ParaConc: Concordance Software for Multilingual Parallel Corpora

Michael Barlow

Rice University
Dept. of Linguistics
Houston, TX 77005
barlow@rice.edu

Abstract

Parallel concordance software provides a general purpose tool that permits a wide range of investigations of translated texts, from the analysis of bilingual terminology and phraseology to the study of alternative translations of a single text. This paper outlines the main features of a Windows concordancer, ParaConc, focussing on alignment of parallel (translated) texts, general search procedures, identification of translation equivalents, and the furnishing of basic frequency information. ParaConc accepts up to four parallel texts, which might be four different languages or an original text plus three different translations. A semi-automatic alignment utility is included in the program to prepare texts that are not already pre-aligned. Simple text searches for words or phrases can be performed and the resulting concordance lines can be sorted according to the alphabetical order of the words surrounding the searchword. More complex searches are also possible, including context searches, searches based on regular expressions, and word/part-of-speech searches (assuming that the corpus is tagged for POS). Corpus frequency and collocate frequency information can be obtained. The program includes features for highlighting potential translations, including an automatic component "Hot words," which uses frequency information to provide information about possible translations of the searchword.

Keywords: alignment, parallel texts, concordance software

ParaConc is a tool designed for linguists and other researchers who wish to work with translated texts in order to carry out contrastive language studies or to investigate the translation process itself.

1. Alignment

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

The alignment utility in *ParaConc* is semi-automatic. When files are loaded, the user enters information about the format of the files either through reference to SGML tags or via specifications of patterns. The user specifies the form of headings and the form of paragraphs. *ParaConc* uses the information to align the documents at this level and the user can make adjustments by merging/splitting units, as appropriate. Sentence level alignment, if it is not indicated by SGML tags, is performed using the Gale-Church algorithm (Gale and Church, 1993). The alignment information is saved to a file as part of the workspace, as described in Section 6.

No use is made of bilingual dictionaries or of any kind of language-particular information, but the user can enter pairs of anchors, such as cognates, numerals and dates, which the program will track. These anchors are not used in the alignment process itself, but aligned units which do not contain the appropriate corresponding anchors are highlighted for manual checking by the user.

If the parallel texts are pre-aligned, then it is simply necessary to indicate the manner in which the alignment is marked.

2. Loading the Parallel Corpus

When the LOAD CORPUS FILE(S) command is given, a dialogue box appears, enabling particular parallel files to be loaded, as shown in Figure 1.

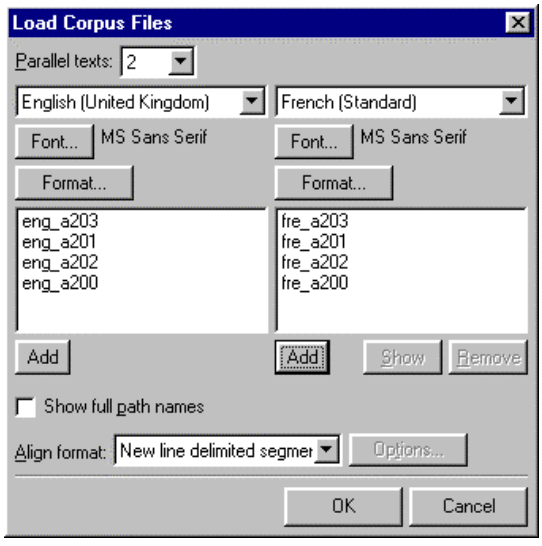

Figure 1. Loading Corpus Files

The heading PARALLEL TEXTS at the top of the dialogue box is followed by a number in the range 2-4 (i.e, two to four different languages). The FORMAT buttons allow the user to describe the form of headings, paragraphs, and sentences, as discussed above. Filenames can be reordered by dragging them to the appropriate position.

3. Searching and Analysing Parallel Texts

The program processes the files as they are loaded, counting words, recording the position of alignment indicators, and processing other format information.

Once a corpus is loaded, some new menu items related to the analysis and display of the text appear on the menu bar. These are FILE, SEARCH, FREQUENCY, and INFO. In addition we can obtain information in the lower left corner of the window relating to the number of the files loaded and in the lower right corner a word count for the two corpora is provided.

Selecting SEARCH from the SEARCH menu initiates the search process and the program starts to work though the loaded files looking for the search string. The search can be based on any of the languages represented: either English or French in this example. (The basic search is fairly simple: a word or a phrase can be entered, including simple wildcard characters if necessary. The symbols acting as wildcards are user-defined, but the default symbols are ? for one character; % for zero or one characters; and * for zero or more characters. The symbol @ covers a specified range of words. Information on the span covered by @ and other information such as a list of characters that act as word delimiters is available in SEARCH OPTIONS.)

Below the results of a search for *head* are illustrated. The instances of *head* are displayed in a KWIC format in the upper window. Clicking on one particular example of *head* in English highlights both the English and French lines. (Double-clicking on a particular line evokes a context window, which provides an enlarged context for the particular instance of the searchword.)

The lower part of the window contains the French sentences (or text segments) that are aligned with the hits displayed in the top window. This display of equivalent units in the two languages is, of course, a consequence of the alignment process. Thus if the first instance of *head* occurred in segment 342 of the English text, then the program simply throws segment 342 of the French text into the lower window, and this process is repeated for all instances of *head*.

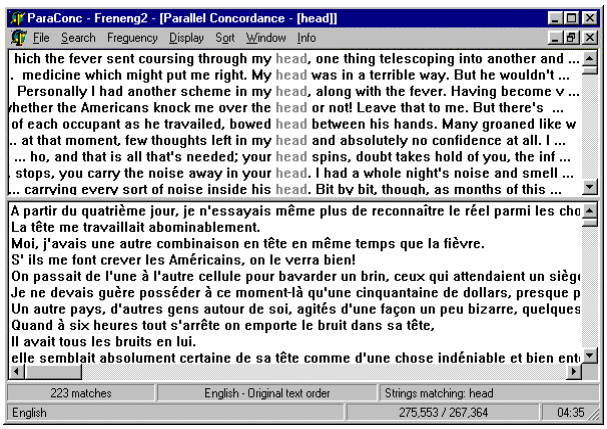

Figure 2: The Results of a Simple Search

Let's follow this example further. Once the search is ended, we can bring to bear the usual advantages of concordance software to reveal patterns in the results data. One may be interested, for example, in different uses (and translations) involving *head: big head, company head, shower head*, etc. One way to find out which English words are associated with *head* is to sort the concordance lines so that they are in alphabetical order of the word preceding the search term. The advantage of performing this 'left sort' is that the modifiers (adjectives) of *head* that are the same will occur together. One easy way to achieve this ordering is to select 1ST LEFT, 1ST RIGHT, from the SORT menu.

It can perhaps be seen from Figure 2. that while all the instances of *head* are clearly displayed, it is difficult to look through the equivalent French segments in order to locate possible French translations of *head* within each segment. To alleviate this, we can highlight suggested translations for English *head* by positioning the cursor in the lower French results window and clicking on the right mouse button. A menu pops up and we can select SEARCH QUERY which gives access to the usual search commands and hence allows us to enter a possible translation of *head* such as *tête*. The program then simply highlights all instances of *tête* in the French results window.

We can now change the context for the French results so that the results in the lower window are transformed into a KWIC layout (at least for those segments containing *tête.)* First, we make sure that the lower window is active. Next we choose CONTEXT TYPE from the DISPLAY menu and select WORDS. Finally, we rearrange the lines to bring those segments containing *tête* together at the top of the French results window. To achieve this, we choose SORT and sort the lines by searchword, and 1st left. The sorting procedure will then rearrange the results in lower window. (The SORT and DISPLAY commands are applied to whichever window is active.) The two text windows then appear as shown in Figure 3. Naturally, only those words in the French text that have been selected and highlighted can be displayed in this way. By sorting on the searchword, all the KWIC lines are grouped together at the top of the text window; the residue can be found by scrolling through towards the bottom of the window. This is a revealing display, but we have to be careful and not be misled by this dual KWIC display. There is no guarantee that for any particular line, the instance of *tête* is in fact

the translation of *head*. It could simply be accidental that *tête* is found in the French sentence corresponding to the English sentence containing *head.*

The idea behind dual KWIC display is to let the user move from English to French and back again, sorting and resorting the concordance lines, and inspecting the results to get a sense of the connections between the two languages at whatever level of granularity is relevant for a particular analysis.

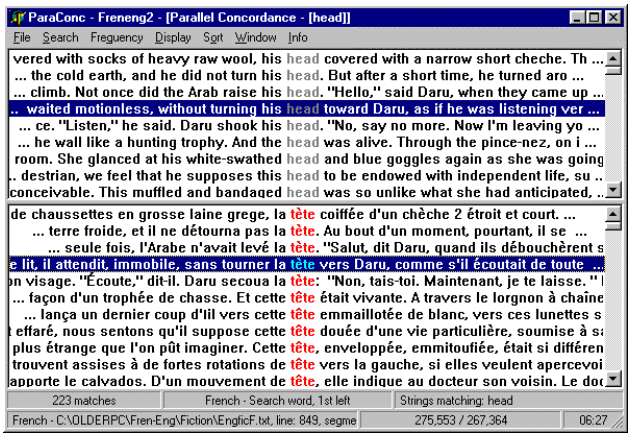

Figure 3: Parallel KWIC displays

4. Hot Words

In the previous section, we described the use of SEARCH QUERY to locate possible translations in the second window. In this section we will look at a utility in which possible translations and other associated words (collocates) are suggested by the program itself. We will refer to these words as *hot words*. First we position the cursor in the lower (French) half of the results window and click using the right mouse button. If we used SEARCH QUERY earlier, we need to select CLEAR SEARCH QUERY and then choose HOT WORDS, which invokes a procedure which calculates the frequency of all the words in the French results window and then brings up a dialogue box containing the ranked list of hot words. The ranked list of candidates for hot words based on *head* are displayed as shown in Figure 4.

To select words as hot words, the program looks at the frequency of each word in the results window and ranks the words according to the extent to which the observed frequency deviates from the expected frequency, based on the original corpus. The words at the top of the list might include translations of the searchword, translations of the collocates of the searchword, and collocations of translation of the searchword.

In addition to the basic display of hotwords, a paradigm option (if selected) promotes to a higher ranking those words whose form resembles other words in the ranked list. This is a simple attempt to deal with morphological variation without resorting to language-particular resources.

Some or all the hot words can be selected. Clicking on OK will highlight the selected words in the results window, and again the words can be sorted in various ways.

Figure 4: Hot Word List

5. Frequency information

ParaConc furnishes a variety of frequency statistics, but the two main kinds are corpus frequency and collocate frequency. The command CORPUS FREQUENCY DATA in the FREQUENCY menu creates a word list for the whole corpus (or parallel corpora), according to the settings in FREQUENCY OPTIONS. The results can be displayed in alphabetical or frequency order and the usual options (such as stop lists) are available.

Choosing COLLOCATE FREQUENCY DATA from the FREQUENCY menu displays the collocates of the search term ranked in terms of frequency. In *ParaConc,* the collocate frequency calculations are tied to a particular search word and so the frequency menu only appears once a search has been performed. The collocation data produced by the COLLOCATE FREQUENCY DATA command is organised in four columns, spanning the word positions 2nd left to 2nd right. The columns show the collocates in descending order of raw frequency.

One disadvantage of the simple collocate frequency table is that it is not possible to gauge the frequency of collocations consisting of three or more words. To calculate the frequency of three word collocations, it is necessary to choose ADVANCED COLLOCATION from the FREQUENCY menu and select one or more languages. The top part of the dialogue box associated with ADVANCED COLLOCATION allows the user to choose from up to three word positions, for example, SEARCHWORD 1ST RIGHT, 2ND RIGHT. The program counts and displays the three-word collocations based on the selected pattern.

6. Workspace

The loading and processing of a parallel corpus in particular can take some time since the program has to process alignment and annotation data before searching and analysis can begin. Since the same sets of corpus files are often loaded each time *ParaConc* is started, it makes sense to freeze the current state of the program, at will, and return to that state at any time, rather than starting *ParaConc* and reloading the parallel corpora afresh. This is the idea behind a workspace. A workspace is saved as a special (potentially large) ParaConc Workspace file (.pws), which can then be opened at any time to restore

ParaConc to its previous state, with the corpus loaded ready for searching. Searches and frequency data are, however, not included in the saved workspace. (Only the search histories are saved.)

A workspace can be saved at any time by selecting the command SAVE WORKSPACE or SAVE WORKSPACE AS from the FILE menu. The usual dialogue box appears and the name and location of the workspace file can be specified in the normal way. Once a filename for the saved workspace has been entered, the user is asked to choose some different workspace options. The line/page and the tracked tag info can be saved as part of the workspace. (The saved workspace consists of a saved file and an associated folder of the same name.)

7. Advanced Search

The simple searches described in Section 3 will suffice for many purposes and are especially useful for exploratory searches. The basic TEXT SEARCH is also very useful when used in conjunction with a sort-and-delete strategy. Particular sort configurations can be chosen to cluster unwanted examples (words preceded by *a* and *the* perhaps), which can then be selected and deleted. For more complex searches, however, we need to use the ADVANCED SEARCH command. This command brings up a more intricate dialogue box (displayed in Figure 5), which at the top contains the text box in which the search query is entered.

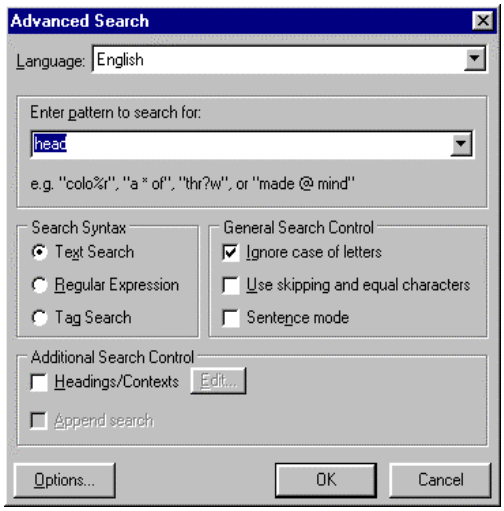

Figure 5: Advanced Search

The most important part of the ADVANCED SEARCH dialogue box is labelled SEARCH SYNTAX. The three radio buttons allow users to specify the kind of search they wish to perform. The first, TEXT SEARCH refers to the basic searches described in the section above.

The REGULAR EXPRESSION search allows for search queries containing boolean operators (AND, OR and NOT). For example, a regular expression to capture the *speak* lemma might be given as **sp[eo]a?k[se]?n?**. This expression will match the string *sp* followed by *e* or *o*, an optional *a*, a *k*., an optional *s* or *e*, followed by an optional *n*. (Word boundaries or spaces would also have to be specified in order to eliminate words such as *bespoke*.) The software also supports the expanded set of regex metacharacters: \d, \w, \s, \S, etc.

The third option in the advanced search dialogue box is TAG SEARCH, which allows the user to specify a search query consisting of a combination of words and part-of-speech tags, with the special symbol **&** being used to separate words from tags in the search query. This search syntax is used whatever particular tag symbols are used in the corpus. (Thus it is necessary to enter the form of the tags in TAG SETTINGS before a tag search can be performed.) To give an example: the search string **that&DD** finds instances of *that* tagged as a demonstrative pronoun, which may appear in the corpus as *that<w DD>*. Similarly, a tag search for **&JJ of&** will find all instances of adjectives followed by the word *off*. (The dialogue box in Figure 5 contains a variety of other options controlling the search function, which will not be discussed in this paper.)

Finally, one kind of search tailored for use with parallel texts is a parallel search, which is one of the options within the SEARCH menu. This type of search, shown in Figure 6, allows a search to be constrained based on the occurrence of particular strings in the different parallel texts.

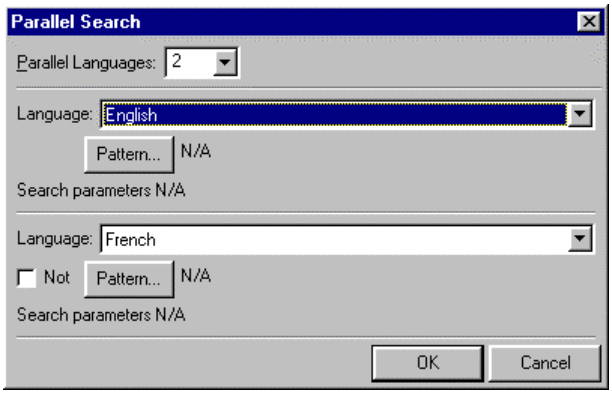

Figure 6: Parallel Search

Clicking on the Pattern box under Language: English brings up the normal advanced search dialogue box and a search query can be entered. In this case, the search term **head** has been entered. Moving to Language: French and again clicking on Pattern, it is possible to enter another search string such as **tête**. Clicking OK initiates the search routine and the software locates examples in which head occurs in the English text segment and tête is also found in the corresponding French segment. If the NOT box (under Language: French) is selected, then the search routine will display head only if tête does not occur in the equivalent French segment.

8. Summary

This paper has provided a brief overview of a Windows parallel concordance program which can be used by a variety of researchers working on the analysis of multilingual texts for translation or linguistic purposes. This article has focussed on the overall design and operation of the software and no linguistic analyses have been presented here, but the potential for cross-linguistic analyses and for the investigation of the translation process is, we hope, reasonably clear.

The main factor impinging on the usefulness of the software is probably the availability of aligned parallel corpora and of parallel corpora in general.

9. References

Gale, W. A. & Church, K. W. (1993). A program for aligning sentences in bilingual corpora. In *Computational Linguistics*, 19, 75—102.