

This document contains

- 1) **my review of the Corpus Presenter software suite by Raymond Hickey as published on the LinguistList (posting 15.681, dated 24 February 2004),**
- 2) **Raymond Hickey's 'reply' to me personally which I received on 18 March 2004,**
- 3) **Raymond Hickey's reply to the LinguistList (posting 15.981, dated 23 March 2004),**
- 4) **my reply to the LinguistList (posting 15.1087, dated 24 March 2004).**

My review of Corpus Presenter as published on the LinguistList (posting 15.681, dated 24 February 2004)

AUTHOR: Hickey, Raymond
TITLE: Corpus Presenter
SUBTITLE: Software for language analysis [...]
PUBLISHER: John Benjamins
YEAR: 2003

Reviewer: Stefan Th. Gries, University of Southern Denmark

(References to parts of the Corpus Presenter suite are printed in blue for easier orientation.)

Description/summary of the bundle

1) The book

Part I (p. 1-27) is a brief introduction into corpus linguistics. This part provides an overview over some corpus-linguistic terminology (types of corpora, tagging, corpus headers, etc.) and a brief section entitled 'examining corpora', which introduces a few basic notions such as *concordance*, *tagging* and *lexical cluster analysis* (each with about two paragraphs). P. 14 to 27 discuss very briefly a few sample analyses of corpus data (mainly Irish English data); these involve frequency data for the presence or absence of particular linguistic forms, relative frequencies of for collocations and relative frequency data in an investigation of an author's style.

Part II of the book (p. 28-183) consists of descriptions of the modules or programs of the [Corpus Presenter](#) suite (henceforth [CP](#)). Much of this part corresponds to the help files of the software bundle.

Part III of the book consists of several appendices. Appendix 1 and 2 of the book (p. 184-9) provide information about the installation; while Appendix 3 (p. 190-200) lists a set of common commands, i.e. commands which are found in several parts of [CP](#). Appendix 4 (p. 201-4) describes the file interface of [CP](#), most of this will be familiar to most users of Windows. Appendix 5 (p. 205-7) gives some troubleshooting information and Appendix 6 (p. 208-9) introduces three additional dataset files describing three corpora from the ICAME CD-ROM. Finally, this part contains a glossary of short definitions of corpus-linguistic and a few statistical terms.

Part IV of the book (p. 237-76) is a description of A Corpus of Irish English, which is followed by a general bibliography, a glossary for this corpus and a combined subject/name index.

2) The software

[CP](#) offers multiple functions for compiling, annotating and processing corpora.

- [CP](#) can search for strings in texts in order to output them as a concordance:
 - o [CP](#) allows for detailed specification of strings and corpus text settings including those for case-sensitive searches, sentence and word delimiters, punctuation signs;
 - o in addition, one can look for single expressions, lists of expressions and larger expressions (by specifying the left and the right part of a complex expression as well as their maximal distance);
 - o an especially useful option is the possibility to include special characters and symbols from every installed font into the search pattern;
 - o one can specify stop words and output options such as full sentence or x words to left/right of the search word;
 - o one can specify Cocoa settings to include only files with particular attributes into the search;
- [CP](#) can generate collocation frequencies for words in a span of max. eight words around a search/node word;
- [CP](#) can generate some text statistics (word counts) as well as (regular or reverse) word lists of individual words and/or clusters of up to 8 words;
- [CP](#) can perform search and replace operations on files to alter texts (e.g. tagging and normalizing files for lemmatization);

- **CP** can collate files and compile corpora by collecting and manipulating files of various sorts, organize the files and their contents hierarchically and add header information (e.g. Cocoa) for future searches;
- **CP** can convert files to different file types, manipulate their attributes (e.g. date stamps, extensions) and do lots of file handling operations (cut, copy, past, duplicate, merge, etc.);
- **CP** offers a few useful functions that go beyond some limitations of the Microsoft Windows OS such as a cumulative clipboard or an undo buffer for deleted text.

The program comes with many different help files, FAQ files and a brief tutorial with 44 slides and written/read-aloud instructions (.WAV format).

Critical evaluation

The **CP suite** is a set of programs that offers a vast range of possibilities for working with corpus data. It was mainly tested on a notebook computer (Pentium III 1000 with a 20GB hard disk and 256MB RAM running an English Windows XP Professional; some additional tests were performed on a desktop computer (Athlon XP 1800+ with a 40GB hard disk and 640MB RAM) running a German Windows 2000 (both systems are completely updated in terms of Microsoft Service Packs etc.). The program was tested by myself alone, but in order to make the evaluation slightly more objective, I also asked a colleague for her opinion on some issues. In order to discuss some of **CP**'s properties, I will make reference to a few concordancing programs, namely WordSmith Tools 3 (WST), MonoConc Pro 2.2 (MCP) and WinConcord 2.0 (WC).

1) Speed and power of **CP**

The author (henceforth RH) stressed that “a special fast retrieval mode has been incorporated into *Corpus Presenter* to minimize the time one has to wait for returns to be made during searches” (<CP_GUIDE.RTF>). However, at least my own experiments do not support this assessment (with one exception mentioned below all time-taking experiments were performed after immediately after a system reboot).

- Concordancing example 1: On the above-mentioned notebook computer, searching the Brown Corpus for the word form *best* using the most powerful ‘text retrieval’ level lasted an astonishing 35.7 seconds (and some 35.94 seconds for *after; CP Main Programme*'s own output) even though all settings concerning Unicode were optimized for the processing of plain ASCII files. By contrast, MCP took about 2 seconds to load the file and 4 seconds to produce the concordance.
- Concordancing example 2: On the above-mentioned notebook computer, searching 674 files from the BNC part A (without tags) for the word form *best* once took 1030.48 seconds (with several applications open but unused in the background) and 334.21 with no other applications running (*CP Main Programme*'s own output) even though all settings concerning Unicode were again optimized for the processing of plain ASCII files. By contrast, WST took about 57 seconds to produce the desired concordance ...
- Concordancing example 3: On the above-mentioned notebook computer, simply finding out how that my British National Corpus (BNC) directory contains 4,054 .TXT files required 47 seconds (**CP Flash**) – both MCP and WST need about a second or less.
- Word list examples: Making a simple word list of the Brown Corpus (without the reversed list) required 469.46 seconds with **CP Main Programme** (and I canceled **CP Flash** after 30 minutes!) but only 11 seconds with MCP. In this connection, it is worth pointing out that the program has an upper limit of 32,000 words for word lists. RH does claim that this is “virtually ample for every corpus” (p. 59), but the word list for the one-million Brown Corpus already had about 10,000 entries so it is easy to find corpora whose word lists will exceed this limit: the word list of the one-million FROWN Corpus mentioned in RH's book itself has 50,000+ lines (processing time with **CP Main Programme**: 781.41 seconds; processing time with WST: 7 seconds), and the word list of the 100-million-words BNC available from A. Kilgarriff's website (at <http://www.itri.bton.ac.uk/~Adam.Kilgarriff/>) has 938,000+ lines ...
- Merging files: With **CP File Manager**, merging 15 text files (with an overall size of 6,758KB) required 4:08 minutes.

All in all, thus, CP is rather slow, especially when compared to other contemporary programs.

2) Ease and convenience of use of **CP**

My own impression of the usability of **CP** is rather negative, especially when compared to the other three corpus programs that I work regularly with for teaching and research mentioned above. While I do admit that the range of functions is large and that I may not be able to do justice to all features the suite has to offer, I am not very happy with a variety of features. My main concerns are as follows:

2a) The modules of **CP**

The **CP suite** comes with a wide variety of different modules and is intended to bring together modules to carry out a huge number of different tasks into a single suite, which basically sounds like a good idea. If **CP** and other similar programs such as WST, MCP and WC were located on a ‘modularity scale’, then MCP and WC would have the simplest structure such that all commands can be accessed from a single window with one menu bar; by contrast, WST is a suite with three modules doing different corpus jobs plus four modules for file handling etc.;

and CP is a suite of 27 modules in five groups. Compared to the other programs, CP's structure, thus, appears relatively complex, an impression that was involuntarily confirmed by some unguided experimentation: while I could use many capabilities of WST, MCP & WC without having looked at any documentation, now after several years experience doing corpus-linguistic research with different corpus programs and Perl scripts I was unable to do a simple corpus search with CP without having looked at the documentation provided with the bundle.

A related point of criticism is that many of the modules serve purposes for which even the most modestly equipped (corpus) linguist probably already has resources available that can perform (most of) what is needed. For example, for the potential buyer it is worth pointing out that more than half of the 27 modules provided on the CD-ROM are applications that strongly resemble Microsoft Windows, Microsoft Office or OpenOffice products:

- **CP Slide**, a program which “will group any set of files into a list which one can page through like slides on a projector (from one to the next, without interruption, on a clear screen)” (<CP_GUIDE.RTF>), a set of functions many of which Microsoft PowerPoint or OpenOffice Impress can perform;
- **CP Browser**, a web browser, which provides functions most of which Netscape Navigator, Internet Explorer etc. already provide;
- **CP File Manager**, **CP FileManager Lite** and **CP Quick Backup**, a program “similar to the file manager but slightly different in its organization” (<CP_GUIDE.RTF>), all allow you to perform various file manipulation and storing operations; most, though not all, of these can of course be performed by the regular Windows Explorer or other (freeware) programs; the same holds for the module **CP Find Text**;
- **CP Diary**: a program that is intended to remind you of important dates and allows you to have a yet-to-do list, i.e. it offers part of the functionality of Microsoft Outlook etc.;
- **CP Jotter** “provides a small and very quick version of the fuller text editors of [CP]” and, thus, does the same thing as <Notepad.EXE> on every Windows system (or TextPad or UltraEdit or ...); in addition, CP also has a command ‘view returns storage’ which provides yet another window where you can enter data for later storage just like <Notepad.EXE>; and there's also **CP Text Editor** and **CP Text Tool**, which are text editing utilities ...;
- as if the previously mentioned text editing modules were not enough, there is also **CP Word Processor**, which does the same things as Microsoft Word or OpenOffice Writer;
- **CP Easy Chart** “will generate a pie, bar or line chart from any series of input numbers” (<CP_GUIDE.RTF>), which is of course what one normally uses Microsoft Excel / OpenOffice Calc for;
- **CP Database Editor** and the separate **Database Manager** serve the purpose of processing database file (e.g. .DBF), a function for which again most people use Microsoft Excel or OpenOffice Calc;
- **CP Internet Editor** allows you to edit your homepage(s) and, therefore, does the same thing as Microsoft Frontpage or any other freely available HTML editor;
- **CP Control Centre**: a small module that gives you access to a variety of system setting options most of which are already accessible from the Windows Control Panel ...;

(My further discussion of CP below will not address all of these modules at the same level of detail since many of the modules and/or their functions are not relevant in a more narrowly defined corpus-linguistic sense; in addition, many options of these ‘non-corpus-linguistic’ modules I have tested were not superior in functionality to their Windows/Office counterparts anyway. For example, the possibilities to generate charts with **CP Easy Chart**'s chart options appear to be much less sophisticated than, say, Microsoft Excel's options especially since the latter can generate graphs directly from automatically updated pivot tables without the whole lot of manual effort required by **CP Easy Chart**.) Also (a minor point though), many of the modules themselves contain commands which are nice little gimmicks but which add little to the linguistic functionality/utility of this corpus processing suite. Examples for these include the possibility to access a calculator or the time/date from several modules, the possibilities of adjusting color and/or wallpaper or font settings for many modules, the possibility to access **CP Jotter** from some modules' menus, the option to view the RH's CV etc. ... these options of course probably don't really hurt, but they do of course inflate the number of commands beyond what is necessary and easily/intuitively handable ...

2b) CP and Windows

Another important usability issue is concerned with the way CP integrates into, or makes use of the capabilities of, the Microsoft Windows operating system. While RH emphasizes that the program is designed “afresh, utilizing to a maximum the possibilities of the newer operating system” (p. 28), I would not quite agree to this assessment. Consider, for an admittedly painfully detailed example, the installation of CP:

Upon double-clicking <setup.exe>, the program copies some files onto the hard disk and opens a window with (i) three installation options (installing CP, installing a database software and installing a sample corpus of Irish English) and (ii) a huge "Installation Advice Text". Among other things, this text explains, firstly, that the program is installed into the folder <C:\Corpus Presenter> and – if the program is installed elsewhere – that the links to the 27 modules must be altered manually!

Secondly, the installation process is split into two different steps. (This information is provided in the advice

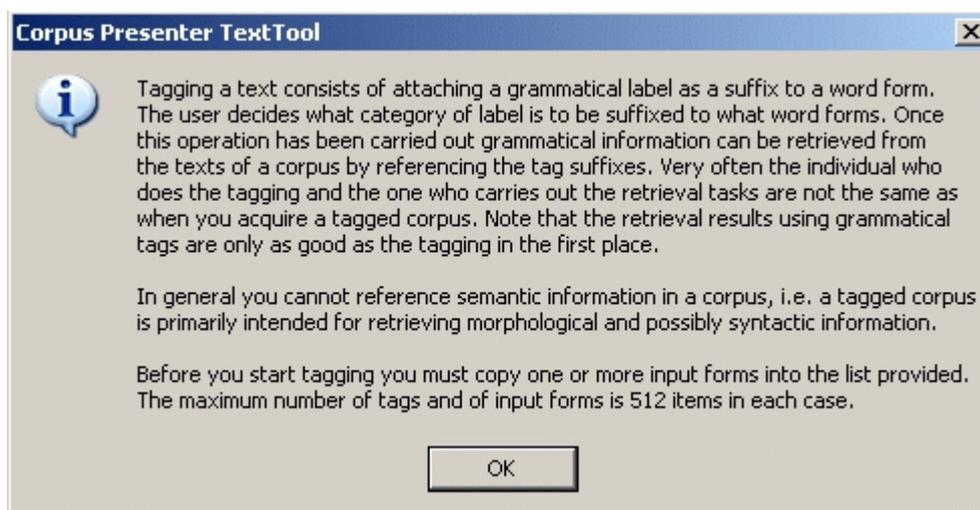
text twice, once before a list of the contents of the CD-ROM and once again after this list; the confusion is increased by the fact that, in the second occurrence of the otherwise identical text segment, a different path is used.) The first step is called on by clicking on “Installing Corpus Presenter” so that all the program files are copied to one’s hard disk; you cannot specify where the files are installed unless you manipulate Windows settings about default program directories. In the second step, Windows system files are copied. Surprisingly, you are prompted where these system files should be installed, and if you decide to install CP into a different directory (e.g. <D:\>), then the system files are copied to the system directories where they belong anyway, but the directory which you chose for installation only contains .EXE files for CP Programme Launcher and CP itself, but all the other 477 files CP has installed before are still in the directory CP mentioned before, namely, on an English windows machine, <C:\Program Files\Corpus Presenter>.

Similar comments hold for the database manager and the sample corpus: you need to install the database manager separately (which is ok), but CP expects it to be located in a particular directory without spaces in the name, and the sample corpus is simply installed to <C:\Corpus_Irish_English> regardless of where you would want to have it ...

Although RH explains in the final paragraph of this advice text that many of the shortcomings are due to the Windows operating system, it remains completely mysterious to me why the user cannot simply enter the desired path for all to-be-installed components, and the program organizes itself internally as it needs to and outputs the requisite links as is customary with nearly every other Windows program I know. The way it is now, the installation process and its result are painful if you do not know his way around Windows quite well; your system partition <C:\> is then cluttered with different directories that you would perhaps have preferred to be on an ‘applications’ partition or on a ‘corpus’ partition. In addition, the uninstallation with the Windows control panel did not remove all parts of the installation properly: the corpus as well as the files in <C:\Program Files\Corpus Presenter> and <D:\Corpus Presenter> simply remained on the hard disk.

Unfortunately, there are many more inconvenient things that falsify the claim of the maximum of the possibilities of the newer Windows system:

- Some of the programs seem to adopt the previous color settings of the desktop rather having own settings. Doesn’t sound like a big deal? Well, on a notebook with a black desktop it can result in your not being able to read the black text in the overview windows of CP Programme Launcher until you have figured the out how to change the two available and (misleading) color settings.
- When you start CP Programme Launcher, you get to see a menu bar at the top which is none: Rather than opening a menu with commands to choose from, each expression in this menu-like section is already a command in itself. For users with a well-entrenched knowledge of the Windows system, this is at first perplexing, which is why buttons should have been used here in the first place.
- Why are nearly all program windows opened such that they cover the whole screen and hide all other applications? When you turn to the help function to get information on some window, the help screen hides the window for whose options you look for clarification. When you open another module, it hides all other software which you might have needed to see (e.g. to enter data from it into CP Easy Chart) And why aren’t the windows that cover the whole screen maximized so that a click on the restore down icon would reasonably reduce the window size. And then, some windows don’t allow downsizing or maximizing at all: CP Easy Chart does – CP Flash doesn’t.
- In some programs (e.g. CP File Manager and CP Main Programme), lists of files can be sorted by clicking on a column heading (e.g. size, type etc.) – in others (e.g. CP Create Data Set), they cannot.
- Right-clicking does not always open a context-dependent menu (sometimes it just does the same as left-clicking and sometimes it just offers to perform one particular action), and in windows consisting of several horizontally or vertically separated frames, you can often not change the window parts’ relative sizes to see more of the more important information although this is of course standard in all Windows applications.
- While there is a huge amount of commands available in the ‘help’ menu (twenty in CP! – even Excel XP only has eight commands, as has WST), many of them don’t seem to belong there (what does benchmarking the system, the system information, running the graphics program CP Easy Chart or exploring CP’s home directory have to do with a user struggling with CP’s many settings and looking for help?). Also, CP does not afford Windows users the by now familiar option of a help index to enter key words describing your problem in order to retrieve a list of all help topics related to this notion. Finally, not all help texts are really useful: when I tried out the interactive tagging function of CP Text Tool, I was confronted with the window in which I had to enter the words and the tags for the tagging. Since I did not immediately understand the makeup of the window (containing four text fields, seven buttons, four fields to tick and some text information), I clicked the help button of this window. However, instead of getting information on how the information must be entered, I get the following text which is singularly unhelpful when it comes to entering the necessary information.



(If I do the same in some window in Excel, I get precise information on all buttons and all fields of the respective window ...) Also, the window offers the option to tag forms as words or strings – neither does the corresponding section of the book explain what a string is nor is the word ‘string’ listed in the index of the whole book ...

- Windows programs usually allow the user to enter data into several fields of an input window by jumping from data field to data field by pressing the TAB key; CP does the same, but – at least in CP Easy Chart, the program does not switch from one field to the immediately adjacent and thematically related one, but arbitrarily to some other field, which doesn’t make entering data any easier ...

2c) Some other functionality quibbles

The following is a list of other shortcomings of some modules which are not directly related to the integration into Windows; I begin with CP Main Programme.

- If you want to save your results (of a corpus search) such that collocates at different positions can be accessed easily, you cannot simply choose to save it as a text file with tabs as delimiters (at least I didn’t find out how). Instead, you must save it as a database file (.DBF), which entails you must use CP Data editor (or, say, Excel) to retrieve the data again and cannot use your favorite text editor etc. first.
- If a particular search of CP’s main program is interrupted, then – unlike other concordancers – CP does not present the results obtained so far; it presents none.
- If you wish to increase the number of collocates to be displayed in a results window, you do so counterintuitively by clicking an arrow pointing downwards.
- While CP can output the collocates of a particular search word, it is not quite easy to locate this option: all other concordancers I know simply have a command called ‘collocations’ (or some equally telling name), but in CP you have to find out somehow that the command (in CP Main Programme) is called ‘return lines’, which is not only very unintuitive but also somewhat difficult to find since (i) there is no help index (cf. above), (ii) *collocate* and *collocation* cannot be found using the find function of the main help text and (iii) the word *collocation* only occurs twice in the whole program folder (as determined with a grep tool), neither occurrence of which explains this function. The only way to find this option if you don’t already know it is the index of the book where the third page entry for *collocates* points you to the right page in the book for this option.
- If you want to search for bipartite expressions where one part can be instantiated by several different forms (such as inflectional forms of one lemma, say, *put*, *puts*, *putting*), then you can use the option of editing an input list – but you cannot simply edit the list by entering a few words and do a search, you must either load an existing list or enter the list manually and save it.
- Surprisingly, CP cannot sort concordance lines according to a user-specified position in the vicinity of the search word: you can only sort concordance output according to the leftmost word of a cell and the word *sort* or any derivative is not even mentioned in the index of the book, something I find strange for a program (suite) the main purpose of which is handling text(s).

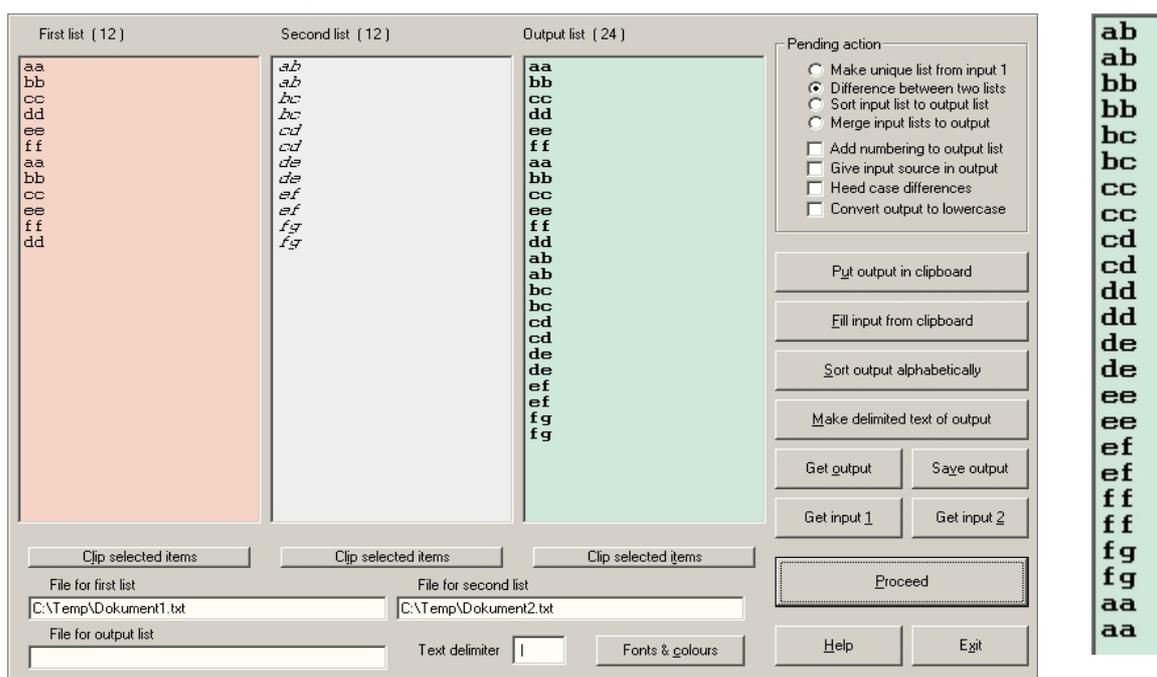
The module CP Create Data Set also deserves some comments. If you do not simply load text files as a corpus but want to compile a corpus, CP needs information on how the corpus is organized. You can either simply create a text file with a particular format with any text editor providing this information for CP by yourself or, alternatively, you can use this module. However, although the module is explained on only three pages in the handbook, it is relatively complex, and its output is the very same text file description of the corpus. In other words, one must again enter all information for each corpus file separately and manually. In addition, several

windows this module opens are not discussed in the book or the corresponding section of the help file (which are identical anyway) and handling the module is not always intuitive to say the least:

- I have not been able to find out how the order of files is changed using [CP Create Data Set](#) (other than, of course, by manually editing the text file itself);
- subheadings of a corpus must make reference to empty dummy files;
- deleting nodes from your corpus structure does not really delete the nodes until the data set file has been saved so you work with empty nodes and empty files etc.

Other shortcomings of this module are, again, due to the fact that Windows has not been utilized fully. Why can I highlight all corpus files which I want to assign to a lower structure level in my corpus, but cannot also change their level assignment all in one go? Why does this module not allow me to simply load a list of files and convert them to a dataset by providing information as to the structure of the corpus? (Guess what – you have to turn to a different module for this option, namely [CP Flash](#), but when you read the section on [CP Create Data Set](#) to find out whether such a possibility exists, the book doesn't tell you – you must find out for yourself some other way!) Why is it not possible to use drag and drop options etc. to determine the structure of the corpus? Why is there no assistant to guide you through the creation of the corpus structure (just like Excel has a guide for pivot/contingency tables and WST has a brief guide to generate a concordance)? I don't know.

There are similar usability problems throughout [CP](#). I cannot discuss all of them here since the review is already (too?) lengthy so a few final examples must suffice for the moment: First, [CP Quick Note](#) makes it possible to structure a text using embedded table of contents markers. These markers can be embedded using the very same module ... but not with the menu 'Insert' as every normal user would suspect – rather, to insert these markers the menu you have to open is called ... 'Display'. Second, the program [CP List Processor](#) allows the user to manipulate one or two lists such that, for example, the lists are merged or differences between lists are shown. However, there is a little bug in the program concerning the alphabetical output of the program: the result of the alphabetical sorting is not correctly sorted as is obvious from the screenshots below.



Finally, let us return the interactive tagging procedure of [CP Text Tool](#). You open a text file containing words to tag, and you need to have one file with words to tag and one file with tags. For the automatic tagging function, you choose the 1-512 word forms to be tagged with one tag, choose automatic tagging and [CP Text Tool](#) adds the tag to all the word forms; since you can specify more than one word to be tagged at a time, this is a huge advantage over replacing functions of, say, Microsoft Word or TextPad. With interactive tagging, the program goes through the corpus text, stops at every instance of one of the to-be-tagged word forms and asks the user which tag to add to the word form. This function is implemented a little clumsily since you are not simply prompted to choose a tag but have to use some more mouse-clicks and whenever you want to choose a tag other than the default one to assign and click on reject in this window, the list of available tags is recursively added as is shown in the succession of screenshots showed below ... Thus, although the interactive tagging function works basically ok, there is some bug here that needs correction.



3) Nitpicking, typos etc.

This section is concerned with only minor errors and some other short comments/questions in a simple list form.

Concerning the book:

- On p. 8., the first line of the second paragraph of section 3.5 is garbled.
- On pp. 5f., 42 and 164f., RH mentions the normalization of corpora, but with the exception of one example buried within a table he restricts himself to normalizing spelling variants; the issue of lemmatization would have deserved more emphasis here (for analyses of author style or collocations) but it is only mentioned once in the glossary (though, surprisingly, not in the index);
- The notion of tagging is explained briefly under the rubric of '3.2 Versions of corpora' in three paragraphs (p. 5) and once again under 'Tagging a corpus' on p. 8f. – why not put this together?
- Why are the BNC and the Cobuild Bank of English not mentioned at all (not even in the glossary although this includes several other entries of words not mentioned in the book; cf. below) although they are probably the most widely distributed and available corpora?
- The brief explanations of central corpus-linguistic terms in Part I, sections '3 Preparing corpora' and '4 Examining corpora' are very brief, having an average length of about two to three paragraphs only and thus cover these issues only very superficially.
- On p. 22f., RH discusses an example of collocate analysis, namely the frequency of particular collocations for *deal* in the London-Lund Corpus. He states that "139 finds were reported in 15 files. 57 finds were with *great* as immediate left collocate and 41 finds with *good* in the same position. The results in a visually effective form can be shown as a pie chart [...]." However, the pie chart to which RH refers reports percentages which do not fit the data described in the text. 57 [*great deal*] out of 139 [** deal*] are 41% and not the 60% represented in the chart; the same holds for *great*: 41 [*good deal*] out of 139 [** deal*] are 29.5% and not 38.89% ...
- On p. 67, RH describes the command 'Frequently asked questions' in the module [CP Main Programme](#) as follows: "The third text file aims at answering typical questions which users might come up with who have started working with the present corpus. This file should preferably be written by someone who has been connected with the compilation of the corpus." However, when the test corpus is loaded and one tries to access the FAQ for this corpus using this option, what one gets is the FAQ for the [CP Main Programme](#), not for the corpus and it is unclear why this should have been "written by someone who has been connected with the compilation of the corpus."
- On p. 72, section 1.3 ('Normalising texts') consists of one paragraph only and accidentally interrupts an otherwise coherent and numbered description of the search parameters available in [CP Main Programme](#).
- On p. 73, RH gives an example of a word search using the frame option by suggesting that "if you wished to find all instances of negated adjectives in a text then you could enter a frame consisting of *un* and *able* [...]" – obviously, this search would not produce *all* negated adjectives since *inadequate* and *impossible* would not be retrieved.
- On p. 81, RH states that the file extension for the output of a concordance as a plain text file is .OUT, but in the program it's .TXT.
- On p. 92, a sentence runs "This very is important for [...]."

Some minor comments concerning the glossary:

- Some of the explanations of statistical terms in the glossary (which are not mentioned in the book otherwise) are far from optimal. For example, to define an alternative hypothesis as "an assumption in statistics that two variables are different" (p. 211) is perhaps a little too low-level even for a glossary

definition.

- Similarly, the definition of a Chi-square test (p. 213) is grammatically incorrect (“A common test in linguistics is to determine if the probability that a difference between sets of values is due to chance alone.”) and much too vague since the above sentence also characterizes a t-test, a U test, an ANOVA etc. Also, I am not sure that most scholars would subscribe to the following statement: “A typical cut-off point for significance is $p < 0.001$ ” (p. 213).
- I do not know why a corpus processing book needs glossary entries for *cookie*, *email*, *inkjet printer*, *laser printer*, *PC*, *RS232C* and *TFT*.
- I would not equate *lemma* and *lexeme*.

I already mentioned that large parts of Part II of the book are just the help files of the program, but sometimes the book itself is also a little redundant. For instance, a part of the general description of CP’s main module on p. 30f. is repeated verbatim in the more detailed section on p. 48f. In addition, sometimes the names of the modules used in the book are not identical to the names of the modules used in CP Programme Launcher, the application from which RH recommends to access all other modules. For example, the book has sections on CP Easy Chart and CP Structure, but in CP Programme Launcher the very same modules are called CP Chart Generator and CP Structured Texts respectively. This is of course no big deal (which is why it is in this nitpicking section of the evaluation in the first place), but, just like the fact that the book doesn’t discuss the modules in the same order in which they are listed in CP Programme Launcher although (i) this would be easier to follow and (ii) perfectly possible since the listing in CP Programme Launcher is completely arbitrary, it simply does not speak in favor of careful editing.

Unfortunately, the book is not very well organized in terms of software learnability either, which is probably a direct consequence of Part II of the book largely being the help files. It would have been extremely useful if the book had provided at least one sample analysis which is designed in such a way as to lead the beginning user through the many modules (perhaps in combination with a website) unlike the sample analyses in the book which make no reference at all to how exactly they would have been generated with CP. Let me give an example of what I have appreciated very much. One could have some corpus files on the publisher’s webpage which one would then turn into a hierarchically structured corpus using the text editing modules and CP Create Data Set. Then this corpus could be tagged and lemmatized using CP Text Tool. Then one could perform a sample analysis on the basis of this corpus, for example the collocational differences of *strong* and *powerful* (to use the textbook example) with CP Main Programme or CP Flash and use finally use CP Easy Chart and CP Slide to prepare a presentation of the results and CP Internet Editor to present the results on a website. For all of this, the website could provide interim results to allow the user to check whether he has mastered the tasks so far. Sadly, however, none of this is provided although it would have enhanced the value and quick learnability value of the bundle by many orders of magnitude.

Concerning the software:

- The installation advice text contains the same paragraph twice (with different paths, though); cf. above.
- In the rubric ‘Retrieving information’ of the CP help, there are three paragraphs §2.
- Sometimes, when a window outputs certain results, you can change the size of the window as such, but not the size of the part of the window that contains the output; i.e. you get a larger window, but just a larger frame, not more information.
- The fact sheet for the installed test corpus start with “Some essential facts about the Test Corpus.”
- Sometimes, the program uses somewhat idiosyncratic commands: instead of clicking on ‘ok’ to close a window and accept what one has entered/changed, one has to click on ‘conclude.’

Lastly, although CP is a very recent program, it does not have some of the added-value gimmicks that competing programs offer (it is only fair to repeat here that it of course also has functions these competitors do not have). For example, CP does not provide corpus-based statistics such as indices of collocational strength etc. (like, say, Michael Barlow’s Collocate). Also, although the issue of analyzing style is brought up repeatedly in the book, CP does not allow for the automatic identification of key words in texts (unlike WST).

4) Conclusion

All in all, I am the first to admit that CP is a program that offers many functions that can be useful for the compilation, annotation and processing of corpora. I also freely admit that the evaluation of usability is by definition a relatively subjective task. I also believe, however, that many of the flaws I have pointed above would render the program much more difficult to use than competing products. From what I have seen, the (only) positive side I have been able to detect is in fact the large number of functions, i.e. the ‘what the program does’. But, the negative sides of CP are the ‘how the program does it’: (the larger part of) the program is

- difficult, inconvenient and counterintuitive to handle, sometimes violating even elementary usability issues;

- overloaded with many redundant functions (containing four text editors alone) that are part and parcel of regular operating systems and office software;
- painfully slow to execute even some of the most basic concordancing tasks.

Note in this connection that many functions of CP (other than the hierarchical corpus compilation functions of course) are available as (parts of regular) office suites and freeware programs. In addition, the software book is largely identical to the many help files that come with the software and sloppily edited in many ways. Although I had been waiting quite some time for the program after having it seen announced as commercially available soon, I am rather disappointed with the final result and hope that the most frustrating bugs will be considered for an update soon.

Biographical sketch

Stefan Th. Gries is Associate Professor at the Department of Business Communication and Information Science at the University of Southern Denmark. His research interest mainly lies with corpus linguistics and linguistic methodology, esp. the syntax-lexis interface as well as corpus-based, quantitative approaches to word-formation processes (e.g. blending and suffixation), syntactic variation (dative movement, particle movement etc.) and semantic issues (near synonyms, word senses etc.). He is currently co-editing two volumes on corpora in cognitive linguistics and is also one editor-in-chief of a new journal, Corpus Linguistics and Linguistic Theory, to be launched in 2005.

Raymond Hickey's reply to this review sent to me personally on 18 March 2004

Return-path: <r.hickey@uni-bonn.de>
Delivery-date: Wed, 17 Mar 2004 20:32:18 +0100
Received: from [194.97.50.136] (helo=mx3.freenet.de)
by mbox66.freenet.de with asmtmp (ID exim) (Exim 4.30 #6)
id 1B3gm1-0008Ti-Bf
for stgries@01019freenet.de; Wed, 17 Mar 2004 20:32:17 +0100
Received: from msec2.sdu.dk ([130.225.156.40])
by mx3.freenet.de with esmtmp (Exim 4.303 #3)
id 1B3gm0-0006xY-Uc
for stgries@freenet.de; Wed, 17 Mar 2004 20:32:17 +0100
Received: from ADM-EXCH2B.adm.c.sdu.dk ([130.225.160.5])
by msec2.sdu.dk (8.12.11/8.12.11) with ESMTP id i2HJWFX3002055
for <stgries@freenet.de>; Wed, 17 Mar 2004 20:32:16 +0100
Received: from msec1.sdu.dk ([130.225.156.41]) by ADM-EXCH2B.adm.c.sdu.dk with Microsoft
SMTPSVC(5.0.2195.5329);
Wed, 17 Mar 2004 20:32:16 +0100
Received: from f2node23.rhrz.uni-bonn.de (node23.rhrz.uni-bonn.de [131.220.14.223])
by msec1.sdu.dk (8.12.11/8.12.11) with ESMTP id i2HJWETS000568
for <STGries@sitkom.sdu.dk>; Wed, 17 Mar 2004 20:32:15 +0100
X-Spam-Filter: check_local@msec1.sdu.dk by digitalanswers.org
Received: from my-computer.uni-bonn.de (ascend-tk-p145.dialin.uni-bonn.de [131.220.244.145])
by f2node23.rhrz.uni-bonn.de (AIX4.3/8.9.3/8.9.3) with ESMTP id UAA35132
for <STGries@sitkom.sdu.dk>; Wed, 17 Mar 2004 20:32:08 +0100
Message-Id: <5.1.0.14.0.20040317202146.00a7f0e0@mailin.uni-bonn.de>
X-Sender: uzr60006@mailin.uni-bonn.de
X-Mailer: QUALCOMM Windows Eudora Version 5.1
Date: Wed, 17 Mar 2004 20:32:49 +0100
To: STGries@sitkom.sdu.dk
From: Raymond Hickey <r.hickey@uni-bonn.de>
Subject: Your review of Corpus Presenter in the Linguist List
Mime-Version: 1.0
Content-Type: multipart/alternative;
boundary="===== _4839734== _ALT"
X-Virus-Scanned: by amavisd-new
X-OriginalArrivalTime: 17 Mar 2004 19:32:16.0213 (UTC) FILETIME=[8DFD6050:01C40C56]
Delivered-To: stgries@freenet.de
Envelope-to: stgries@freenet.de
Delivered-To: stgries@01019freenet.de

--===== _4839734==_ALT
Content-Type: text/plain; charset="us-ascii"; format=flowed

Dear Stefan Gries,

It has been brought to my attention that you wrote a review of my book + software Corpus Presenter for the Linguist List. This appeared at the end of January which is some six weeks back and I feel that it is too late to participate in a public discussion at this late stage. Nonetheless, I feel I should write to you to express my dissatisfaction, not with the legitimate critical points you make, which I have taken on board: You rightly noticed a bug in the List Processor which I have rectified and I have given the new version to many colleagues, it is freely available to me. Furthermore, I accept your criticism on the structure of some help files and have been working on updates for these. However, you know that there is a FAQ text in the Launcher which is specifically designed to tell users what the programmes can do and how they realise their goals with them.

My main disagreement is with the general approach to reviewing which you adopt. I have been living in Germany for 27 years and find that your review illustrates all too sadly a type of petty-minded pea-counting which is so common among German academics. What you should have done - but clearly failed to do - is to consider first of all in broad terms what this book + software intends to do. As the name says, it is concerned with presenting texts in a tree-structured fashion which allows users to search flexibly among the texts of a corpus presented in this manner. This the programme does and an entire corpus, my own A Corpus of Irish English, along with a test corpus are given with the programmes so that users can see in advance what needs to be done. The test corpus can be edited easily to produce trees for corpora which users might wish to construct. In your review you neither mention the presentation of corpora nor the Irish English corpus. Furthermore, there are supplied trees for existing corpora including the Helsinki corpus - this would be reason enough for using the software. The main programme also offers many flexible ways of exporting data from retrievals, over and above the means offered in other programmes.

One of the main strengths is the sophisticated type of text retrieval (syntactic frames) possible with Corpus Presenter. These are not mentioned by you at all. If you were a balanced and competent reviewer you would have discussed these elements and highlighted them, especially as the flexibility for syntactic searches is on a much higher level than WordSmith, for instance.

Corpus Presenter is supposed to be an *alternative* to other programmes like Word Smith and offers a sophisticated interface not found in programmes like the latter. True there is a trade-off and this is one of speed. I grant you that, but there is gain in terms of flexibility - that is something which you should have mentioned. Another point: If you don't like the number of utilities in the suite, then just ignore them. They are not necessary for the operation of the main programme and each of them was included because of requests from users (I assure you this software was tested over years). But whatever you do, stop complaining because you have got too much!

There are a number of factual misrepresentations in your review. It certainly is possible to sort concordance returns on words to the left and right of the keyword (this can be done for up to 8 words each side of the keyword). By the way, the term "restructure return lines" is perfectly acceptable as a description of this function. I suspect that you may have difficulties with the English language - your review certainly shows that -

and you also complain of a sentence which you quote as ungrammatical which is perfect grammatical. So, be careful before you fall into a frenzy of petty-minded quibbling.

I must stress again, writing such software, one has to be all things to all people. I spent years trying to fulfil the needs of colleagues in the history of English studies and feel that I have done this fairly well (and have had sufficient feedback which confirms this). At the very least as a reviewer you should have taken this fact into account, it would have put your own criticism into better balance.

There is another point which is important to stress: in your "korinthenkackerische" manner you complain about the installation programme. Did it ever dawn on you for a single moment, that the structure of this installation is determined by Windows and not due to my poor programming as you seem to imply? The reason the database manager and the rest of the software are separate is that they were written with different compilers, did that ever strike you? I also find your endless minute complaints about small points of command structure more than irritating, e.g. complaining when an option leads to a direct command, rather than a pull-down menu. For crying out loud, Gries, this is utterly ridiculous criticism! May I add that the Launcher makes the use of the suite a cinch, something which has been confirmed by the many colleagues who have been using it. Again, it is petty-minded for you not to realise and acknowledge this.

I would like here to put you straight about another important matter which you have criticised but which shows that you have not looked at the programme suite properly: Towards the end you say:

Lastly, although CP is a very recent program, it does not have some of the added-value gimmicks that competing programs offer (it is only fair to repeat here that it of course also has functions these competitors do not have). For example, CP does not provide corpus-based statistics such as indices of collocational strength etc. (like, say, Michael Barlow's Collocate). Also, although the issue of analyzing style is brought up repeatedly in the book, CP does not allow for the automatic identification of key words in texts (unlike WST).

But CP Text Tool has a function for Lexical Clustering which does precisely that!! So before you open your mouth so widely, I suggest you check on what the software can do. And just in case you say this is buried somewhere in a small utility, it is not: Lexical Clustering is mentioned on several occasions, including the various guides available within the Launcher.

There are also many minor points which show that you did not understand the programmes at all. For instance, you mention the following.

- In some programs (e.g. CP File Manager and CP Main Programme), lists of files can be sorted by clicking on a column heading (e.g. size, type etc.) - in others (e.g. CP Create Data Set), they cannot.

You are not supposed to sort the items of a corpus tree when editing it! For God's sake, that should be more than obvious. It is **completely** unfair of you to mention this fact in this manner. The unsuspecting reader will think I have missed out on something here. The contrary is the case: this is simply due to your dull-witted lack of understanding!

On your concluding criticism:

- difficult, inconvenient and counterintuitive to handle, sometimes violating even elementary usability issues;

RH: This is your opinion, not mine obviously.

- overloaded with many redundant functions (containing four text editors alone) that are part and parcel of regular operating systems and office software;

RH: Again, if you don't want the other modules, don't bother with them (you can delete them if you want to, the main programme will still work)

- painfully slow to execute even some of the most basic concordancing tasks.

RH: I am the first to admit that it is slower than WordSmith. This is a DOS application, transported to Windows and written in Delphi (I think). So it has very much a "bare bones" feel to it, and reaches its speed by the small number of options. But I have discussed this with so many colleagues and found that for them, the flexibility in searches, returns and interface is more important, especially when using more individualised corpora like those available for the history of English.

Yours,

Raymond Hickey

Raymond Hickey's posting to the LinguistList (15.981), dated 23 March 2004

It was drawn to my attention that a review of my book and software was posted to the Linguist List by one Stefan Gries, a German academic working at the University of Southern Denmark. I must confess I have not been following all the reviews on the Linguist List which is why this one missed my usually watchful internet eye.

I read the review carefully and thought about it for some days. Given the negative and one-sided nature of the review I naturally contemplated writing a rebuff. But quite obviously no one is interested in an altercation between a reviewer and the author of a book. However, because Gries's review is skewed, not to say on many points downright wrong, I felt that it really is necessary to try and put straight some of the more serious misrepresentations contained in his review and – importantly – to mention some of the features of the software which he does not discuss. I will keep my reply to an absolute minimum and therefore not take him to task for each and every inaccuracy in his review. I should add that I have never heard of Stefan Gries before so there is no personal motivation in providing this reply.

Gries concentrates on a small selection of features in his review. In his discussion of tagging and of concordance building with large corpora he provides sensible criticism which I have already reacted to by improving the two programmes which come in for particular criticism (see "Getting updates" below). But the main drawback of his review is that he only mentions those aspects of the software which interest him. The entire thrust of Corpus Presenter, the presentation of corpora in intuitive tree form and the sophisticated retrieval techniques for finding syntactic and morphological constructions in such corpora, are not even mentioned by him. A Corpus of Irish English which is supplied with the book and which can be used as instructional software for teaching about corpus linguistics, for example, is not discussed by Gries at all. Gries also complains about the structure of the book: this was discussed with the publishers and vetted by their reviewers. If he doesn't like it I can't help it but he can rest assured that there were good reasons for presenting it the way it is.

Gries compares Corpus Presenter with other corpus processing software and finds it wanting. My software is not an attempt to replicate WordSmith or any other software tool which is already available. Instead it offers a suite differently structured with different aims and organisation. I am not interested in reinventing the wheel and if Gries prefers WordSmith then he should stick to it. Many of the retrieval options, particularly for syntactic frames, are only available in Corpus Presenter and the retrieval options show a flexibility which is not found in other software. It is true that for huge corpora, Corpus Presenter is slower than WordSmith (though I doubt Gries statistics). More flexibility does have a price, but speed is not everything, functionality and range of options are

equally important and anyway with the increasing speed of computers, this question is not the burning issue it was years ago.

I take grave exception to Gries talking of “bugs” in his final remarks. Corpus Presenter works properly and fulfils the functions which it claims to perform (Gries acknowledges this, if only begrudgingly). I have over 20 years of experience in computing and am a committed programmer who spent several years and had many, many discussions with colleagues around the world in the preparation of this software. Gries does not appear to like the interface of the programmes (there is no accounting for taste), but it is quite unjustified of him to go on and on for several pages complaining about the features the software suite contains. He seems to find too many: all the utilities supplied in the CD are the result of suggestions by colleagues in the field who felt the need for the functions they fulfil. Now if Stefan Gries does not like these utilities he can just ignore them. It is stated explicitly that they are not necessary for the functioning of the main programme and he can delete them if he wishes. Furthermore, there are clear instructions in the main programme about how to remove all the software and auxiliary files. His frenzy of petty-mindedness in discussing the installation procedure is patently absurd. I might add that the setup is determined by Windows and not by me. If he does not like it, then I suggest he directs his comments to Microsoft and see if he can find anyone to listen to him.

There are a number of factual misrepresentations in Gries’s review. To keep things short I will only mention two serious ones. (1) It certainly is possible to sort concordance returns on words to the left and right of the keyword (this can be done for up to 8 words each side of the keyword). Gries does not like my terminology – “restructure return lines” – but as a native speaker of English I beg to maintain that this is an acceptable description of this function. (2) Gries thinks that the analysis of style is not treated in Corpus Presenter, but the special text editor, CP Text Tool, has a function for Lexical Clustering analysis which does precisely that. It will allow users to determine the occurrence of stylistic features in a flexible manner and so help them answer such questions as text authorship. Lexical Clustering is mentioned on several occasions, including the various guides available within the Launcher so Gries should have seen this is if he had looked at the material properly.

Getting updates:

The updates for the Corpus Presenter software can be downloaded at my homepage “www.uni-essen.de/~lan300/HICKEY.htm”. Here you click on “Computer Projects” and then on “Corpus Presenter” to get to the page you need (this cannot be done directly unfortunately). Then click on “Updates” and download what the files you require. There is an update for CP Flash Processor (greatly increased speed of directory listings) and CP Text Tool (better interface for tagging texts) as well as a text on tagging and a list processing utility.

Raymond Hickey
Tuesday, 23 March 2004

My reply to Raymond Hickey’s as posted to the LinguistList (15.1087), dated 24 March 2004

For a more reader-friendly version of my review, please go to http://people.freenet.de/Stefan_Th_Gries/Research/CP_review.pdf. Apart from all communication on this matter (including this text), this page provides some screenshots of the CP and, in the interest of locating the discussion of different modules of CP, highlights all occurrences of CP’s program names in blue). In what follows I would like to briefly comment on some of the points raised in Hickey’s response to my review. The first part is concerned with aspects of Hickey’s book/software (henceforth CP), the second with the more general tone of Hickey’s response.

I agree with Hickey that my review did not cover all aspects of the book/software that one could have mentioned. The book alone comprises nearly 300 pages and the software offers an extremely vast range of functions (as I also state in my review), so every review – especially when due in maximally six weeks time – is by necessity selective even though the review is already extraordinarily long: Given the vast range of functions the program has to offer, I think it is only natural that not all features can be discussed (to the satisfaction of the author).

Then, it goes without saying that I accept responsibility for, and regret, all errors or misrepresentations that are not due to my selective emphasis although below I also have something to say about how they arose in the first place. However, it is necessary to set straight some of the points that Hickey criticizes about my review and I will also demonstrate below that some points of critique mentioned by Hickey are due to his not having read the review thoroughly enough.

As a first example, Hickey complains about my having neglected the presentation of corpora, the retrieval

techniques and the Corpus of Irish English, but on the one hand, the review does mention that CP can be used to compile and annotate corpora hierarchically and it mentions the installation of the Corpus of Irish English. On the other hand, while I certainly agree that I could have discussed these issues more prominently, I found it more important to discuss the way CP deals with corpora in general rather than about one corpus in particular and consider this a reviewer's legitimate option.

Second, Hickey objects to my comparing CP with other software. I do not see why this is problematic. Sure, CP is a tool that is very different from competing products such as MonoConc Pro 2.2, WordSmith Tools 3/4 (WST) and WinConcord 2 – no doubt about that – but (i) the by far largest part of the review is not concerned with comparing CP to competing products anyway (WST is mentioned 11 times (on [9 pages with 6,544 words]) and (ii) I do not see any reason why I as a reviewer should not be allowed to compare selected aspects of a program with competing applications. Moreover, I do not know how well Hickey knows the competing products he refers to: (a) He simply affirms that CP offers the most flexible retrieval tools without offering any evidence (and MonoConc Pro's retrieval options are in fact extremely versatile). (b) Where I give evidence about the speed of the program, Hickey simply states he doubts my evidence (which, as I mention in my review, are the statistics CP itself outputs!) but does not provide a single argument or perhaps a comparable figure to support his point. Lastly, while I do agree that some may consider speed less relevant nowadays, my own experience is different: searching the BNC or merely larger parts of it for regular expressions of varying degrees of complexity or determining significant collocates for tens of thousands of adjectives can be so time-consuming that speed sometimes matters quite a lot. Be that as it may, reporting performance statistics cannot really be wrong by definition ...

Third, let me now turn to the so-called factual misrepresentations. As for one, Hickey states that, contrary to what I wrote, “[i]t certainly is possible to sort concordance returns on words to the left and right of the keyword (this can be done for up to 8 words each side of the keyword).” If the function ‘restructure return lines’ is activated for a particular concordance, CP outputs a bipartite window, the lower part of which provides absolute and relative frequencies of collocates of some defined span (1, 2, 3, 4, 6, 8 words on one or two flanks – why not also 5 and 7?). The upper part of this window now provides a part of each concordance line in tabular form such that the line is split into as many slots as were previously defined. It is true that this part of the output can then be sorted (if you want to sort according to the first word of the right of the node/search word, you must click on its column name, which is “Item 1” just like the name of the first word to the left) and copied to the clipboard (and only then to disc), but it is not the complete concordance line that can be dealt with this way, it is only the previously defined part of the concordance line that is sorted accordingly, which is why context further away from the node/search word cannot be accessed this way. Thus, unless I have missed some other function, I do not see my statement disproven.

In this connection, Hickey states “Gries does not like my terminology – ‘restructure return lines’ – but as a native speaker of English I beg to maintain that this is an acceptable description of this function.” While Hickey simply skips over my point of critique that this function is too difficult to locate since no help index is available and no entries for “sort” or “restructure return lines” (! ;-)) exist in the index of the book, he is completely right: I do not like his terminology. I pointed out some other idiosyncratic names of functions in his program, and I just leave it to the readers to decide whether it is really just due to chance that most, if not all, other programs use the command “sort” for sorting as do some programming languages (e.g. Python and R language, which also has “order”).

In order to address the only other case of “factual misrepresentation” Hickey cares to mention (in spite of the multitude of errors he implies there are), I have to quote him again at length. He states

Gries thinks that the analysis of style is not treated in *Corpus Presenter*, but the special text editor, *CP Text Tool*, has a function for Lexical Clustering analysis which does precisely that. It will allow users to determine the occurrence of stylistic features in a flexible manner and so help them answer such questions as text authorship. Lexical Clustering is mentioned on several occasions, including the various guides available within the *Launcher* so Gries should have seen this if he had looked at the material properly.

Unfortunately, Hickey himself has not cared to read my review with the necessary attention. Here's what I said in the review:

Lastly, although CP is a very recent program, it does not have some of the added-value gimmicks that competing programs offer (it is only fair to repeat here that it of course also has functions these competitors do not have). For example, CP does not provide corpus-based statistics such as indices of collocational strength etc. (like, say, Michael Barlow's Collocate). Also, although the issue of analyzing style is brought up repeatedly in the book, CP does not allow for the automatic identification of key words in texts (unlike WST).

As it turns out, CP can really not output collocations statistics, but more importantly, while it can output lexical clusters in the way that WST can, it cannot compute key words as defined in WST, which I explicitly referred to. Key Words in WordSmith takes two corpora (one ‘research corpus’, one ‘reference corpus’), checks the

frequencies of all words in both texts and then outputs key words sorted by their p-values (where key words are words which are significantly overrepresented within the research corpus as compared to the reference corpus [measured in terms of Chi-square tests or the Log-likelihood test]). Hickey's Lexical Cluster Analysis does not do this, and I have not been able to locate any other such function in his program, which is why this claim of his remains as much in need of support as many others.

Let me turn to the final factual point. I pointed out before that my main problems with CP do not derive from its functionality, i.e. what the program can do. Let me state it as directly as possible: the functionality of CP is great, it can do more than any other corpus program I have ever seen. My main quibble is with usability, i.e. how the program lets you do it. I do not wish to bore the readers with all the details of the original review but refer them to it instead, but let me just say two sentences about Hickey's complaint that I devote too much space to criticizing the many utilities CP offers. First, I think it is only fair to point out to the potential buyer that many of the twenty-something modules the program contains just do what the operating system or other freely available software can already do. For some people, this may be an interesting argument against the bundle, and thus this is something that should be pointed out in a review. Second, the usability of a program is not enhanced by crowding it with many modules or functions one is later invited to delete or neglect – rather, a program should offer its functionality in such a way that the user can make intuitive choices from a reasonably small set of alternatives: a help menu with twenty different entries that even includes a command to benchmark the system is simply not the most usable way to design a program, and the fact that no other program goes to similar extremes testifies to this point.

As to the book, Hickey objects to my lack of appreciation of the structure of the book and that there were good reasons for it. I cannot substantially comment on this point since Hickey does not mention the good reasons he alludes to but just states that, if I don't like the structure, he can't help it. This is doubtlessly correct, but neither does it constitute a rational argument nor do I see why I as a reviewer should not be entitled to criticize the structure of a book (not to forget the many editing errors / typos) especially when I also outline a constructive proposal as to how a from my point of view didactically more feasible structure may look like, as I do at the end of my review. I am, however, very happy to learn that my review has – in spite of all its limitations – already resulted in some bugs being fixed.

Let me finally say something about the general tone of his reply. First, I (and at least three other colleagues who have read his reply) cannot fail to notice the *ad hominem* undertone underlying (parts of) his reply. I do not see in what way it is relevant to a discussion of the merits (or lack of them) of my review that I am “one Stefan Gries”, “a German academic at the University of Southern Denmark”, or that Hickey has never heard of me before. Similarly, Hickey states that “Corpus Presenter works properly and fulfils the functions which it claims to perform (Gries acknowledges this, if only begrudgingly)”. What is the purpose of salting his reply by attributing such emotional states to me? Neither did CP perform all the functions Hickey claimed it to perform – remember Hickey's own statements about the bugs he fixed as a reply to my review? why is there already an upgrade if CP already performed all functions as intended? – nor is there any statement in my review that can straightforwardly be interpreted as *begrudgingly* if one has not already built up some prejudices. Quite the contrary: I mentioned clearly that I had been looking forward to the program quite some time after having it seen announced as commercially available soon. And as usual, note that Hickey simply attributes this emotional state to me, but – as before – does not cite any sentence whatsoever to support this attribution. I would have welcomed a more sober exchange than the one that has now actually taken place, but it is instructive in this connection to not just consider the reply Hickey posted to the LinguistList, but also the reply he had sent to me personally earlier, in which a compound involving a vulgar German verb for *to defecate* plays a prominent role in characterizing my review (go to <http://people.freenet.de/Stefan_Th_Gries/Research/CP_review.pdf> to access the original review with example screenshots and all following communication). This will therefore be my final statement in this matter.

Stefan Th. Gries
University of Southern Denmark