Collocate 1.0:
Locating collocations and terminology

Michael Barlow

ATHELSTAN

Requirements:
Windows 95/98/NT/XP/…
Disk space: 1.5 MB
RAM: 32MB

Related products:
MP 2.2
ParaConc
Corpus of Spoken Professional American English
MonoConc 1.5
Concordances in the Classroom. 1997. C. Tribble and G. Jones
Learning with Corpora. 2001. Guy Aston (ed.)

PREFACE

Collocate is designed to provide information about the collocations in a text or corpus. It is possible to search for a particular word or phrase and get the collocates of that search term, ordered by frequency or a statistical measure such as Mutual Information. In addition, the software can produce a bigram, trigram, n-gram list for a corpus. And finally the software can be used to retrieve all the collocations in a corpus, taking into account the settings specified by the user.

It has to be said that if you use *Collocate* to process your text files, the results produced are unlikely to be all the collocations or terms that you want and nothing else. Using the software involves some experimentation to get the right balance between precision and recall for a particular corpus and for your particular needs.

CONTENTS

0. TEXT ANALYSIS WITH Collocate

Collocate is designed to reveal lexical patterns in texts. The program uses raw frequency and some statistical measures (Mutual Information, T-score, and Log Likelihood) to highlight patterns in the text. In using the statistical tests, the aim is to provide alternative views and alternative rankings of the results, the assumption being that the actual statistical scores are too difficult to interpret with respect to probability of occurrence.

If, for instance, we search for 2-word phrases containing energy in the Biological and Health Sciences part of MICASE, then we get the following frequency results.

Freq	Collocation
26	of energy
16	the energy
10	energy that
9	energy in
8	energy is
7	energy to
6	energy and
6	energy that's
5	and energy
5	in energy
5	energy for
5	energy use
4	energy equation
4	balanced energy
4	your energy
4	kinetic energy
3	energy so
3	more energy
3	energy has
3	an energy

Figure 1: Collocations ranked by frequency

If we examine the same results ordered using the Log Likelihood score, we obtain the ranking shown in Figure 2.

Freq	LL	Collocation
26	71.099282	of energy
4	55.740500	balanced energy
4	35.207808	energy equation
5	27.553384	energy use
3	17.186455	high energy
6	16.428413	energy that's
9	13.085124	energy in
4	12.948427	your energy
16	12.554549	the energy
10	11.155189	energy that
5	11.060908	energy for
8	9.866656	energy is
3	9.501266	energy has
3	8.551006	more energy
3	7.930387	an energy
7	5.614379	energy to
3	3.102033	energy okay
5	3.067220	in energy
6	1.493312	energy and
3	1.132557	energy so
5	0.569459	and energy
4	N/A	kinetic energy

Figure 2: Collocations ranked by Log Likelihood

By inspecting the list, we notice that what we might call good collocations are more likely to be ranked higher in the Log Likelihood list. The top ranked collocation of energy, is not so good, but following that we have *balanced energy*, *energy equation*, *energy use*, and *high energy*. These collocations occurred at the top of the list despite the fact that their raw frequency is low. The collocation that was not ranked highly is *kinetic energy*, which has no value (N/A), which means that there was a division by zero as part of the calculation.

If we take another view of the collocations based on a ranking of Mutual Information scores, we get the results shown in Figure 3.

Freq	Mutual Inf.	Collocation
4	11.406116	kinetic energy
4	10.821154	balanced energy
4	7.705677	energy equation
3	5.515345	high energy
5	5.344340	energy use
4	3.639587	your energy
3	3.590199	energy has
3	3.340027	more energy
6	3.234523	energy that's
3	3.174095	an energy
26	3.119825	of energy
5	2.809927	energy for
9	2.129217	energy in
8	1.926589	energy is
10	1.803046	energy that
3	1.750884	energy okay
7	1.490237	energy to
16	1.435831	the energy
5	1.281220	in energy
3	0.978454	energy so
6	0.770096	energy and
5	0.507062	and energy

Figure 3: Collocations ranked by Mutual Information

Once again see that the good collocations are at the top of the list. This simple analysis illustrates the use of Collocate. The program can be used to present different views of the data to the user who must evaluate the different results obtained by the application of different statistical measures.

1. INSTALLATION

Summary: Copy the file Collocate.exe to the hard disk.

copy files

Collocate can be installed by copying the file on the CD-ROM to the hard disk on your computer. In most cases, the first step is to make a directory called Collocate and then simply copy the files to that directory. (*Collocate* does not come with an installation utility, and if you are unsure how to copy files, you should refer to your Windows manual.)

1.1 Requirements

1.1.1 Disk space

The software files constituting *Collocate* require around 1 MB of disk space on the hard drive. Since *Collocate* carries out some computational intensive statistical analyses the software first creates an index of the corpus and so may write large, temporary files to the disk. The software 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.

1.1.2 Compatible versions of Windows

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

1.1.3 RAM

It is recommended that a computer with Windows 95 installed have 32 MB of RAM and a Windows XP system should have a minimum of 64 MB of RAM. As always, the more RAM the better, but it is possible to run the program (slowly) on a fairly minimal system; the program is designed to work with whatever memory is available.

2. STARTING Collocate

Summary: Double-click on Collocate to open the program. HELP is available in the INFO menu.

To start the program, double-click on the file Collocate. Once the program is open, a simple screen appears, as shown below in Figure 4. This initial screen looks rather bare, containing only a blank window and three menu items: FILE, OPTIONS and INFO.

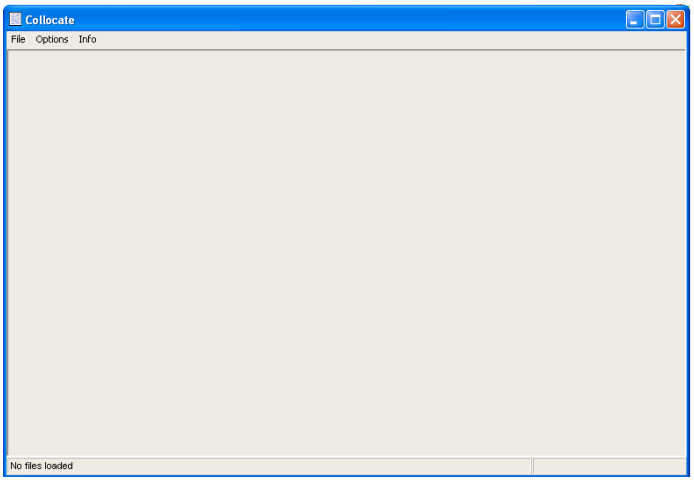


Figure 4: Initial screen

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

commands

Before examining the functionality of the program, we should first establish a protocol for the description of commands in the program. In this guide, the operation of the program is typically described in terms of commands issued via the mouse. This is partly for expository purposes; the phrase "select LOAD CORPUS FILE(S) from the FILE menu" is much clearer than "execute ALT-F L." In terms of actually using the software, learning some basic keyboard commands will make program operation a little faster and smoother. And if you prefer a keyboard style rather than a mouse style of issuing commands,

then in reading through this text, you will be able to observe the keyboard equivalents of the mouse-based commands by paying attention to the following conventions. The menus and commands are displayed in the text with one letter underlined (as they are on the screen). For menus, this underlining indicates the "ALT letter" sequence that will select the menu. For instance, to select the FILE menu, enter ALT-F. In the case of commands, the underlined letter indicates the keyboard letter that will execute the command once the appropriate menu is selected. For example, to initiate a LOAD CORPUS FILE(S) command when the FILE menu is selected, enter L from the keyboard. Thus the complete keyboard command to load a text is ALT-F L. Since loading texts into the program is an important operation, a control sequence, CTRL-L, has also been assigned to this function. (Note: although the control character is standardly represented here in upper case form for clarity, the lower case version should actually be used—the command is CTRL plus lowercase l, not L.) Thus there are actually three ways in all to invoke this command: (i) using the mouse and selecting LOAD CORPUS FILE(S), (ii) entering the sequence ALT-F L, or (iii) entering the control command CTRL-L.

FILE, OPTIONS Let us start at the beginning and work through the different features methodically. Once the program is open, a simple screen appears which contains a blank window and three menus: FILE, OPTIONS, and INFO. Let us deal with INFO first. This menu is always present; it provides access to HELP and to some basic information about *Collocate*, as well as some contact information for Athelstan. The HELP menu is organised according to topic. Double-clicking on the appropriate heading will open the topic file, allowing perusal of the associated descriptions. In addition, the Windows Help utility provides other ways (such as FIND and INDEX) to locate the required information. Navigation of HELP is straightforward and will not be described here. (The Help file, Collocate.hlp, must be available in order for this information to be displayed.)

A further source of information is the brief description of each command which automatically appears at the bottom left of the screen when the command is selected. If you are not sure what action is associated with a particular menu item, you can move the pointer to the command (without clicking) and read the short description that appears in the lower left corner of the window. For example, if the cursor is on LOAD CORPUS FILE(S), the description that appears is "Add file(s) to current

corpus," which is at least marginally more descriptive than the name of the command itself: Load Corpus File.

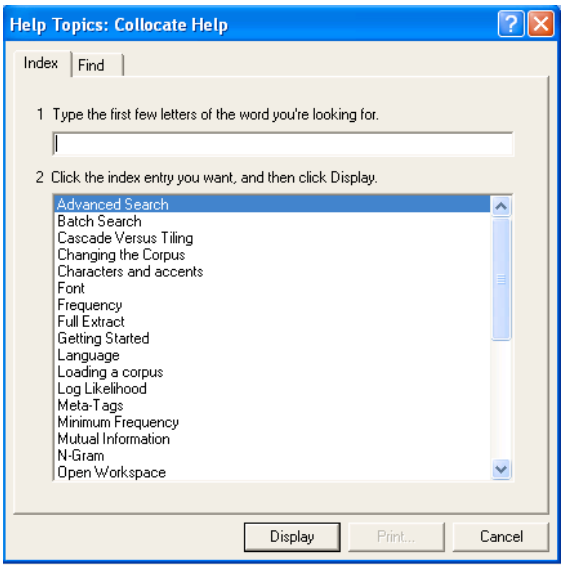


Figure 5: Help contents

The OPTIONS menu contains two commands: COUNT and CONSTRAINTS.

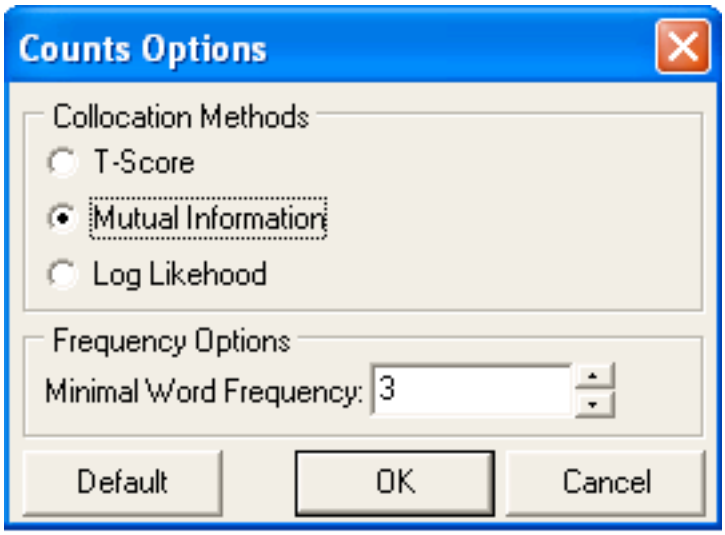


Figure 6: Counts Options

The upper section of the COUNTS OPTIONS dialogue box contains info on COLLOCATION METHODS, which allows the user to choose the statistic used, in addition to raw frequency, in calculating the collocational strength between two words. The options are t-score, mutual information (MI) and Log Likelihood. If you are not used to these statistics, then for now we can just note that Mutual Information tends to highlight strong associations between words without taking into account how frequent, or infrequent, the word pairs are. The other two statistics are sensitive to frequency in the sense that higher scores are associated with higher frequency of occurrence.

The assumption in using frequency and different statistical measures in *Collocate* is that the actual scores are not very meaningful in terms of probability. In other words, we will not talk about probabilities and random events; rather, we will use the statistics to present the use with different views in terms of different rankings in order to highlight interesting patterns in the data.

The second section of the COUNTS OPTIONS dialogue box is FREQUENCY OPTIONS. This contains the minimum cut-off for the different frequency lists: the minimum number of times a bigram, trigram, etc. must occur to be included in the results.

The CONSTRAINTS dialogue box contains some default settings, which are discussed in Chapter 10.

Selecting the FILE menu reveals several commands, one of which is the now well-known LOAD CORPUS FILE(S) option, but before making a corpus available for searching we first need to take a look at a couple of other commands: LANGUAGE and TAG SETTINGS, which are described in the next section.

3. WORKING WITH A CORPUS

Summary: Check the configurations in LANGUAGE and TAG SETTINGS in the FILE menu and then choose LOAD CORPUS FILE(S).

It is often a good idea to enter descriptions of the form of annotations before initiating the loading of file in order for the software to take advantage of the information encoded in texts. (Although this process can also be done after the files have been loaded.) 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 the next section, is a good way to ensure that the information about the annotations used in the corpus only has to be entered once.

Once the corpus files are selected and loaded, an automatic corpus-indexing procedure is instigated.

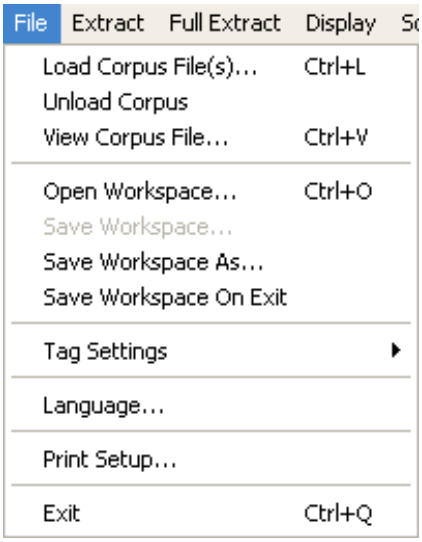


Figure 7: The File menu

3.1 Choosing a Language

Russian, The appropriate method of entering characters with accents and the definition of alphabetical (sorting) order depends on whether the current language is, for instance, English, Russian or Thai. Choosing LANGUAGE displays a list of languages, allowing the selection of the one appropriate for your corpus. (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.

Chinese If *Collocate* is running under a system as different as, say, Chinese Windows, then the settings and behaviour will naturally be somewhat different from the description given here.

Warning:
If a language other than the current language is selected by choosing Okay (or hitting return) in the LANGUAGE dialogue box, then any loaded text files will be removed (unloaded). In other words, changing the language from English (United States) to English (United Kingdom) will cause the automatic unloading of the corpus, and so it is advisable to acquire the habit of first selecting the language and then loading the corpus.

3.2 Counting words and word types

When the corpus files have been indexed the number of word tokens and number of word types will be displayed in the lower right of the window.

3.3 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 source of the text or about the components of the text, such as the division into header and body. In some cases the mark-up is minimal, but in some cases it is extensive.

In this corpus 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, which are discussed in Section 3.3.2, 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.

Increasingly, corpora are highly tagged with information representing the source and structure of texts. For example, one version of a BNC file contains texts that look like the following.

tagged corpus<text decls="CN004 QN000 SN000" org="composite">
<body TEIform="div1">
<div1 type="u">
<head type="MAIN"><s n="1"><w type="VVG-AJ0">Starring </w><w type="NN1">role </w><w type="PRP">for </w><w type="NN1">TV </w><w type="CJC">and </w><w type="NN1">film </w><w type="NN2">props</w></s>
</head> <head type="BYLINE"><s n="2"><w type="PRP">By </w><w type="NP0">Robert </w><w type="NP0">Shrimsley</w></s
</head> <p><s n="3"><w type="AT0">THE </w><w type="NN1">couch </w><w type="VVD">cavorted </w><w type="PRP">upon </w><w type="PRP">by </w><w type="NP0">Joanne </w><w type="NP0">Whalley-Kilmer </w><w type="PRP">in </w><w type="DPS">her </w><w type="NN1">portrayal </w><w type="PRF">of </w><w type="NP0">Christine </w><w type="NP0-NN1">Keeler </w><w type="PRP">in </w><w type="AT0">the </w><w type="NN1">film </w><w type="NN1">Scandal </w><w type="CJC">and </w><w type="AT0">the </w><w type="NN1">pulpit </w><w type="VVN">used </w><w type="PRP">by </w><w type="NP0">Peter </w><w type="NP0">Sellers </w><w type="PRP">in </w><w type="AT0">the </w><w type="NN1">film </w><w type="NN2">Heavens </w><w type="AV0">Above </w><w type="VBB">are </w><w type="PRP">among </w><w type="AT0">the </w><w type="NN2">items </w><w type="TO0">to </w><w type="VBI">be </w><w type="VVN">sold </w><w type="PRP">in </w><w type="AT0">an </w><w type="NN1">auction </w><w type="PRF">of </w><w type="NN1">stock </w><w type="PRP">from </w><w type="AT0">the </w><w type="NN1">country</w><w

type="POS">'s </w><w type="AJS">largest </w><w
type="NN1">supplier </w><w type="PRF">of </w><w
type="AJ0">theatrical </w><w type="CJC">and </w><w
type="NN1">television </w><w
type="NN2">props</w><c type="PUN">.</c></s>

Figure 8: Corpus uith part-of-speech tags

As far as *Collocate* is concerned, this sample text is well-structured and hence can be manipulated easily for the user's convenience. For our purposes, we can divide the file contents into three different types of information: (i) tags (or normal tags) illustrated above by <text decls=…> and ; (ii) part-of-speech tags, which here precede the word they categorise and which appear in a form such as <w type = NN1>; and (iii) the words themselves (e.g., *TV*). As described below, it is possible to indicate in TAG SETTINGS the form of these different annotations and once we have done that, the program will help us manage the complexity of the corpus by, for example, suppressing the normal tags and part-of-speech tags so that the mark-up doesn't appear at all in the display of the corpus (or in the search results). Suppressing the tags in this way will cause the excerpt from the BNC shown above to be displayed as follows.

suppress tags

Starring role for TV and film props
By Robert Shrimsley
THE couch cavorted upon by Joanne Whalley in her portrayal of Christine Keeler in the film Scandal and the pulpit used by Peter Sellers in the film Heavens Above are among the items to be sold in an auction of stock from the country's largest supplier of theatrical and television props

Figure 9: Corpus with tags suppressed

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 10) 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.

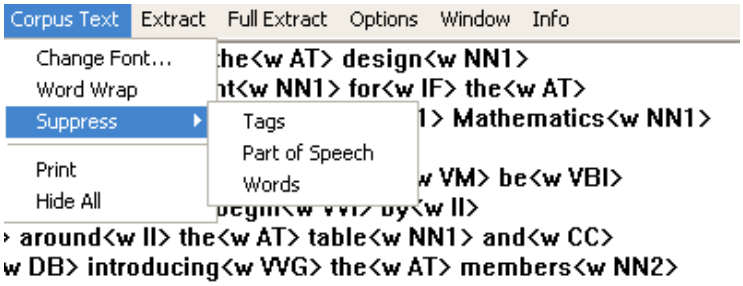


Figure 10: Suppressing tags

To fully exploit the information in a corpus it is often necessary to work with tags; however, it is possible to use *Collocate* without knowing about all the details related to tagging discussed in the remainder of this section and it may be that you will mostly work with untagged texts. You may therefore want to move to the next section and come back to the tag-related information discussed below once the basics of searching and extracting data have been mastered.

3.3.1 Indexing

indexing

When a corpus is being loaded, *Collocate* carries out some pre-processing and indexing of the files in the corpus. This process makes use of information about the form of mark-up or tags entered in TAG SETTINGS. Since there is no single, conventional way of indicating mark-up, it is necessary to tell the program about the format of the tags used in the corpus. Once this is done, the program is able to distinguish the mark-up or tags from the text itself, which ultimately makes it easier for the user to analyse the corpus.

We discussed above the use of the information in TAG SETTINGS for suppressing the display of different components of the corpus files. In addition, since the calculations involved in *Collocate* are computationally quite intensive, it is necessary to create an index of the corpus so that the calculations can be performed in a reasonable amount of time.

3.3.2 Normal Tags

To set the form of tags, we select TAG SETTINGS from the FILE menu and choose NORMAL TAGS. 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 4 below.

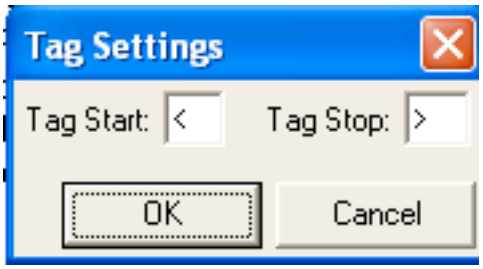


Figure 11: Setting the tag format

Once this information is entered and the files have been indexed, *Collocate* will distinguish between mark-up and text to the extent shown in the following table.

Action	**Text/Tag Distinction**
Word/Phrase Search	Yes
Regex Search	Yes
Word/Tag Search	Yes
Sorting	Yes
Collocate Frequency	Yes
Display/Suppress	Yes
Word Count	Yes

Let us briefly examine a couple of options presented in the table. The simple WORD/PHRASE search in EXTRACT distinguishes between text and mark-up, which means that a search for **name** will not find *name* occurring in the tag <*name*> but will find instances of *name* in the text itself. In the WORD/TAG search,, described in more detail in Chapter 7, the & is used to distinguish words and tags. Thus strings before the & refer to words (e.g., head&) and strings after the & specify

tags (e.g., &NN1). Of course, it is possible to specify word and tag together (.e.g., head&NN1).

3.3.3 Part of Speech Tags

speech NN1

We select TAG SETTINGS and this time choose PART OF SPEECH 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, as in the *BNC* files, as *<w AT0>the*, then we select OUTSIDE WORD (see Figure 5) and enter <w type = and <c type =(for punctuation) in TAG START and > in TAG END. In addition, since in the BNC the tag precedes the word it classifies, we must select the BEFORE WORD box.

The settings shown in the dialogue box below are appropriate for the *Corpus of Spoken Professional American English*.

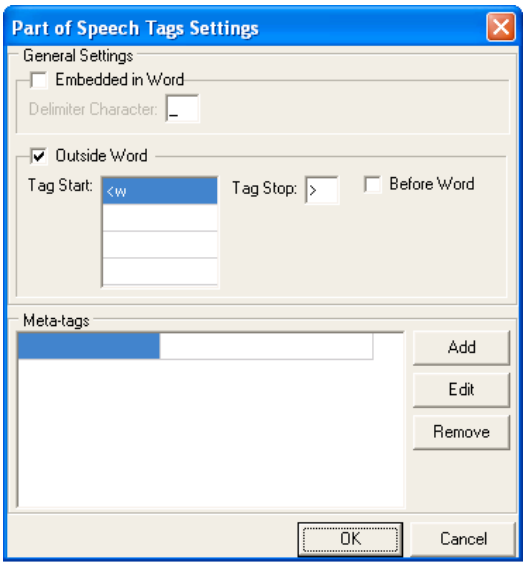


Figure 12: Defining POS tags and meta-tags

3.3.4 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 articles such as *a* and *the* and determiners such as *this*, then in the META-TAGS area of the part of speech tag settings, we choose ADD and in the resulting dialogue box we add a name for the meta-tag, Det-Art, and in the CONTENTS box, we simply enter AT0 and DT0, 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.

3.4 Loading and unloading a corpus

Let us examine the options for loading a corpus, that is, making one or more text files available for processing by *Collocate*. This operation is initiated by choosing LOAD CORPUS FILE(S) from the FILE menu (Figure 7) or by issuing the CTRL-L command.

When the LOAD CORPUS FILE(S) command is given, 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 (Figure 13).

multiple files

Further control over the display of file names is accomplished by entering a combination of characters and * in the box FILE NAME in the lower part of the dialogue box. Here it is possible to enter a string such as **g*.spo** (and press the return key) in order to display just those files that start with a G and have the extension SPO. 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). Clicking on OPEN (or pressing Return) loads the selected files into *Collocate*, making them available for searching. As discussed above, the program scans the files and creates an index.

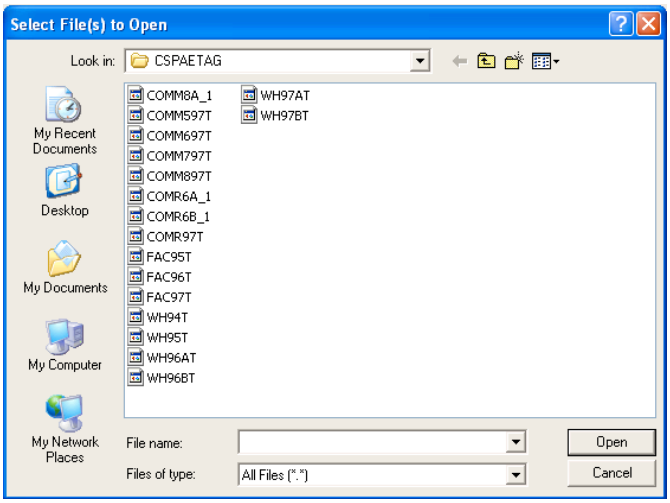


Figure 13: Selection of files comprising the corpus

size limit

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 processing the data, *Collocate* actually swaps chunks of text in and out of memory, with the result that the program should be able to handle any size of text.

To load files contained in different directories, the load file process must be repeated; this procedure adds files to the corpus and does not involve the removal of files already available.

Once the files are loaded, the index of words in the corpus is created.

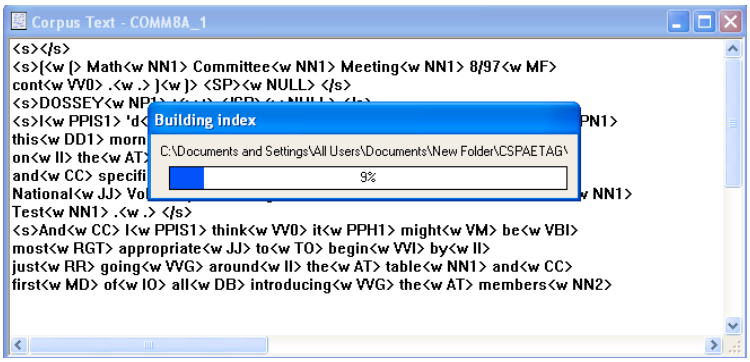


Figure 14: Indexing procedure

Once a corpus is loaded and the index created, some new menu items related to the analysis and display of the text appear on the menu bar. The menus now present are FILE, CORPUS TEXT, EXTRACT, FULL EXTRACT, OPTIONS, WINDOW and INFO. In addition, looking at the screen in Figure 15 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 and word type count for the corpus. Any of the corpus files can be examined using the scroll bars or the PageDown, End keys, etc.

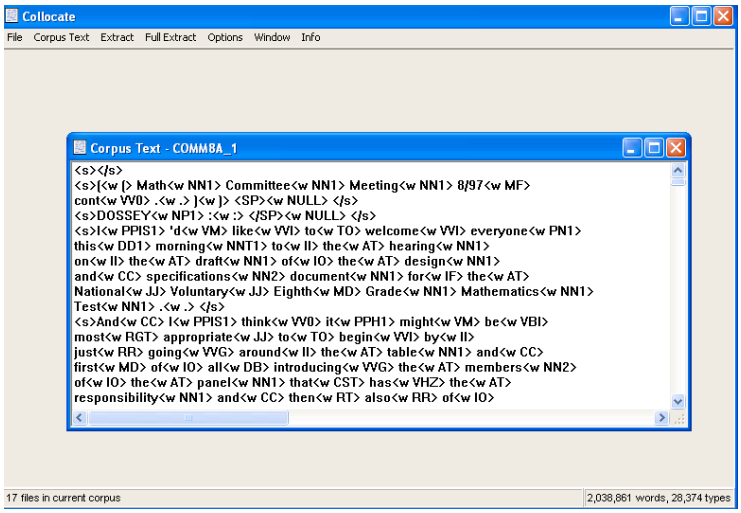


Figure 15: View of a corpus file

If several corpus files are open and the screen is too cluttered, select HIDE ALL from the CORPUS TEXT menu to close all the corpus text windows. Even if you "hide" all the corpus files, they are still available for searching and so EXTRACT and the other menus remain visible. (The CORPUS TEXT menu, however, is not available when all the files are hidden.)

word wrap

If the text in the corpus window is not easily viewed because the end of the lines extend past the right edge of the window, then simply select the WORD WRAP option from the CORPUS TEXT menu.

Before initiating a collocate 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. (See Section 3.6 for more detail.) Any (or all) of the loaded files can be viewed by choosing VIEW CORPUS FILE/URL from the FILE menu (or CTRL-V). This command brings up a window containing a list of the names of the files that are loaded. Choosing one or more files from the list will open a window for each selected file enabling the text contents to be viewed.

Collocate 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 regular file type (such as a Word or RTF document), which contain boldface and other formatting information tied to the particular word-processor. Whether or not the regular files actually contain formatting (boldface etc.), they must be saved as text-only files before they can be analysed 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.

gárblëd text 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 *Collocate* 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.

3.5 Displaying the corpus files

view corpus Once the corpus files are open, the CORPUS TEXT menu, which contains commands relating to the display and printing of the corpus file, becomes available. In many situations the corpus files will not be examined directly, but if they are to be examined, the commands CHANGE FONT and WORD WRAP control the size and positioning of the text on the screen. The SUPPRESS command leads to three choices: TAGS, PART OF SPEECH, and WORDS. Choosing one or two of the three options will lead to the non-display of the selected text objects. The suppression of the tags was illustrated above in Section 3.3.

3.6 Changing the corpus

unload corpus 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, OPTIONS and INFO menus available. All other windows and menus are closed.

remove files The selective removal of one or more files is accomplished by selecting VIEW CORPUS FILE from the FILE menu, selecting one or more files and clicking on the REMOVE button.

add files To add new files to the existing corpus, simply repeat the LOAD CORPUS FILE(S) command.

3.7 Printing the corpus file

Setting up the printer is accomplished by the command PRINT SETUP in the FILE menu.

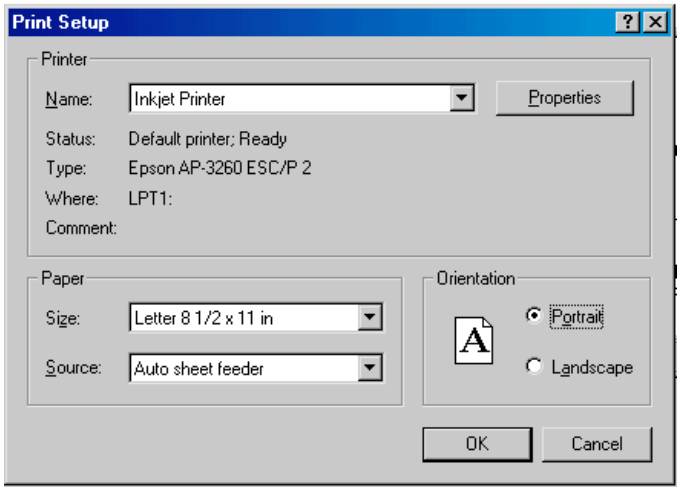


Figure 16: Print set-up

(To specify the default printer, use the control panel.) If you want to print the current corpus file, the one in the active window, select PRINT from the CORPUS TEXT menu or enter CTRL-P. Since the corpus file may be large and the print command may be selected in error, a dialogue box appears, giving an estimate of the size of the file. The print job can then be cancelled if necessary.

4. USING WORKSPACES

Summary: Saving a workspace avoids the need to reload corpus files and re-index the files.

avoid reload

As is clear from the previous section, loading and processing a corpus can take some time. Since it is often the same set of corpus files which are loaded each time *Collocate* is started, it makes sense to freeze the current state of the program, at will, so that the analysis of a corpus can be continued at any time. This is the idea behind a workspace. A workspace is saved as a special *Collocate* Workspace file (.cws), which can then be opened at any time to restore *Collocate* to its previous state, with the corpus loaded ready for searching. Windows containing search results and frequency data are, however, not included in the saved workspace. (Only the search histories are saved.)

4.1 Saving a workspace

A workspace—the current corpus and settings of *Collocate*—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. (Generally, it is only the links to the corpus that are saved, not a copy of the corpus itself.)

Warning: Keep corpus files and workspace files separate. It is advisable to create a folder (called Workspaces) in which 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.

SAVE ON EXIT

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

4.2 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 *Collocate* with the corpus loaded is to double-click on a saved workspace file.

5. EXTRACTING COLLOCATIONS: WORD/PHRASE SEARCH

Summary: Select WORD/PHRASE from the EXTRACT menu and enter a search string. The parameters controlling the search are listed in OPTIONS. The results can be saved or printed.

5.1 Using Extract

It is possible to extract all the two-word phrases in a corpus (see Chapter 8), but we will start off with a simple search for a specific item and extract two-word collocations associated with the word *test*. To do this, we select WORD/PHRASE from the EXTRACT menu or enter CTRL-W, (shown in Figure 17)

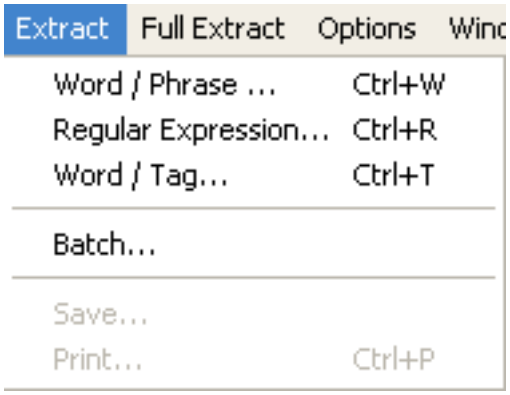


Figure 17:Extract menu

In the text box at the top of the dialogue box that appears (Figure 18) type in the search term **test** and click on OK (or press enter). Note that the span is set to 2.

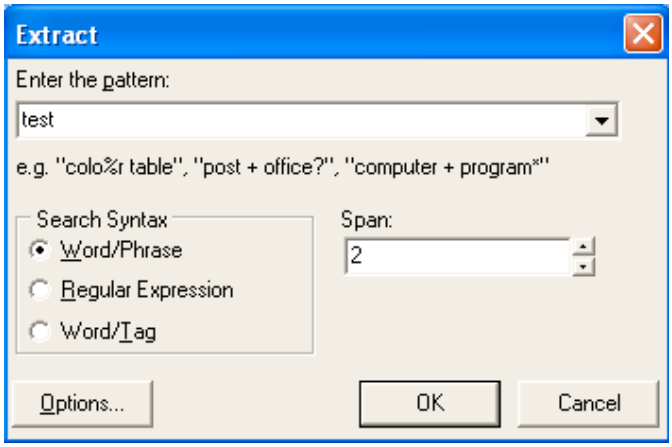


Figure 18: Extract dialogue box

Note: The parameters of the search are determined by the settings in force in OPTIONS. The CONSTRAINT OPTIONS cover such things as whether the hyphen is treated a word boundary. The COUNT OPTIONS control the statistic used to calculate collocational strength and the minimum frequency for the word pairs or bigrams.

results

The results should appear quite quickly in the form of a table. (Note that some of the words in these results are quite long due to the use of hyphens. As noted above, whether hyphens are part of a word or a word boundary is controlled by the settings in CONSTRAINT OPTIONS.)

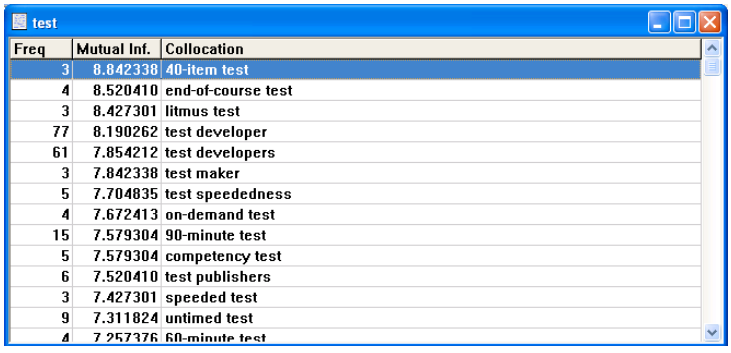


Figure 19: Two-word collocations containing test

The information is presented more clearly in the table in Figure 20.

Freq	Mutual Inf.	Collocation
3	8.842338	40-item test
4	8.520410	end-of-course test
3	8.427301	litmus test
77	8.190262	test developer
61	7.854212	test developers
3	7.842338	test maker
5	7.704835	test speededness
4	7.672413	on-demand test
15	7.579304	90-minute test
5	7.579304	competency test
6	7.520410	test publishers
3	7.427301	speeded test
9	7.311824	untimed test
4	7.257376	60-minute test
4	7.141899	norm-referenced test
3	7.105373	timed test
20	7.098177	45-minute test
3	6.967869	aptitude test

Figure 20: Bigrams containing test

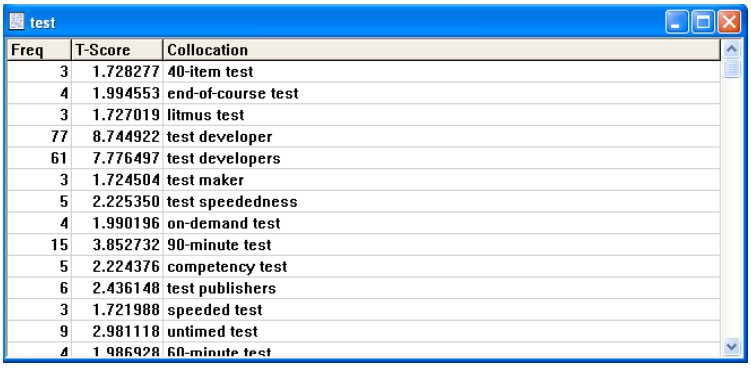
The table shows the frequency of occurrence of the two-word phrase in the first column, followed by the statistic, Mutual Information (MI) in this case, and in the third column the phrase containing the word *test*. You can see that the results are ordered according to the MI score rather than raw frequency. Notice too that the phrases produced in this list appear to consist of good collocations rather than random phrases that happen to include the word *test*.

This example provides a good introduction to the basic usage of the software. Let us now consider the statistic used and then look at some sorting options.

With the creation of a results window, two additional items appear in the menu bar: DISPLAY and SORT.

Figure 21: Display menu

The display menu shows that frequency is to be displayed, along with the Mutual Information score. Other potential statistical scores are T-SCORE and LOG LIKELIHOOD. (There is also the command CHANGE FONT, which can be used to make the font larger, etc.) If we choose t-score, then the relevant t-score for the existing results is shown (Figure 22).



Freq	T-Score	Collocation
3	1.728277	40-item test
4	1.994553	end-of-course test
3	1.727019	litmus test
77	8.744922	test developer
61	7.776497	test developers
3	1.724504	test maker
5	2.225350	test speededness
4	1.990196	on-demand test
15	3.852732	90-minute test
5	2.224376	competency test
6	2.436148	test publishers
3	1.721988	speeded test
9	2.981118	untimed test
4	1.986928	60-minute test

Figure 22: Bigrams with t-score

Notice that the order of the results stays the same, essentially ordered by MI score, but the t-score value is shown rather than MI. What is if we wish to re-order the results according to t-score? Let us look at sorting options.

sorting

To sort according to the current statistic, t-score, we go to the SORT menu and select SCORE for the primary sort and FREQUENCY for the secondary sort. This means that the results will be ordered firstly by score (t-score in this case) and for those instances where several phrases are associated with the same t-score, the results will be sorted according to frequency.

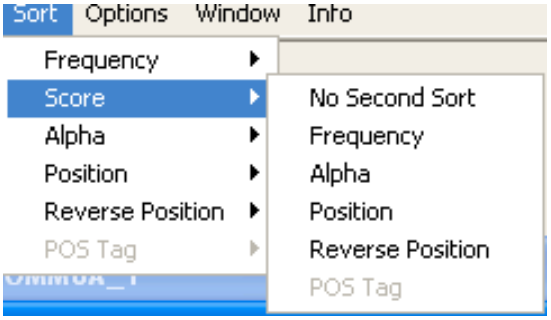


Figure 23: Sort menu

The results of this sort are shown in Figure 24 and the top ranking items are shown in Figure 25.

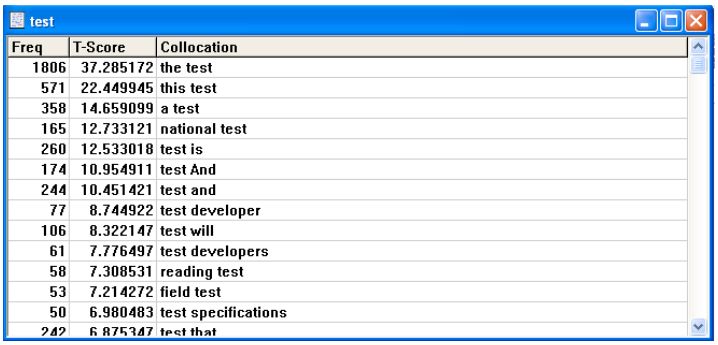


Figure 24: Bigrams sorted by t-score

Freq	T-Score	Collocation
1806	37.285172	the test
571	22.449945	this test
358	14.659099	a test

165	12.733121	national test
260	12.533018	test is
174	10.954911	test And
244	10.451421	test and
77	8.744922	test developer
106	8.322147	test will
61	7.776497	test developers
58	7.308531	reading test
53	7.214272	field test
50	6.980483	test specifications
242	6.875347	test that
47	6.725996	math test
65	6.581395	test So
42	6.270967	test security
37	6.031902	voluntary test

Figure 25: Table of bigrams containing test sorted by t-score

As you can see, the results are mixed. Some of the phrases look like good collocations (*national test, test developer*), but other word pairs are not good candidates for collocations such as *the test*. There are a couple of ways to approach these results. One is to just assume that the user will inspect the results using several different statistical measures and ignore those instances that are not of interest. Other approaches are to use a stop list (see page 60) or to use POS tags (see Chapter 7).

5.2 Using wildcards

Let us consider a search to extract patterns based on SPEAK where SPEAK stands for the word family or lemma speak, i.e., *speak, speaks, speaking, spoke,* etc. To approximate a lemma, we have to make use of wildcard characters in this simple search. (We will discuss more precise and sophisticated alternatives in Chapter 6.)

* wildcard

The first wildcard character is the asterisk *. This wildcard in a search string will match zero or more characters and therefore 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. There are a couple of possibilities for defining a search string for forms of *speak* using * but the most straightforward is the search query **sp*k***, which will find all words starting with *sp* and having a *k* in them somewhere. Notice that this search string will in fact find words like *spank* and *sprocket,* as well as words that we are looking for such as *spoken*.

We can initiate a search from the extract menu and enter the search string **sp*k***. At the right end of the text box in the search dialogue box is a drop-down arrow. Clicking on this arrow reveals a list of previous search strings. If required, one of these search queries can then be reselected to be used in a new search. The program will store up to 20 previous searches in this list.

? wildcard

The wildcard character ? stands for a single alphanumeric character. We might use this to capture *himself* and *herself* by using the string **h??self**.

% wildcard

The special character % represents zero or one character. Using ? and % for members of the SPEAK lemma, we can enter the search term **sp?%k%%%**. The combination of **?%** following **sp** matches at least one and at most two characters, so it will find forms based on the stem *speak*, which has two vowels between *sp* and *k*, and forms based on the string *spoke*, which has only one vowel separating the stem consonants.

* alone

It is a feature of *Collocate* 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 word/phrase search based on * by itself—or you could try it. This is one way of producing n-grams. We will return to this approach in Chapter 8.

If you want to search for ? as a literal character, you will have to substitute a different symbol for the wildcard in the constraints options CONSTRAINTS OPTIONS dialogue box. (See Chapter 10.)

Below is a summary of the wildcard characters relevant for a WORD/PHRASE SEARCH (and for a WORD/TAG SEARCH) :

*	matches zero or more alphanumeric characters
%	matches zero or one character
?	matches exactly one character

5.3 Extracting larger collocations

To extract chunks associated with the word *test* that are longer than 2 words, we select WORD/PHRASE from the EXTRACT menu or enter CTRL-W, and set the span parameter to any number between 3 and 12. The procedure is basically the same as searching for two-word phrases. However, t-score and log likelihood are not used; only Mutual Information is calculated for phrases longer than two words.

adjacency — If we enter the search term **the test**, then collocations of the specified span must contain *the* followed immediately by *test*.

> — If we wish to find instances of TAKE and *test*, but exclude *test takers*, we can use the ordering symbol > and specify **tak* > test**. Using this search pattern with a span of 3 yields the following results.

Freq	Collocation
65	take the test
24	taking the test
21	take this test
15	taking this test
6	takes the test
4	taken the test
4	taking a test
4	take a test

Figure 26: Trigrams based on tak > test*

Note that there are no instances like *test taken at* where *taken* follows *test*.

\+ — If we don't want to restrict the order or position of *tak** and *test*, we specify the search term as **tak* + test**.

Summarising, we can note the following types of search string:

A B	A must be adjacent to and precede B
A > B	A must precede B
A + B	A and B can occur in any order

5.4 Batch search

Using BATCH search allows multiple search items to be extracted in one go.

5.5 Saving the results

To save the contents of the results window, select SAVE … from the EXTRACT menu. A dialogue box appears and the name of the results file can be entered.

5.6 Printing the results

The results can be printed by entering CTRL-P or selecting PRINT from the EXTRACT menu. The appropriate results window must be active for printing to occur.

6. EXTRACTING COLLOCATIONS: REGULAR EXPRESSION SEARCH

Summary: Select REGULAR EXPRESSION from the extract menu and enter a search string using regex (boolean) syntax.

In the previous chapter we described the use of wildcard characters to capture the lemma SPEAK. A REGULAR EXPRESSION search provides a powerful search syntax which will give us much more precise searches. Essentially, regex searches allows search queries to contain 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 WORD/PHRASE search.)

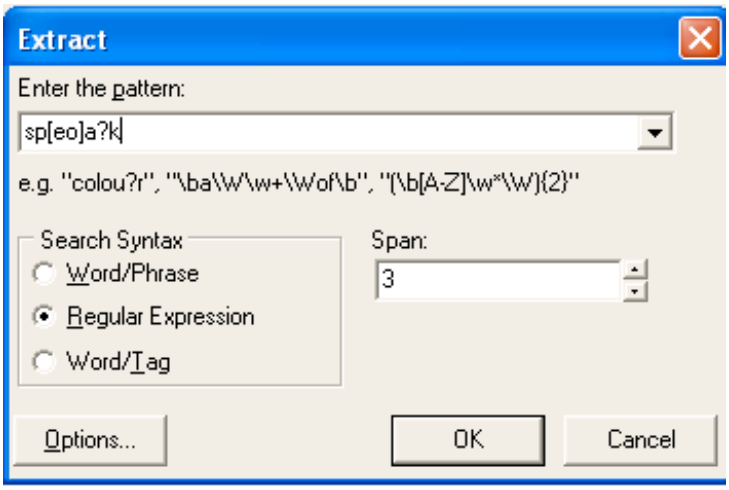


Figure 27: Extracting phrases using a regex expression

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 particularly useful regular expressions, you might want to save those along with appropriate descriptions in a file so that search strings can be retrieved rather than recreated.

AND, OR, NOT To initiate a regular expression (regex) search, select the radio button labelled REGULAR EXPRESSION located on the left side of the EXTRACT 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.

string search Unlike in many other implementations, regex searches are not string searches. In Collocate if *let* is specified as the search term, then *letter, toilet* and *complete* etc. will not be potential matches for this search term.

\<word\> If word boundaries must be indicated explicitly 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\>**.

\bword\b A more general and slightly easier way of specifying a word boundary is by using the meta-character **\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.

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.

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 *Collocate*.

[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 OPTIONS. Conversely, \D is any non-digit and \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 **speak|speaks|spoke|spoken** 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 **spea** followed by **k** or **s**.)

a? optional a Another way to perform this lemma search is to specify the query **speaks?|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.)

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*.

scope If we wish to search for *speak* followed by either *to* or *with*, and we try the string **speak\Wto|with**, 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 **speak\W(to|with).**

NOT ^ If you want to search for *test*, but omit preceding words containing *t*, you can use the not operator and specify the search string as **[^t] test**

\1 again Let us look at a rather different search query: **\w*(.)\1\w***. The \1 option is essentially a way of repeating whatever is specified in the preceding parentheses. The search string above will find words with a repeated letter in them. (If there is more than one set of parentheses, then \1, \2, can be used to reference the corresponding groupings.)

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.

.	any character
[a-z]	any lower case letter
[0-9]	any number
[aeuio]	any vowel (in English)
[^lr]	not l or r
\b[a-z]+\b	lower case word
\b[A-Z][a-z]*\b	word with upper case initial letter
\b[0-9]+\b	number
\b	word boundary
\B	word non-boundary
\d	any digit
\D	any non-digit
\w	any word character (= alphanumeric)
\W	any non-word character (defined by word delimiters)
\s	whitespace
\S	non-whitespace
*	zero or more

+	one or more
{n}	n instances of previous expression
{n,m}	from n to m
{n,}	at least n

7. EXTRACTING COLLOCATIONS: WORD/TAG SEARCH

Summary: If a corpus contains part-of-speech tags, the tags can be used to guide the retrieval of collocations. Select the Word/Tag Option from the Extract menu.

7.1 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**.

wildcards % *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*.**

7.2 Working with POS tags

A corpus in which each word is annotated with part-of-speech tags provides a richer, more structured database allowing targetted searches and opening up the possibility of using POS tags to constrain collocations. The structure of the tagging must be specified before the corpus is loaded. See Section 3.3.

It should be noted that a regular WORD/PHRASE search can be used on a tagged text. It is also possible to use a REGULAR EXPRESSION search, but only to search for the words. The tags are opaque to regex searches.

The third option in the EXTRACT dialogue box is in the advanced search dialogue box is WORD/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*.

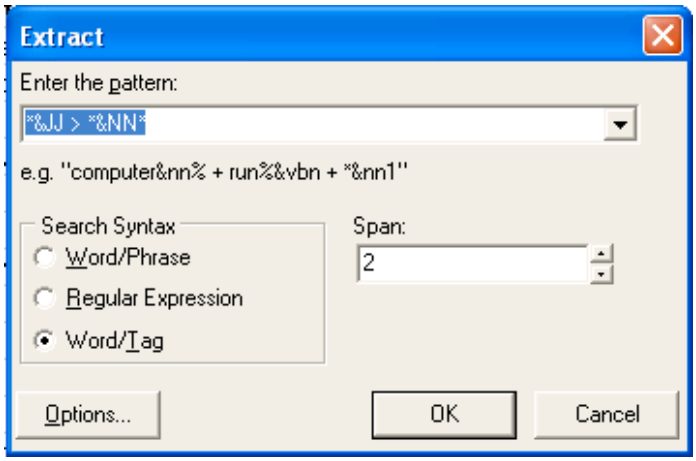


Figure 28: Specifying words and POS tags in a Word/Tagsearch

old&JJ

The search query in a WORD/TAG can contain any permutation of words, tags, or words and tags. The basic syntax of the search query is **word&tag**. Note that this is the case whatever the ordering of words and tags in the corpus itself. 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 under search terns in CONSTRAINT OPTIONS.

7.3 Searching for noun compounds

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

7.4 Searching for words

ignoring tags

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&***.

8. FULL EXTRACT: N-GRAMS AND COLLOCATIONS

Summary: The searches in full extract present a general picture of the corpus as a whole. It is possible to produce an n-gram list for the corpus or create a list of collocations of a certain size taken from the corpus.

8.1 N-gram

Selecting N-GRAM from the FULL EXTRACT menu produces the dialogue box shown in Figure 29. The user can enter the size of the n-gram, and, where appropriate, the option of including POS tags.

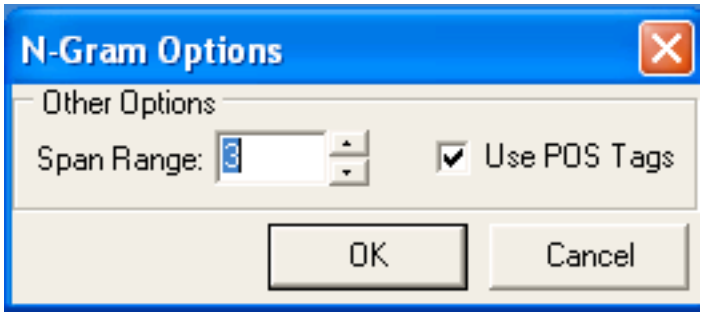


Figure 29: Setting the n for n-grams

The results of such a search will look like the example in Figure 30.

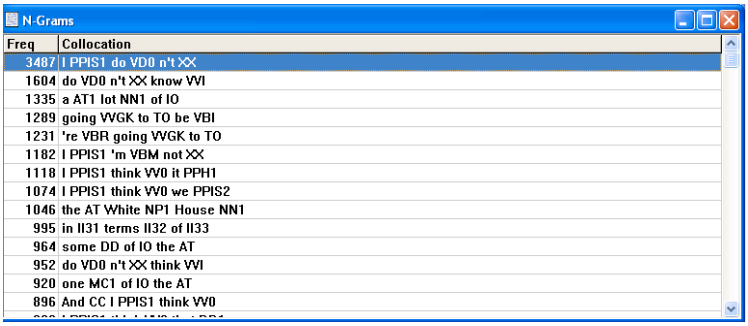


Figure 30: Trigrams with POS tags

The MI score can be calculated, but the resulting score is based only on the words themselves and not on the word-tag combinations.

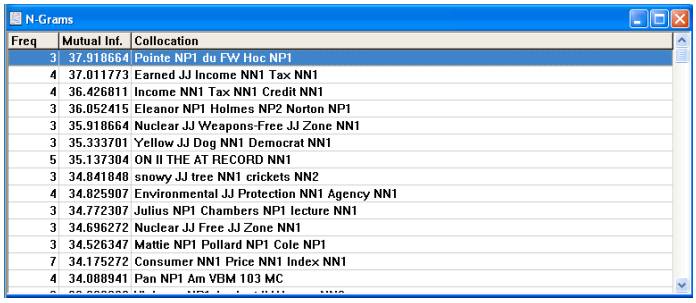


Figure 31: Trigrams sorted by MI

It is possible to sort the results based on the POS tag, as shown in Figure 32.

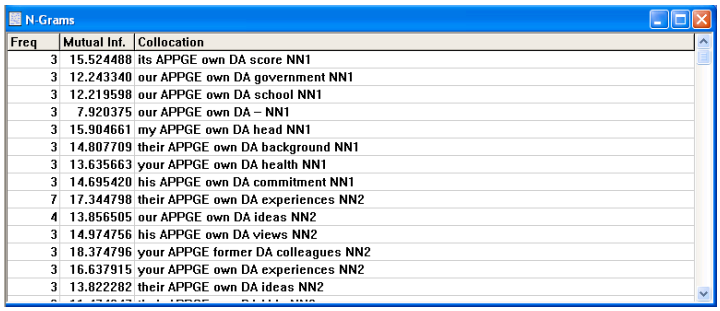


Figure 32: Trigrams sorted by POS tags

An alternative to using the n-gram command is to use the regular word/phrase option and enter * plus some span size. This provides some more statistics if a span of 2 is employed.

8.2 Extract

The second option is EXTRACT. This command processes the whole text and display candidate collocations based on the criteria specified by the user. See Figure 33.

The software creates a list of bigrams and filters those according to thresholds set by the user. The remaiing bigrams form the basis of potential trigrams, which are again filtered according to particular thresholds. And from here the process continues to tetragrans, and so on, depending on the span range specified at the bottom of the dialogue box (Figure 33).

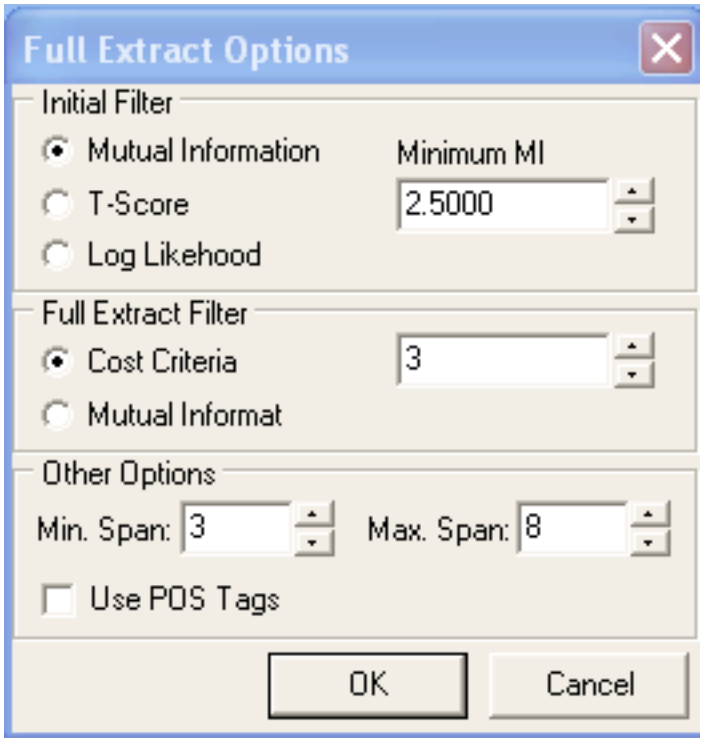


Figure 33: Settings for extraction of collocates from the corpus

The first choice to be made is the statistic and threshold that will be used to filter the bigrams. The options for the statistical test are one of the following: raw frequency, t-score,

mutual information, and log likelihood. One of these must be selected and a threshold value set.

The calculations that guide the inclusion of trigrams, tetragrams etc. are based either on the cost criterion (Kita et al) or Mutual Information. The appropriate measure must be selected and the threshold set.

Once the values are set and the user selects Okay, the program processes the whole corpus. First the bigrams are extracted, then on Pass One the other n-grams are selected and on Pass Two the n-grams are ranked. This process is computationally intensive and could well take a few minutes.

9. SORTING THE RESULTS

Summary: The results can be sorted to reveal patterns in the data. The basic sorting option involves choosing a primary and secondary sort order.

9.1 Sorting

The results displayed initially in frequency or score order. However, it is possible to re-sort the items in various ways. The menu shown in Figure 34 contains the primary ways in which the results can be sorted. An option is greyed out if it is not applicable to active results window. In the sort menu shown in Figure 34 all the options are available, apart from POS Tag. Once a primary sort parameter is selected, options for a secondary sort are presented (including No Second Sort), as illustrated in Figure 35.

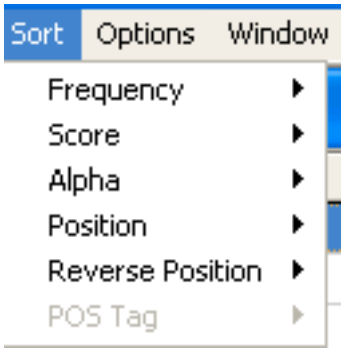


Figure 34: The Sort menu

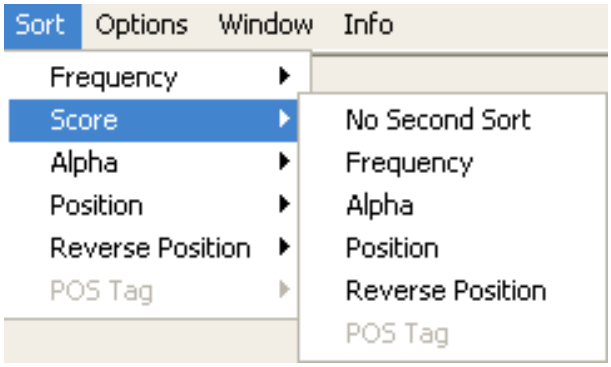


Figure 35: Sort menu showing primary and secondary sort options

The secondary sort comes into play when several lines have the same value according to the primary sort parameter. Those lines are then sorted using the secondary sort parameter.

9.2 Sorting on freqency

In many cases the results will be sorted in order of decreasing frequency.

9.3 Sorting on score

The term SCORE is used to cover any of the statistical measures used in Collocate: Mutual Information, T-score, Log Likelihood or Cost.

9.4 Sorting on alpha

Choose the sort parameter ALPHA will sort the results into terms of alphabetical order, which typically means punctuation, numbers, and then letters in alphabetical order.

9.5 Sorting on position and reverse position

If you extract three-word chunks containing the word *test*, then the word *test* itself may occur in first, second, or third position. Choosing POSITION will order the results so that phrases in which *test* occurs in first position are ordered before those in which *test* occurs in second position, and so on. Using reverse

position will place phrases which end in *test* at the top of the list of results.

9.6 Sorting on POS tag

Choose the sort parameter POS TAG will sort the results in alphabetical order of the part-of-speech tag, assuming that a tagged corpus is being used..

10. OPTIONS

Summary: Settings related to searches are located in CONSTRAINT OPTIONS and COUNTS OPTIONS dialogue boxes.

Before loading a corpus , the appropriate LANGUAGE should be selected and, if the corpus contains tags, then the form of the tags should be entered in TAG SETTINGS. The settings can be made using the FILE menu.

The behaviour of searches follows the values set in CONSTRAINT OPTIONS, shown in Figure 36.

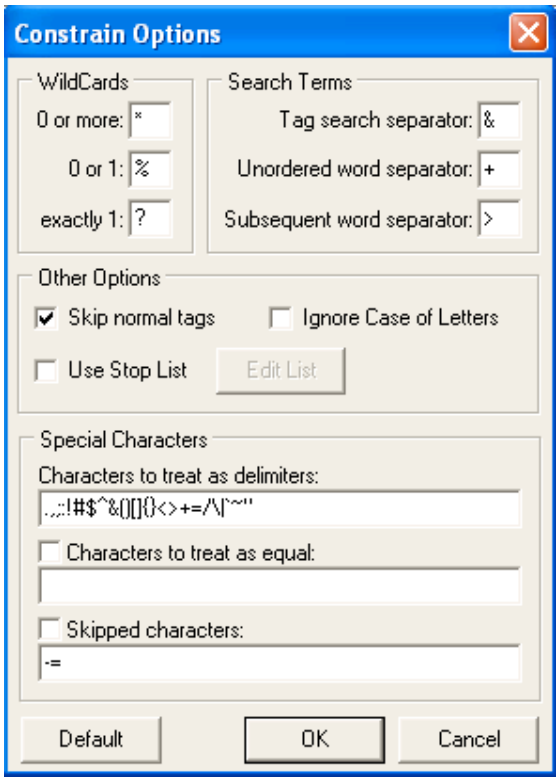


Figure 36: Constraint Options dialogue box

10.1 Wildcard characters and search terms

If may be necessary to use different characters to represent wildcard characters. For example, if ? is a wildcard character, it is not possible to search for ? as a question mark. Any suitable character can be used to represent a wildcard character; it is simply a matter of entering that character into the appropriate textbox in the Constraints dialogue box. The same goes for the search terms +, > and &.

10.2 Skip tags and stop list

The option to Skip Normal Tags means that the words with tags (as defined in tag settings) will not be included in the extraction results. In addition, it is possible to create a stop list containing common words such as *the, of, and*, etc. Collocations containing any of these words will then be ignored.

10.3 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.

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.

10.4 Delimiters: What is a word?

It is clear that a prerequisite for a word extraction 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).

/.:+word\#'{- Let's examine a couple of situations in order to illustrate the subtleties that you are likely to encounter. 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.

Collocate 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*.

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 add the apostrophe to the word delimiters, in the text box in CONSTRAINT OPTIONS.

factory setting

It is possible to restore the original "factory" setting for the list of word delimiters by clicking on the DEFAULT button.

10.5 Characters to treat as equal

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.

10.6 Skipping characters

skipp=ing

The skipped 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. 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.

10.7 Counts Options

The COUNTS OPTIONS dialogue box shown in Figure 37 controls the statistical method used in calculating the collocational strength holding between words in bigrams. The second option, minimal word frequency provides the minimum value for the display of bigram, trigrams, etc. in a results window.

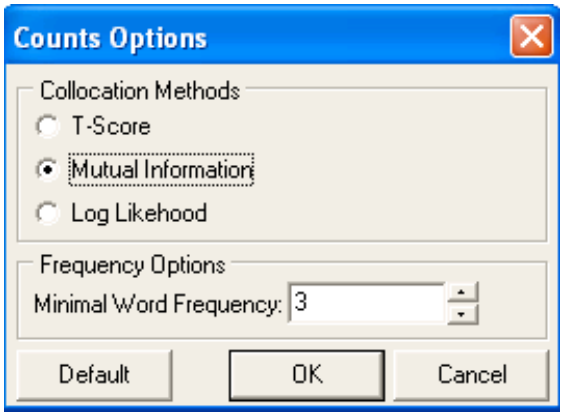


Figure 37: Counts Options dialogue box

11. 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 results is controlled by the DISPLAY menu.

11.1 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.

11.2 Changing Font,

large fonts

To change the font, font style, font size, or font colour, select CHANGE FONT from the DISPLAY menu. The system font can be changed using the LANGUAGE command in the FILE menu.

If the columns are not wide enough to display the figures correctly, grab the column divider in the heading and move it sideways.

11.3 Word Wrap

The display of text in the corpus file can be controlled by the WORD WRAP command in the CORPUS TEXT menu.

11.4 Frequency and statistical score

The choice of frequency and statistical information is controlled by the settings in the DISPLAY menu, as illustrated in Figure 38. If the collocation span is greater than 2, then the menu in Figure 39 is displayed.

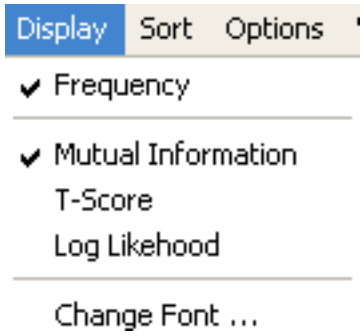


Figure 38: Display menu for bigrams

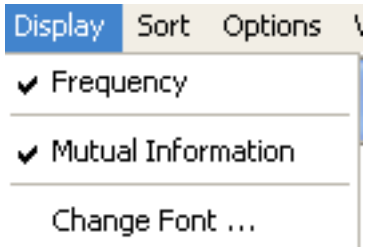


Figure 39: Display menu for trigrams etc.

INDEX