Manual
for the
Simple Corpus Tool

Version 3.0

Author:

Martin Weisser

December 2021

Contents

Figures

1 Introduction

The Simple Corpus Tool (henceforth SCT) is a research tool similar to AntConc that combines analysis and annotation functions. On the one hand, users can manually annotate corpora using a simple, easily readable form of XML, and, on the other, analyse them from a variety of perspectives, including concordancing, word list/n-gram, collocation, and keyword/-phrase analysis, as well as counting patterns across files. For all analysis modules that n-gram frequencies, including unigrams, the text is also automatically pre-segmented, thereby as much as possible avoiding the calculation of n-gram frequencies, collocations, or keywords/-phrases across sentence boundaries.

This new version is written in Python to ensure better usability and portability, but unfortunately I haven't been able to find a suitable tagger PoS module yet, so that there is currently no option for morphosyntactic annotation.

In addition to the basic annotation and analysis functionality, the tool also provides convenient ways for customising the startup options, the individual modules, and the XML resources required for carrying out annotations.

The individual features and options will be discussed and illustrated in the following sections, beginning with a brief overview of the tool. In general, though, all modules include many tooltips that should already help you understand how the program works without constantly needing to consult this manual.

2 Overview of the SCT Functionality

We'll begin our brief overview with an 'interface tour'. Figure 1 below provides a quick overview of the SCT interface as it appears after startup.

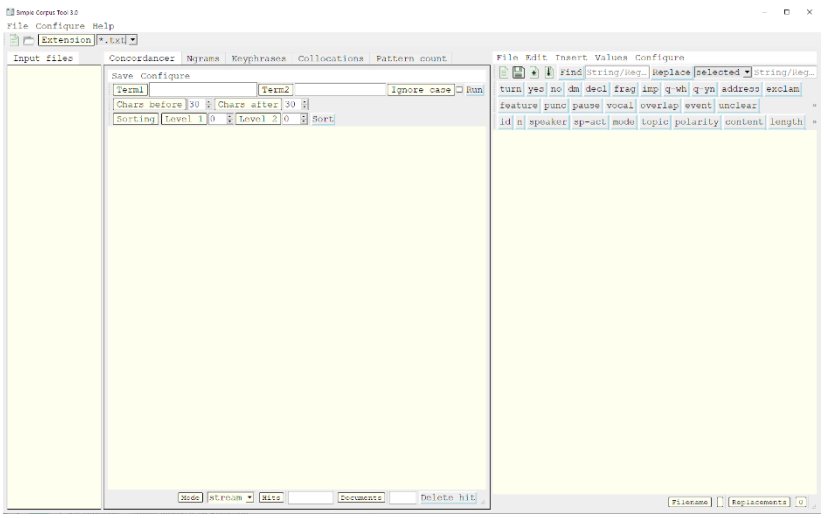


Figure 1 – the SCT Interface

The SCT provides facilities for loading a single file or selection of plain text files (.txt, .xml, .htm(l), etc.) into the 'Input files' workspace, seen on the left-hand side of the program window. By loading files from different locations, it's also possible to compile virtual corpora that essentially comprise list of these files which can then be reloaded.

Once one or more files have been loaded into the input files workspace, different actions can be run on these. These actions may consist in pre-processing or annotating files after loading them into the built-in editor, which is triggered by double-clicking the relevant filename, automatically (re-)numbering turns in dialogue data, inserting punctuation tags, etc., or analysing their features in various ways, specifically running concordances on them via the 'Concordancer' tab, creating word/n-gram frequency lists via the 'Ngrams' tab, lists of keywords/-phrases via the 'Keyphrases' tab, lists of collocates via the 'Collocations' tab, and counting patterns via the 'Pattern count' tab.

The tabs for these analysis modules can be seen in the middle section of the interface, while the right-hand section contains the XML editor that can be used to create and edit XML and plain text files, and which is also linked to the concordance module.

When files are loaded, the extension needs to be specified in the 'Extension' dropdown list to filter the input files according to their type. This option is also configurable via the startup-options file.

The SCT also provides options for editing many of the resources required to perform the analyses/annotation, such as e.g. customising the tag-related files to a particular project/corpus, etc.

As the SCT assumes the encoding of files to be UTF-8, text in virtually any language can be edited and processed with it, provided that the PyQt toolkit used for creating the GUI supports it.

3 File Handling

The SCT 'File' menu comprises the options depicted in Figure 2.

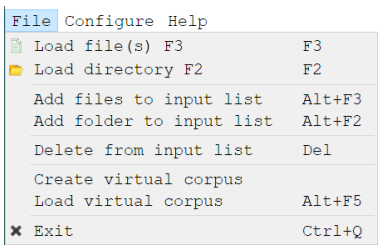


Figure 2 – the SCT 'File' menu

The menu allows you to select a whole directory (F2), individual files (F3), or a virtual corpus (Alt + F5) as input sources, add additional files later (Alt + F3; Alt + F2), perhaps in order to

create a virtual corpus, remove selected files from the workspace (Del), as well as exit the SCT (Ctrl + Q).

For loading standard corpus files, there are two options, either for choosing individual files or a whole folder of files. The default file location searched by the program is the 'data' folder in the SCT program folder, unless you specify a different path in the SCT configuration file. Depending on the choice made in the 'Extension' box on the right-hand side of the menu bar, the relevant file-type option will be pre-selected, but can of course be changed manually. Once one or more files have been loaded, they will be listed in the 'Input files' window, where double-clicking them will open the file in the built-in editor for viewing or editing, as well as making them available for processing from within any of the modules.

To exclude files listed in the workspace window from processing, select the relevant files using standard selection mechanisms, i.e. click for single files, Shift + click for consecutive files, or Ctrl + click for multiple non-consecutive ones, and then either choose 'Delete from input list' from the 'File' menu or press the 'Del' key.

To create a virtual corpus, which is essentially a list of file paths, you first load a set of files, usually from a folder, then add other files or folders, and then save the definition for the virtual corpus through 'File → Create virtual corpus'. If you simply need a subset of files from a folder, you can also open that folder – or an exiting virtual corpus –, remove any unwanted files as described above, and then save the definition. Virtual corpora are stored in the SCT's 'conf' folder and carry the extension '.vco'.

Once you've run any type of analysis, you can save the output from the relevant module to either plain text or an Excel file via the relevant option on the 'Save' menu. The result files will normally be saved to the 'results' sub-folder inside the program folder.

All input and output files are assumed to be UTF-8 encoded, and will also be saved in this encoding again. If your files are in a different encoding, you should normally convert them first.

If SPAADIA/DART data or data using similar XML tags are edited in the SCT editor, lines containing syntactic tags will be colour-coded.

As is to be expected, 'File → Exit' or pressing 'Ctrl + Q' closes the tool.

4 Running Analyses

The 'Run' button inside each module is context-sensitive, so it will trigger the relevant processing option depending on which tab is currently selected. Each module tab contains its own menu and tool bars that contain a multitude of easily accessible analysis options, while the 'Keyphrases' and 'Pattern count' tabs also contains two sub-windows, the top one to display the counts, and the bottom one allowing you load a reference corpus list for keyword/-phrase analysis, or to define the feature names and patterns to be counted in all files for pattern counting.

4.1 Concordancing

Concordance results will be displayed in the window below the 'Concordance' tab. Figure 3 shows the 'Concordance' tab after running a concordance on the modal verb *should* in the FLOB corpus.

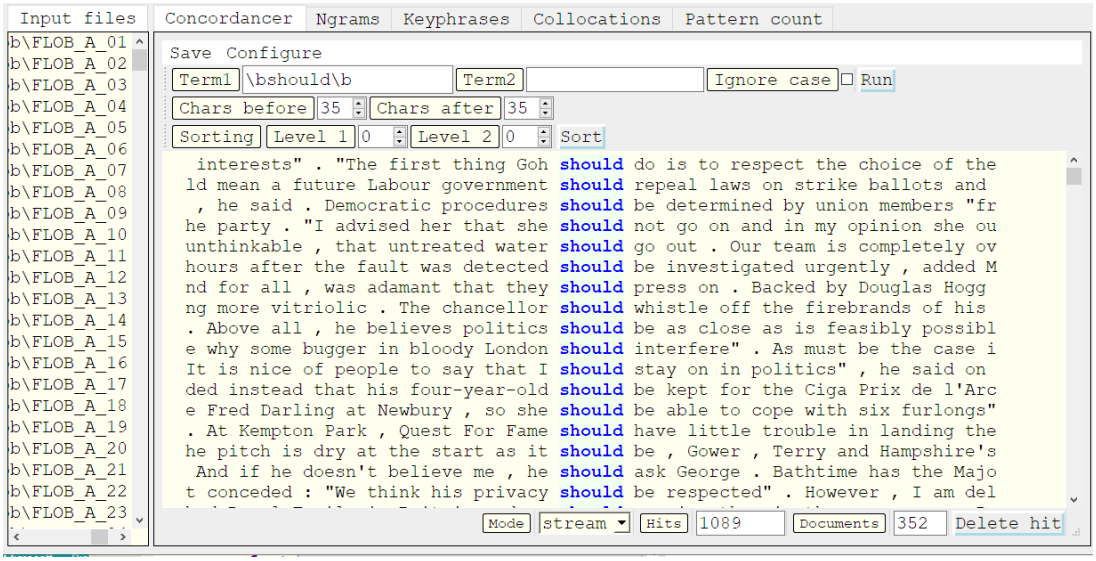


Figure 3 – the Concordancer tab

As the above illustration shows, for concordancing, either one or two search strings can be specified via the respective input boxes for 'Term 1' and 'Term 2'. The box for 'Term 1' always needs to contain a term, while the other one can be left empty. As an alternative to clicking the 'Run' button, pressing the shortcut key 'Ctrl + r' will also trigger a search. It's also possible to define a number of frequently used options for both terms in the configuration, so that the input boxes will offer auto-completion for them for quicker access.

Search strings should ideally be entered using (Python) regular expressions, as entering simple word forms only will generally also find longer words containing them. In the illustration in Figure 3, only exact matches for (lowercase) *should* are found, but not negated forms of the modal verb, due to the word boundary on the right-hand side of the term. Ticking the 'Ignore case' option – depicted on the top right in Figure 4 – would of course also find forms containing an initial capital or those in all-caps.

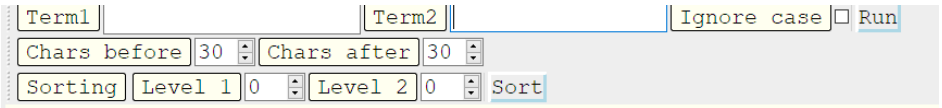


Figure 4 – 'KWIC' mode options

The number of hits and document frequency are shown in the respective entry boxes next to ‘Hits’ and ‘Documents’ in the status bar after a concordance has been run, and can also be copied from there.

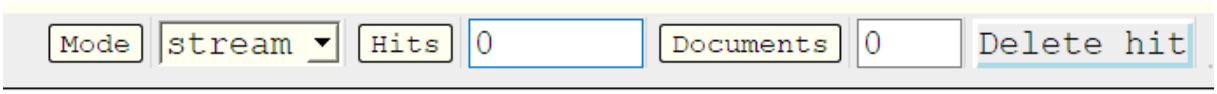


Figure 5 – ‘Concordancer’ module statusbar

The SCT offers two different modes for displaying concordances, the ‘traditional’ KWIC mode, referred to as ‘stream’ in the SCT, and a line-based one for working with data annotated in ‘Simple XML’, the form of XML I advocate where units of text are separated from surrounding annotation elements by line breaks.

In ‘stream’ mode, apart from choosing the ‘Ignore case’ option already discussed above, the left and right context can be adjusted to display a variable number of characters, and sorting of concordance lines specified for two different levels. The sorting is triggered by clicking the ‘Sort’ button once a concordance has been generated, and the numbers specify which word on either side should be used as the basis for the sorting operation, with 0 indicating the search term itself. The context options can also be adjusted in the module configuration, should the default of 30 characters turn out to inappropriate for your purposes in general.

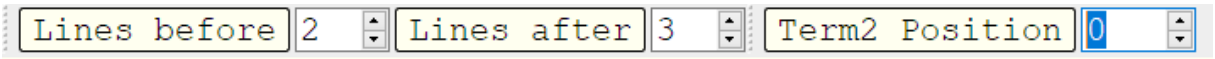


Figure 6 – ‘line’ mode options

Switching between modes is done via the ‘Mode’ dropdown list in the status bar. Activating ‘line’ mode will change a number of search options in the toolbars, as can be seen in Figure 6. First of all, it will remove the option for sorting the concordances according to left and right context, as this doesn’t make sense in line mode. However, instead, it will add an option to specify where, relative to the line containing term 1, how many lines ‘away’ term 2 should be found, either before by specifying a negative number, or following it, specifying a positive one, with 0 meaning on the same line. The prior and following context itself, this time in terms of lines, can be controlled via the spinners next to the ‘Lines before’ and ‘Lines after’ labels. In order to increase the context for an existing concordance, again the search simply has to be re-run after changing the settings.

Double-clicking on a hit will allow you to open the file containing the hit in the appropriate position in order to allow manual editing/further annotation of the data inside the SCT editor. To identify the number of a hit, use ‘Alt + click’, which will pop up a dialogue showing the line number. To select a hit for deletion, use ‘Ctrl + click’, and then click on the ‘Delete hot’ button in the status bar.

4.2 N-gram Analysis

N-gram analysis in the SCT covers both single-word(form) lists and proper n-grams. For these lists, a number of options can be specified on the toolbars below the 'Ngrams' tab, depicted in Figure 7.

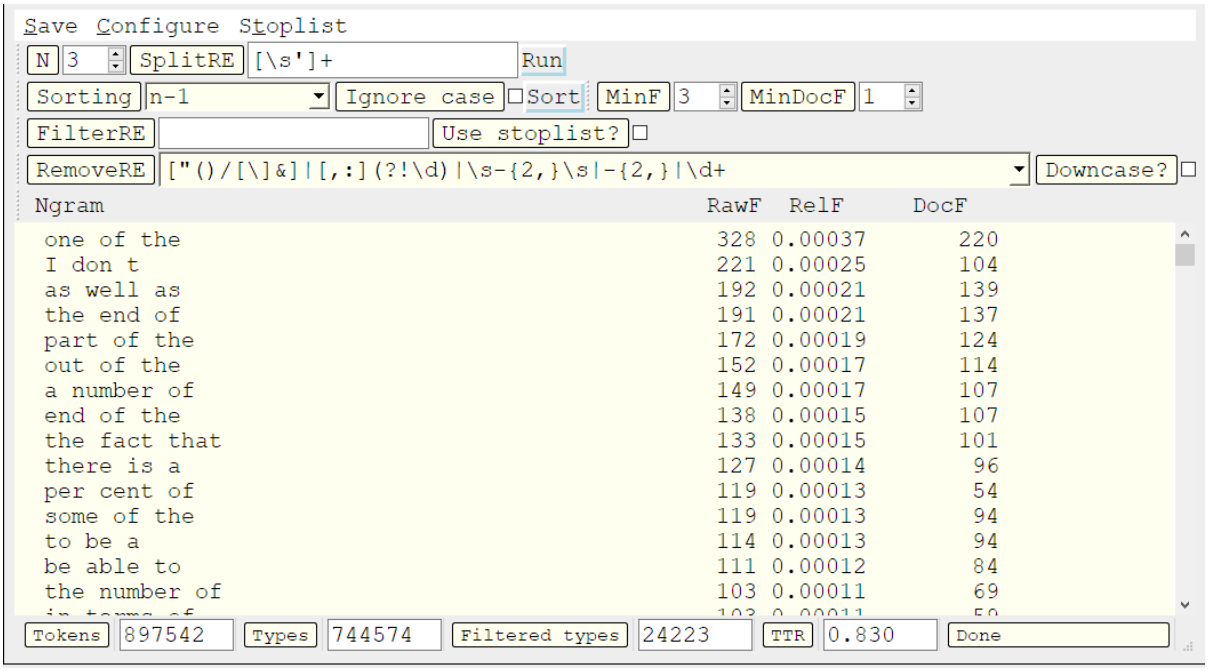


Figure 7 – the 'Ngrams' tab

The 'N' option on the top left-hand side controls the number of words in the n-gram sequence. To obtain a word(form) token count for a single file or corpus, you can simply leave the option set to '1', run the analysis, and then read off the count from the 'Tokens' entry box in the status bar, where also the number of types ('Types') – both filtered and unfiltered –, as well as the type–token ratio are displayed. The 'Filtered types' value may reflect a number of options, the use of a 'FilterRE' to only display types that fit a specific pattern, setting minimum frequency thresholds for minimum raw frequency in the corpus ('MinF'), setting minimum dispersion (document frequency; 'MinDocF'), use of a stoplist ('Use stoplist?'), where the latter is currently only implemented for pure word frequency lists and ignored for proper n-grams, or ticking the 'Downcase?' option, which would potentially conflate spellings with and without capitals, thereby reducing the overall number of types.

The usefulness of being able to easily control the splitting options ('SplitRE') between words can be seen in the second line in Figure 7. As the splitting option is set to include apostrophes, the results here treat the cliticised form *don't* as 2 words, but simply by removing the apostrophe from the regex, we can treat cliticised forms as single words.

The dropdown list for the 'RemoveRE' is a special feature of the SCT that distinguishes it from other tools. It contains a number of pre-defined options for filtering out – i.e. removing –

unwanted content in corpora on-the-fly, and can also freely be edited to improve the filtering further. Some of the options are only there to allow you to remove XML or SGML tags in case you have data that you only want to process without annotations, but essentially you can use the longer, text-oriented options to cyclically refine the analysis if you identify problematic sequences of characters in one or more texts, but without ever needing to clean up or change the original data. This makes this option very powerful, but is, of course, always to be used with caution to avoid any unwanted side effects, and requires you to be very much aware of your data. You especially need to make sure that more complex options are always listed earlier within the regex string, so that they'll be handled first. And if you observe anything strange in the output, always check to see whether your filtering regex may not have caused the problem,

The output itself consists of the n-grams, plus counts for raw frequency ('RawF'), relative frequency ('RelF'), and document frequency ('DocF'), i.e. dispersion. The width of the n-gram display in the first 'column' of the output window depends on the longest sequence identified, but which may not in fact be displayed due to the various options for filtering discussed above. Adding a 'Result filter' allows you to filter the lists by regular expression patterns, for example limiting the display to only trigrams that start with an initial capital letter or a particular word, etc.

The 'Sorting options' control the output order, and are depicted in Figure 8.

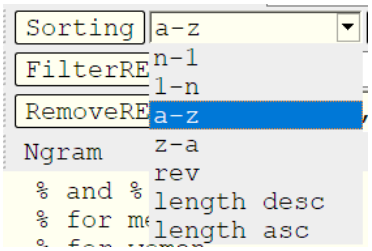


Figure 8 – n-gram sorting options

Although the options should intuitively be understandable, it's probably worth providing some more information on them.

- n-1: descending frequency order; secondary sort alphabetical;
- n: ascending frequency order; secondary sort alphabetical;
- a-z: alphabetical;
- z-a: reverse alphabetical;
- rev: reverse sorting, i.e. by endings or suffixes;
- length desc: longest first; secondary sort alphabetical;
- length asc: shortest first; secondary sort alphabetical.

Depending on which sorting option is initially selected, the list will be created using this option, so essentially once you have created a list and want to change the sort order, clicking the 'Sort' or the 'Run' button has the same effect. The default sort order can also be changed in the configuration file for the module should you find that the original default of 'n-1' doesn't work for you.

The 'Use stoplist?' option makes it possible to filter out stoplist items – more commonly referred to as 'stop words' – defined as regular expressions, one per line, in a stoplist file. This list is compiled into a full regex at runtime. The stoplist file can be created via the 'Stoplist → Edit' menu entry and edited via the 'Stoplist → Edit' entry. Once it's been created, it can be kept open to continuously refine the stoplist filtering process, provided of course the changes are saved each time before re-creating the filtered list. As stated before, the stoplist feature currently only works for unigrams, but may be extended to other n-grams later. A different stoplist file from the default one can also be specified in the 'Ngrams' module configuration file, so that it e.g. becomes possible to switch between different stoplist configurations or languages.

Depending on the size of the corpus and n, creating the n-gram lists may take a fairly long time. During that time, the processing status will be indicated in a field on the right-hand side of the status bar, and ideally you shouldn't click or scroll in the output window. Once processing and output are completed, the message 'Done' will be shown in the status bar.

Double-clicking on the n-gram sequences displayed in the n-gram output window will prime the 'Concordancer' module with a suitable regex that will also include potential interpolations, and activate the module so that it is possible to verify all occurrences directly in a concordance. However, while the interpolation will work in most cases, depending on what has been filtered out through the 'FilterRE', sometimes the primed regex may not find everything, in which case you'll need to play around and adjust it, ideally only retaining the most salient bits that are likely to be found through a concordance.

In general, playing around with the sorting options and adjustments will often reveal interesting features about your corpus, and/or help you to adjust the filtering regex to get an optimal output, so I'd suggest you spend some time doing this.

4.2.1 Editing Stoplists

Stoplists in the SCT have a very simple format; they simply consist of one entry per line. The special thing about them is that, instead of being able to only specify words, you can also use regular expressions to define whole paradigms containing all forms of a specific lemma or word class. When defining single words that may occur as parts of other words, such as e.g. *the*, which may also be part of words like *them* or *thesis*, you should always surround them by regex word boundaries (\b), so you won't accidentally delete partial words.

By default, the minimal stoplist for English called stoplist_en.txt' in the 'conf' folder will be used, unless the file name for the option 'Stoplist file' in the tool configuration is changed to another one. If you already have a number of different stoplists defined, you can switch to a new one by simply editing the configuration.

If you want to create a new stoplist instead, you can select 'Stoplist → Create …' and you'll be prompted for a new file name. Once you've provided a name, the new file will be created and opened in the editor, depicted in Figure 9.

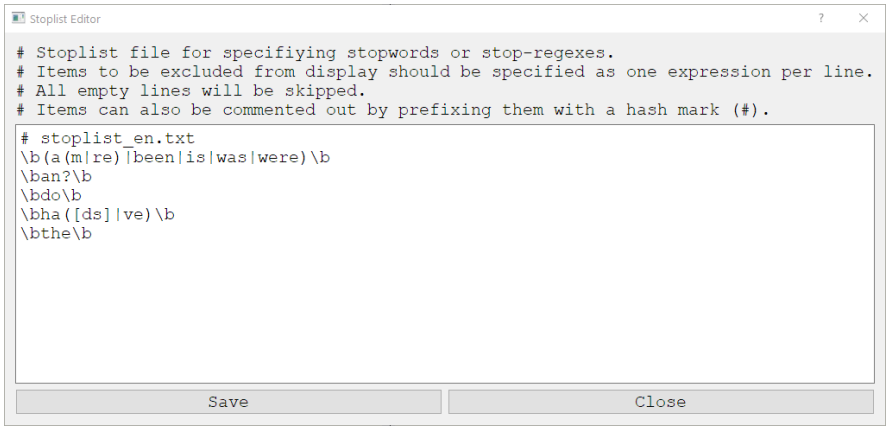


Figure 9 – the Stoplist Editor

Once you've finished editing, save and close it. To use the newly created list, change the name in the configuration dialogue (see Section 6.4) to this one, and it will be set as the currently active one for the session, and as your default list from next time you run the SCT.

To edit an existing stoplist file, choose 'Stoplist → Edit'. This will open the stoplist editor again to allow you to make your changes. For convenience, this editor is set up as a modeless dialogue, i.e. one that can stay open. If you're experimenting with a stoplist, you can hence keep it open, edit, save, and re-create your n-gram list, provided, of course, that you've activated the 'Use stoplist' option. However, I'd suggest you move the editor to a place outside the main area of the SCT where it can remain visible for that purpose.

4.3 Identifying Keywords or Keyphrases

For identifying keywords, as well as keyphrases, i.e. n-grams that occur with unusually high frequency in comparison to a reference corpus, the SCT uses a weighted relative ratio. The weighting is established by multiplying the relative frequencies of target and reference corpus by the document frequency of each item, divided by the size of the corpus. For items that don't exist in either of the n-gram lists created as the basis for comparison, a value of 0.00000000001 is used in order to avoid division by 0. The ability to create keyphrases is yet another feature that distinguishes the SCT from other tools that generally only produce keywords.

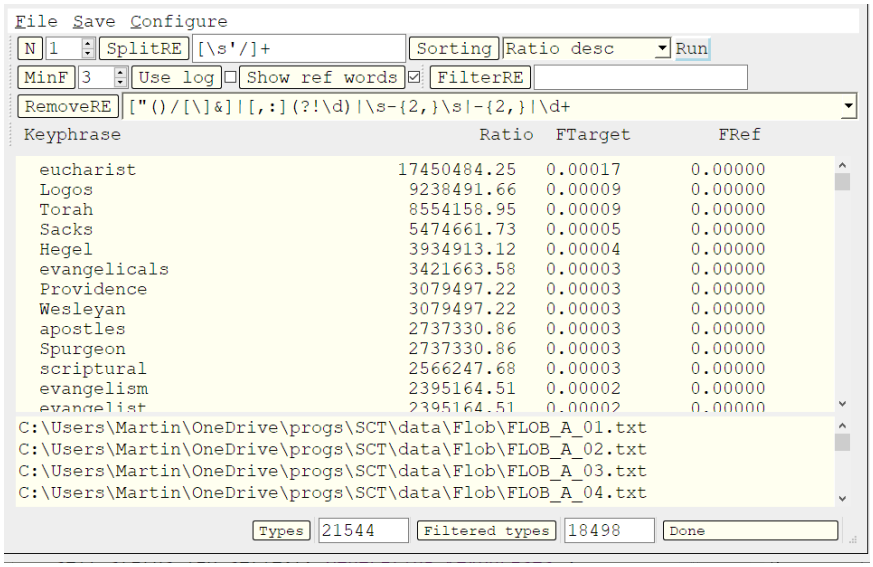


Figure 10 – the 'Keyphrases' tab

As can be seen in Figure 10, the options for controlling the keyword/-phrase generation are frequently the same as for generating n-grams, although the options for lowercasing and filtering by minimum document frequency are missing. The latter is due to the fact that the document frequency already plays a role in calculation the weighting.

There are, however, two additional settings here, one to 'Use log', i.e. scale the frequencies by applying a logarithm, and the other to 'Show ref words'. The former, as it's simply a way of displaying the frequencies in more 'compact' way, essentially has relatively little influence, other than making the transition to negative keywords/-phrases easier to see because the sign changes to a minus at that point. The latter essentially shows what doesn't occur with unusually high frequency, thereby indicating the absence of particular topical foci, as here those words or phrases that only occur in the reference corpus are included in the list of items displayed.

The sorting options, due to the different nature of the tool, also differ from those for n-grams, and include the ones depicted in Figure 11.

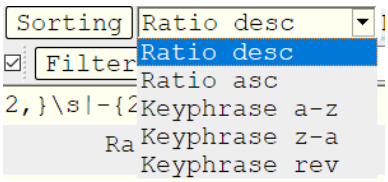


Figure 11 – 'Keyphrases' sorting options

The default setting here is by descending frequency of the ratio, but it is also possible to reverse the frequency order, as well as sort the keywords/-phrases alphabetically or reverse alphabetically, and again based on endings, which generally makes more sense for phrases than simple keywords.

The output fields consist of the keyword/-phrase, the ratio, the relative frequency in the target corpus ('FTarget'), and the relative frequency in the reference corpus ('FRef'). Unlike in the calculation process, though, if an item doesn't occur in either of the corpora, the frequency reported is 0.00000, although items occurring with a relative frequencies below this value will also be truncated to this, even if they do exist in the corpus.

4.4 Identifying Collocations

For identifying collocations, the SCT provides the two most frequently implemented measures, mutual information (MI) and the t-score. For future versions, I'm also planning to provide an option to identify collocate n-grams, such as e.g. non-hyphenated compound nouns as verb collocates, etc., which is why the module already uses the routines implemented for calculation n-grams. One positive side-effect of this is that the pre-segmentation here already improves the accuracy in comparison to other tools that may not provide an option to avoid calculating n-grams across word-boundaries. Figure 12 shows the options available for this module.

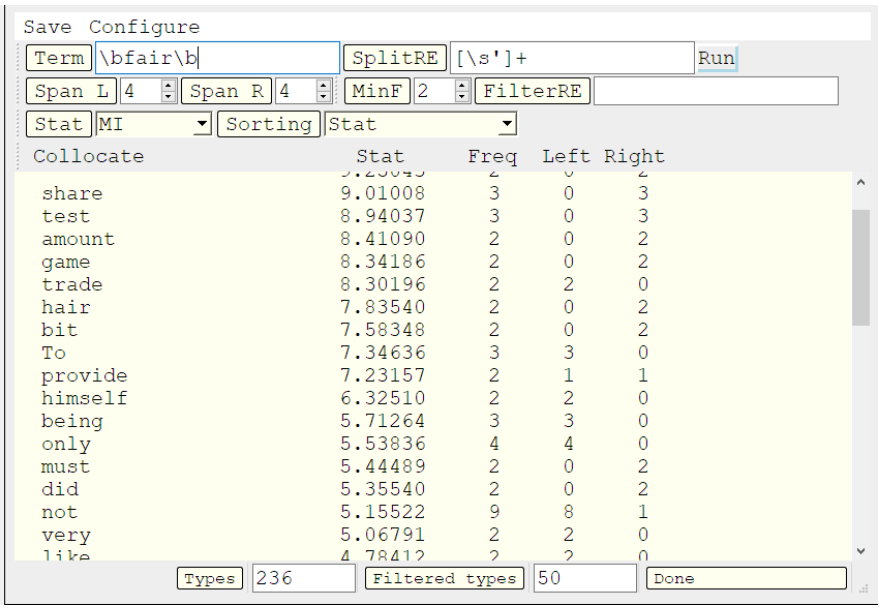


Figure 12 – the 'Collocations' tab

As with some of the other modules that display frequency counts, in addition to the pre-segmentation, it's again possible to control the 'SplitRE' and filter the output via a 'FilterRE', as well as obviously constraining the result by setting the minimum raw frequency with which a potential collocate needs to occur.

The desired statistic is chosen via the 'Stat' dropdown list, which may still be expanded in future. Left and right span in number of words is set via 'Span L' and 'Span R' respectively defaulting to 4, but can be changed in the configuration file for the module.

The output of the module shows the potential collocate, the value for the statistic ('Stat'), as well as its raw ('Freq'), left ('Left') and right ('Right') frequencies.

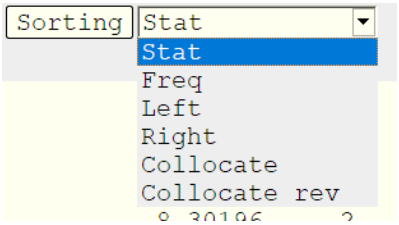


Figure 13 – collocation stat sort options

As shown in Figure 13, collocations can be sorted by the (descending) value of the statistic, the raw frequency, the left and right frequencies of occurrence, the potential collocates, or the reverse sorted collocates, where the latter may of course be useful in identifying inflectional colligations.

4.5 Counting Patterns

For counting features in the relevant module, the options need to be defined as combinations of labels + patterns, separated by a space, double colon, and another space, i.e. ' :: ', in the feature definitions window in the bottom half of the module, as shown in Figure 14, using extremely simplistic and incomplete pattern definitions only for illustrative purposes.

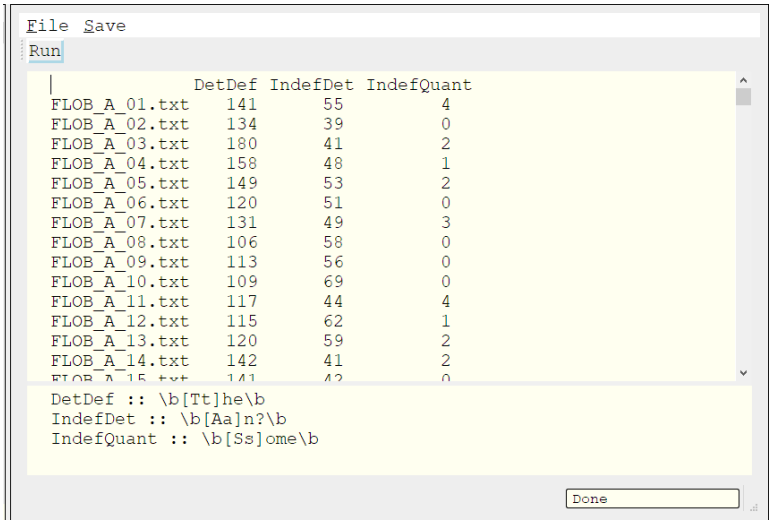


Figure 14 – the 'Pattern count' module

All patterns need to be specified as valid regular expressions. When doing a pattern count, the program will automatically check the patterns and warn if one or more of them are (syntactically) invalid, providing a hint for fixing it and indicating the line number, as well as moving the cursor to that line. It's also possible to run a manual check by choosing 'File → Check definitions' or pressing 'Ctrl + k'. Patterns can also be loaded from text files or saved to them

for repeated use via the relevant entries on the file menu. These pattern files are normally stored in the ‘conf’ folder and have the extension ‘.pat’.

The Pattern counts themselves will be displayed in the pattern count window above the feature definitions window. Here, the column labels are the labels assigned to the patterns in the feature definitions, while the row labels are the file names. Upon saving, in the text output, the feature definitions are listed after the columnar data, while they’re listed on a different worksheet when the data is output to a spreadsheet.

5 Working with the XML Editor

To edit any file from the ‘Input files’ workspace or to access a location within a file directly from inside a concordance, simply double-click the filename or hit to open it in the SCT (XML) editor. Figure 15 shows an example of an XML file from the SPAADIA Corpus loaded in the editor window.

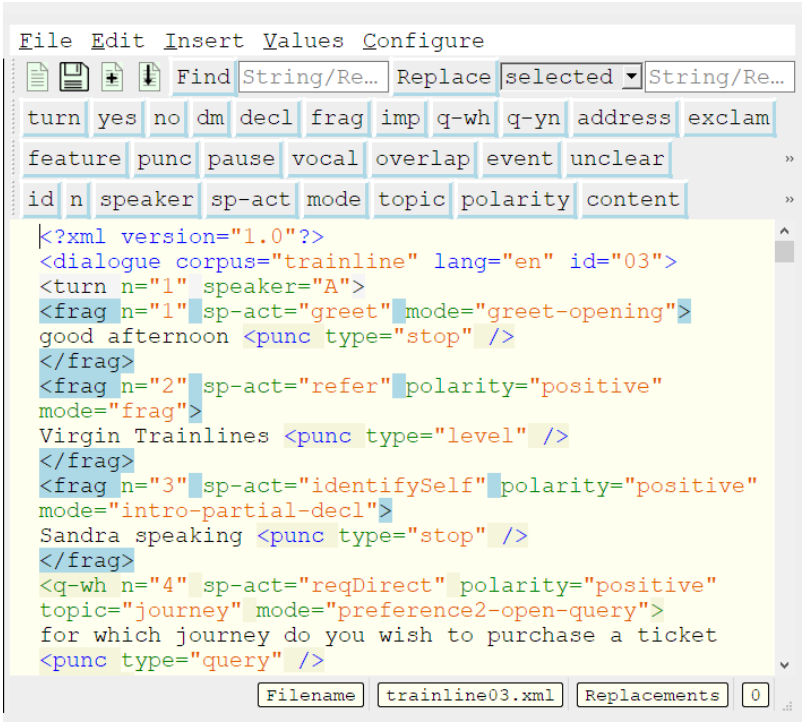


Figure 15 – the SCT XML editor displaying an XML file

If your text should happen to contain tags that are largely similar to the SPAADIA/DART tags for syntactic units, then, if you open an XML file, it will be colour-coded in a similar way. However, if you use your own tags with the tool, most of the XML will simply appear in black. The next version may provide an easy means for customising this, too, but for now, you’ll at least get empty elements and their attributes highlighted.

If you’re editing XML files, then you’ll also see three toolbars directly below the top-level toolbar, where the first one provides buttons for inserting paired opening and closing tags, the

second empty elements, and the third attributes that can be insert into either of the previous two. The configurations for these are freely editable (see next Section), so you can add your own tags, remove existing ones you're unlikely to ever use, or simply comment them out so that they won't appear on the toolbar. The SCT makes a distinction between block-level tags, which, when you're inserting them or wrapping them around selected text, will always have a line break in between them and the text, and inline tags, which will simply be inserted in the text without line breaks.

In addition to customising tags, you can also do the same for the possible values that may be used with attributes. These values are also sub-categorised, and will appear on the 'Values' menu.

As explained before, files can be opened via double-click from either the 'Input files' workspace or the 'Concordancer' module, but of course it's equally possible to open them directly from within the Editor itself, as can be seen in Figure 16, where the top entry on the file menu shows the relevant menu entry, the icon that also appears on the toolbar, as well as the keyboard shortcut.

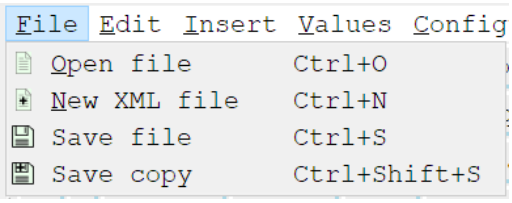


Figure 16 – the 'File' menu of the XML Editor

Choosing the entry 'New XML file', clicking the corresponding icon on the toolbar, or pressing 'Ctrl + n' will create a new skeleton XML file in the editor window. This follows the Simple XML format in that it only contains an XML declaration and a container tag that indicates the type of document (default: dialogue), and where the opening container tag can also contain basic header information in the form of attributes. Both document type and header attributes can also be customised vial the editor's configuration.

As can also be seen in Figure 16, any file that has been edited can also be saved via the menu or the standard shortcut for saving, as well as a button on the toolbar. It's also possible to save a copy of the file, for instance in order to document stages in processing. The latter command is only accessible via the menu or shortcut, though, so as to avoid cluttering the top-level toolbar, depicted in Figure 17.

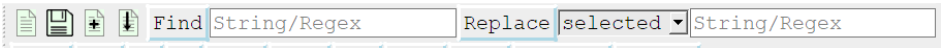


Figure 17 – the top-level toolbar of the XML Editor

As already described, the top-level toolbar contains some buttons related to basic file-handling, replication some items from the ‘File’ menu, but, in addition, also one item from the ‘Edit’ menu, the file icon with a downward-pointing arrow, as well as a number of items pertaining to search-and-replace operations. The icon with the arrow can be used to insert the contents of a plain text file, such as a dialogue transcript, into an XML skeleton, in order to edit and convert its contents to XML. The ‘Find’ button will find the text specified in the box to its right in the document. The search term should be provided as a regex, and, once found, the ‘Replace’ button can be used for regex replacement, i.e. also using patterns, either of the single instance chosen, or, by changing the dropdown list option to its right to ‘all’, throughout the whole text. This makes it possible to convert parts of the text in a highly efficient manner, depending, of course, on your level of expertise in using more sophisticated regex concepts, such as backreferencing.

The ‘Edit’ menu contains a number of convenience functions shown in Figure 18.

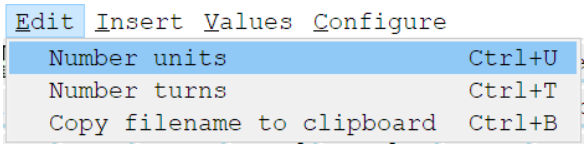


Figure 18 – the ‘Edit’ menu of the XML Editor

‘Number units’ and ‘Number turns’ allow you to automatically number the *n*-attributes of DART scheme syntactic unit categories or *turn* tags. In future versions, options for providing your own unit types may become available. The final entry on the menu allows you to copy the name of the currently open file to the clipboard, so that you can insert it into a publication if you’ve copied some sample text from a file, or to use it as a basis for creating the filename for a copy of the current document. You might also expect to find the standard options for cutting, copying, and pasting here, but those are automatically available via the standardised shortcuts (‘Ctrl/⌘ + x’, ‘Ctrl/⌘ + c’, and ‘Ctrl/⌘ + v’), as well as the context menu, so that I’ve chosen not to replicate them here. The same goes for selecting the whole document (‘Ctrl/⌘ + a’), as well as undo (‘Ctrl/⌘ + z’) and redo (‘Ctrl/⌘ + y’) operations.

The ‘Insert’ menu allows you to insert the contents of the plain text file as described earlier, as well as the current date and time if you want to add a timestamp to any file you’re creating or editing.

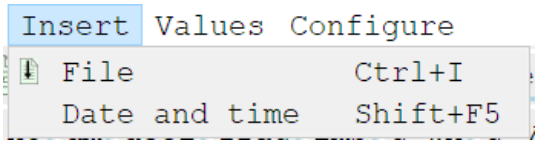


Figure 19 – the ‘Insert’ menu of the XML Editor

The ‘Values’ menu provides sub-categorised options for inserting values to complement XML attributes, and is built dynamically from the definitions in the ‘values.cnf’ file. If you want to

change the subcategories & items, though, you don't need to edit the file directly, but can do so through the Editor's 'Configure' menu.

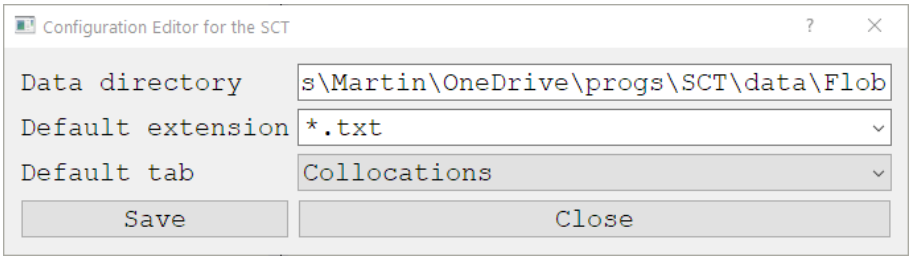
6 Editing Configurations

As previously pointed out, many of the resources for the SCT, its sub-modules, and the XML Editor can be customised. The customisation options are exposed through the individual 'Configure' menus. All customisable files live in the 'conf' folder, and you can make backup copies prior to editing the any configuration, or 'reinstate' them from the zip archive if anything should go wrong.

In general, where applicable, I would recommend copying existing entries and prefixing the original lines with a hash mark (#) so as to comment them out. In this way, you can make your own additions easier to understand and distinguish them from any pre-defined sections.

6.1 The SCT Configuration

Let's start with editing the configuration for the tool itself. This can be done by choosing the 'Configure' option on the main menu, which will then open the configuration dialogue.



The options that are currently configurable are:

- Data directory:
  This indicates the user-specified directory for data, and you can also specify it using a relative path. If you switch between different computers and the relative path cannot be found, the 'data' folder in the SCT program folder will be used as a default instead. When loading files, whichever folder is set as a default will also be shown in the file/directory chooser dialogue or files contained therein opened if you press escape during the directory-selection process.

- Default extension:
  Specifies the default extension for opening files; by default, set to 'txt'. The other option available in the default configuration is *.xml, but you can also specify your own extension option here.

- Default tab:
  Depending on which module you tend to use the most frequently, you can select the

appropriate tab from the dropdown list, so that this module will automatically be activated at startup.

In future, other options may be included.

6.2 The XML Editor Configuration

A number of options can be configured for the XML Editor through its Configuration Editor depicted in Figure 20.

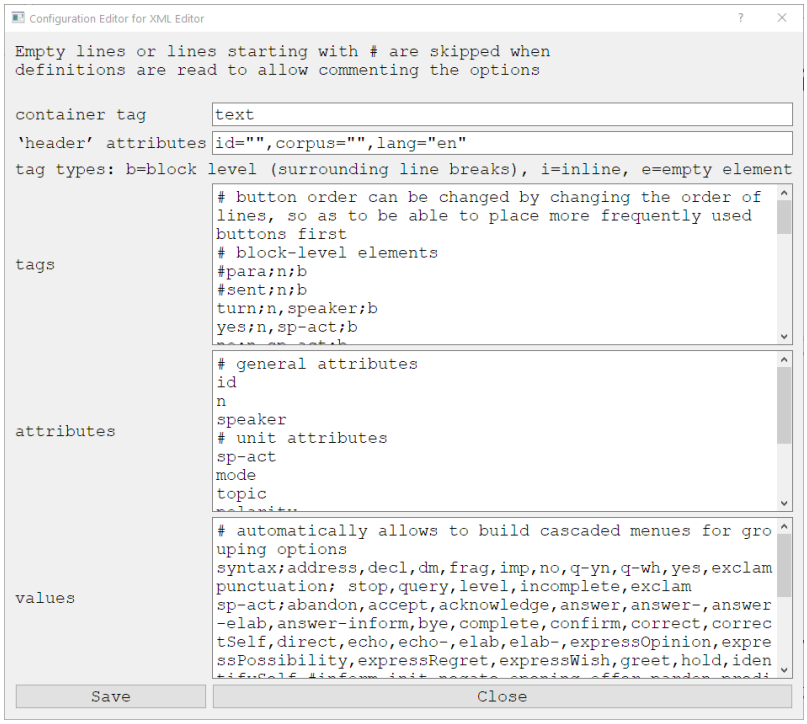


Figure 20 – the Configuration Editor for the XML Editor

The first two options here relate to the creation of new XML skeleton files, where you can specify the container tag (default *dialogue*), as well as any attributes to it that make up the minimal 'header'. For the former, you only need to specify the tag name, usually as a single word, although you could also specify a 'compound' name using an underscore between two words, as spaces aren't allowed. For the latter, you need to specify a comma-separated list, with no commas between the list items. Each item in the list should be a proper attribute–value pair, connected via the = symbol, but with the values quoted in double quotes, even if XML would also allow single ones. The reason for preferring double quotes in Simple XML, which is used in the editor, is that some attribute values in textual data – though not usually in the header – may also contain text that includes single quotes, and that hence double quotes are used for consistency. For many purposes, the default options provided in the SCT should already suffice, but for creating a whole new corpus, you might e.g. want to fill the corpus attribute, so that you won't need to type it in each time you add a file to the corpus.

The configuration dialogue also contains additional information about the resources you can edit and the forms in which you need to specify them, partly as descriptive text on the dialogue form itself, but also – for the format descriptions – in the form of tooltips.

6.2.1 Editing Tags

When editing the lists of tags that appear on the Editor's toolbars, the most important thing to remember is that the SCT distinguishes between block-level, inline, and empty elements. The block-level tags are here always separated from whatever text they contain via line breaks, whereas inline ones appear within the text and enclose parts of it. When customising any tags or adding new ones, it is therefore important to use the right abbreviation letter, 'b' for 'block-level', 'i' for 'inline', and 'e' for 'empty'. The general format for defining tags is `tag name;<comma-separated attributes);element type`, where the angle brackets indicate optionality, and each tag definition appears on a separate line. Thus, to e.g. set up a new tag for textual units, 'unit', with attributes 'n' (for running number) and 'type' (e.g. 'para', 'heading', or 'sentence'), you would write `unit;n,type;b`, as this would be a block-level element. If you want to leave the attributes out, you'd need to write `unit;;b`, still using two semi-cola, but without any content between them. Within the specifications for tags, attributes, and values, you can also add lines that start with a hash mark (#) in order to comment what the tags are for, or to temporarily exclude definitions – and hence tags – that you may want to use again at a later point in time.

6.2.2 Editing Attributes

Attributes definitions simply consist of attribute labels, one label per line. Here, you just need to remember that attribute names can only consist of single words, so that if you wanted to label attributes that would normally require two words, you'd need to combine – and probably shorten – these in some way that still leaves them easy to recognise and memorise. For instance, for speech-act attributes in the DART annotation scheme, I use `sp-act`, but you could just as easily define the attribute as e.g. either `speech_act` or `speechAct`.

6.2.3 Values

Because values in the SCT are also sub-categorised into sub-menus, they consist of pairings of category labels, followed by a semi-colon, and a comma-separated list of value labels, e.g. `punctuation;stop,query,level,incomplete,exclam` for the DART 'phono-prosodic', punctuation-like options to be used with the 'punc' tag.

6.3 The 'Concordancer' Configuration

The options that can be configures for the Concordancer module are depicted in Figure 21.

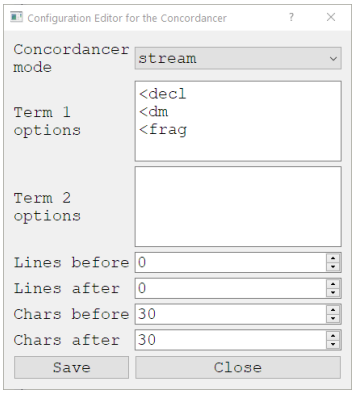


Figure 21– the Configuration Editor for the Concordancer module

The first option here is to set the concordancing mode to either a stream-based (i.e. KWIC) or line-based format by default. The next two fields allow you to specify terms for auto-completion for either of the two term entry fields, which is useful if you're looking for similar terms repeatedly across different sets of data or in different constellations.

The final four entry fields can be used to set or change defaults for the context spans, for line-based mode the number of lines before and after, and for stream-based mode the number of characters.

6.4 The 'Ngrams' Configuration

The configuration options for the 'Ngrams' module are more extensive than for the 'Concordancer' one, as can be seen in Figure 22.

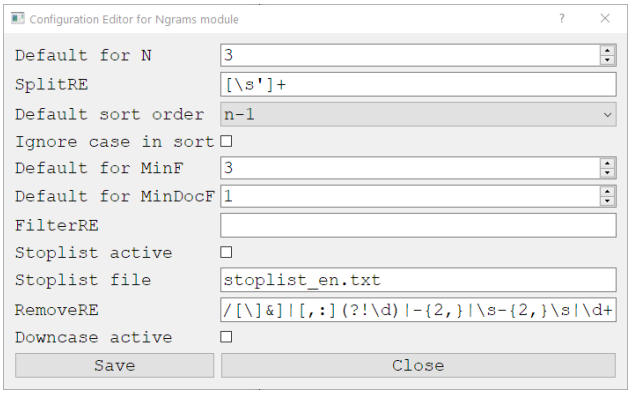


Figure 22– the Configuration Editor for the Ngrams module

Depending on which length of n-grams you investigate the most frequently, the first option you can set here is for *n* ('Default for n'. The next item ('SplitRE') controls the splitting options, e.g. whether you choose to split at whitespace only by default or whether clitics ought to be handled as words, too.

The next two options are related to sorting, changing/specifying e.g. whether you want to sort the results numerically or alphabetically. This, of course depends on whether you're more interested in frequency distributions or n-gram forms, such as specific bundles, etc. The first of these ('Default sort order') changes the default order itself, while the second ('Ignore case in sort') allows you to activate the relevant option by default, i.e. always sorting forms with capital and non-capital initials together.

In the next two fields, you can set default minimum frequencies for raw frequency of occurrence ('MinF') and minimum dispersion ('MinDocF'), while the next four options allow you to set defaults for filtering, 'FilterRE' for pure output filtering, only showing terms that match specified patterns, the stoplist items whether the stoplist specified under in 'Stoplist file' should automatically be activated for every search ('Stoplist active'), so that you don't need to manually activate it every time you want to filter out stop words, and the 'RemoveRE' option acting as an on-the-fly pre-filtering option to 'weed out' messy parts of your data.

The final option ('Downcase active') sets the default for whether all data should be lowercased all the time and should be used with caution. While this option may be extremely useful for sorting and counting sentence-initial uppercased function words together with their lowercased equivalents, in a sense lemmatising them, for languages like German that clearly distinguish between word classes by case, it may be rather counter-productive.

6.5 The 'Keyphrases' Configuration

The options for default settings for *n*, the splitting and the removal regex, as well as the minimum raw frequency, shown as items 2,3, 6, and 7 from the top in Figure 23, are essentially the same as for the 'Ngrams' module, so we don't need to discuss them further here. Nevertheless, despite being identical to those options, they aren't linked to those in the other module, but need to set independently.

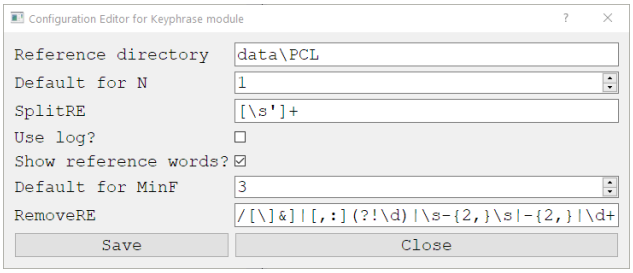


Figure 23– the Configuration Editor for the Keyphrases module

The first item in the dialogue, 'Reference directory', allows you to specify a relative path to where your reference corpus is stored. 'Use log' activates the option for using the logarithm as a scaling factor by default, while 'Show ref words' does the same for including words that don't occur in the target corpus, i.e. 'absolute' negative keywords/-phrases.

6.6 The 'Collocations' Configuration

As can be seen in Figure 24, the 'Collocation' module shares the options for splitting, minimum raw frequency, and filtering with the 'Ngrams' an 'Keyphrases' modules.

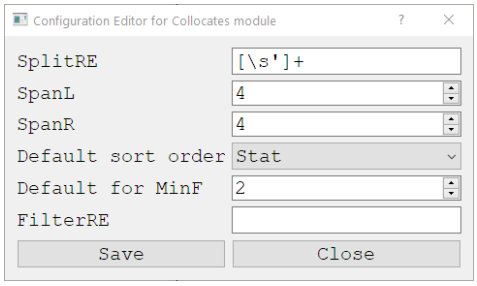


Figure 24– the Configuration Editor for the Collocations module

In addition, it's also possible to set default values for left and right span, and for the 'Default sort order', as in the 'Collocations' module.

The 'Pattern count' module does not have a configuration option, as essentially the different options currently available only exist in the form of pattern files. In future versions, though, options for norming factors or including dispersion may be added.

7 List of Keyboard Shortcuts

There are number of shortcuts defined for different purposes and modules, as well as handling input files in the main SCT window.

7.1 SCT Main Window

F3	Open files for processing
F2	Open folder for processing
Alt + F3	Add files for processing
Alt + F2	Add folder for processing
Alt + F5	Load virtual corpus
Del	Remove selected files form input workspace
F1	Help (opens manual)
Ctrl + q	Exit SCT

7.2 General Shortcuts

Ctrl + r	Run analysis in current module
Ctrl + a	select all
Ctrl + c	copy
Ctrl + x	cut
Ctrl + v	paste

7.3 Keyphrase shortcuts

Ctrl + F3	Load reference files
Ctrl + F2	Load reference folder
Ctrl + F5	Load virtual reference corpus
Ctrl + Del	Remove files from reference file list

7.4 Pattern Count Shortcuts

Alt + F3	Load pattern file
Alt + F2	Save pattern file

7.5 Editor Shortcuts

Ctrl + o	Open a file
Ctrl + n	Create new XML skeleton file
Ctrl + s	Save current file
Ctrl + Shift + s	Save copy of current file
Ctrl + u	Number units

Ctrl + t	Number turns
Ctrl + b	Copy filename to clipboard
Ctrl + i	Insert file contents
Shift+F5	Insert date and time