKIP TAGS, 49, 53
skip tags, frequency calculations, 42
SKIPPING, 73
skipping characters, 74
sort, 31, 38, 83
SORT, active window, 34
SORT, INACTIVE, 84
sort, tracked tags, 85
sort, 83
sort-and-delete, 57
sorting, 83
space, 76
span, collocation, 48
SPECIAL TAGS, 54, 64
splitting, 23
square brackets, 60
start, 15
STEP, 80
Stop List, 42, 43
string searches, 59
SUPPRESS, 25, 39, 65
suppress tags, 50, 79
suppress/display, 54
SYNCHRONIZE TEXTS, 25, 51

TAG SEARCH, 53, 64
TAG SEARCH SEPARATOR, 64, 73
TAG SETTINGS, 50, 79, 81
tagnames, 53
tags, 49, 50, 79
tags, part of speech, 50
tagset, 54, 66
tertiary sort, 83
TEXT SEARCH, 59
text-only files, 18
three-word sentence, 66
track, 53
tracked tag information, 55
TRACKED TAGS, 53
tracked tags, sorting, 85
Two New Lines Delimited, 21

unload corpus, 26

unloaded files, 56

VIEW CORPUS FILES, 22, 25

webpage, save to, 39
whitespace meta-character, 60
whitespace, 63
wildcard, 64, 65, 73, 76
wildcard characters, 38, 39, 77
wildcard in meta-tag, 65
window, 79
Windows, compatible versions, 11
Word, 18, 75
word boundary, 28, 63
word count, 24
word delimiters, 76
word list, 41
word search, in tagged corpus, 65
word wrap, 51, 79
word&tag, 64
word, 75
WORDS, 79
workspace, 11, 17, 55, 56
workspace options, 55

zero derivation, 64