AntConc: Design and Development of a Freeware Corpus Analysis Toolkit for the Technical Writing Classroom

Laurence Anthony
Waseda University
anthony@antlab.sci.waseda.ac.jp

Abstract

In this paper, I will describe AntConc, a freeware, multi-platform, multi-purpose corpus analysis toolkit, designed by the author for specific use in the classroom. AntConc includes a powerful concordancer, word and keyword frequency generators, tools for cluster and lexical bundle analysis, and a word distribution plot. It also offers the choice of simple wildcard searches or powerful regular expression searches, and has an extremely easy-to-use, intuitive interface. After explaining the background to AntConc, I will give an overview of each of its tools, and explain their value to learners. Then, I will discuss the current limitations of the software, before explaining how these will be addressed in the future.

Keywords: *corpus linguistics, concordancer, collocation, software, educational technology, vocabulary*

1 Introduction

Over the past ten years, corpora of language data have started to play an increasingly important role in determining how languages are taught.[1] As Chapelle [2, p. 38] describes, there appears to be a 'corpus revolution.' Corpora have started to be applied in a wide range of areas, including translation studies, stylistics, and grammar and dictionary development.[3] In the classroom, Johns [4] proposes that learners can use corpora to investigate for themselves the way that language is used in target contexts, in a so-called 'data-driven' approach to learning. This can be particularly effective in the technical writing classroom, as the learners are often from a variety of different fields each with its own set of characteristic language features.[5] In such a context, it is almost impossible for the teacher to know all these features and deal with them on a one-to-one basis. As Chapelle [2] argues, a further advantage of 'data-driven' learning with corpora is that it helps to develop corpus analyses skills that will serve learners long after the course has finished.

A corpus of language is virtually useless without a computer software tool to process it and display results in an easy to understand way. Although many concordancers and corpus analysis programs now exist, including *WordSmith Tools*[1], *MonoConc Pro*[2], and WordPilot[3], few have been developed specifically for learners in a classroom context. Rather, they have tended to be aimed at researchers, and thus either include a wide range of features rarely needed by most learners (for example *WordSmith Tools*), or a very limited number of features sufficient to perform only a specific task (for example *Web Concordancer*[4]). In addition, the design of the graphical user interface (GUI) has been less of a concern in most cases, resulting in overly complex or rudimentary interfaces that lack the familiar feel of a modern windows based application.

In this paper, I will describe the development of *AntConc*[5], a corpus analysis toolkit designed by the author for specific use in the classroom, that includes a powerful concordancer, word and keyword frequency generators, tools for cluster and lexical bundle analysis, and a word distribution plot. *AntConc* is a freeware, multi-platform application, making it ideal for individuals, schools or colleges with a limited budget running either Windows or Linux/Unix based systems. Also, it is distributed as a single executable file that can be simply copied onto the user's computer and launched without requiring any installation.

0-7803-9028-8/05/$20.00 © 2005 IEEE.

In Section two, I will describe the background to *AntConc* and give a summary of its features. In Sections three to seven, I will give an overview of each of its tools, and explain their value to learners. Then, I will detail the current limitations of the software in Section eight, before explaining how these will be addressed in the future in Section nine.

2 Background and Summary of Features

AntConc was first released in 2002. At the time, it was a simple KWIC (Key Word in Context) concordancer program designed for use by over 700 students in a scientific and technical writing course at the Osaka University Graduate School of Engineering. *AntConc* was developed in a Windows environment using the PERL 5.8 programming language, and the graphical user interface (GUI) was developed using the PERL/TK 8.0 toolkit. This enabled the program to be easily ported to a Linux/Unix environment, which was necessary as the course was initially taught in a Linux based CALL (Computer Assisted Language Learning) laboratory before being moved to a Windows based CALL laboratory the following year.

Following the release of *AntConc* 1.0, the program was uploaded to the author's website from which researchers, teachers, and learners around the world could easily download and use the software free of charge for non-profit use. This generated wide interest in the program and many users reported on successes, problems and features they would like to see added, resulting in new, improved versions of the software. Interest in the program increased further after it was chosen to be included in Morphix NLP[6], a CD linux distribution containing a wide range of natural language processing (NLP) tools.

At the time of print, the latest version is *AntConc* 3.0. It was released in December 2004, and includes numerous tools and features, as summarized in Table 1.

3 Concordancer Tool

The central tool used in most corpus analysis software, including *AntConc*, is the concordancer. As Sun & Wang [6] describe, concordancers have been shown to be an effective aid in the acquisition of a second or foreign language, facilitating the learning of vocabulary, collocations, grammar and writing styles. For example, research has shown that new vocabulary can only be acquired through meeting words in diverse natural contexts [7] and in varied situations.[8] Based on only intuition, it is almost impossible to find a sufficient number of examples of a specific word or phrase to satisfy these conditions. On the other hand, using a reasonably large corpus, a concordance program can find and display a huge number of examples in varied contexts and situations quickly and efficiently.

Figure 1 shows a screenshot of *AntConc* while a

Table 1. Summary of Tools and Features in AntConc 3.0

- **Freeware License**
- **Small memory requirement (~2 MB of disk space)**
- **Multiplatform**
  - Windows 95 or later
  - Unix / Linux
- **Extensive set of text analysis tools**
  - KWIC Concordance
  - Search Term Distribution Plot
  - Original File View
  - Word Clusters / Lexical Bundles
  - Word lists
  - Keyword lists
- **Powerful Search Features**
  - Regular Expressions (REGEX)
  - Extensive Wildcards
- **Multiple-Level Sorting**
- **HTML/XML Tag Handling**
- **Unicode Support**
- **Easy-to-use, intuitive GUI**

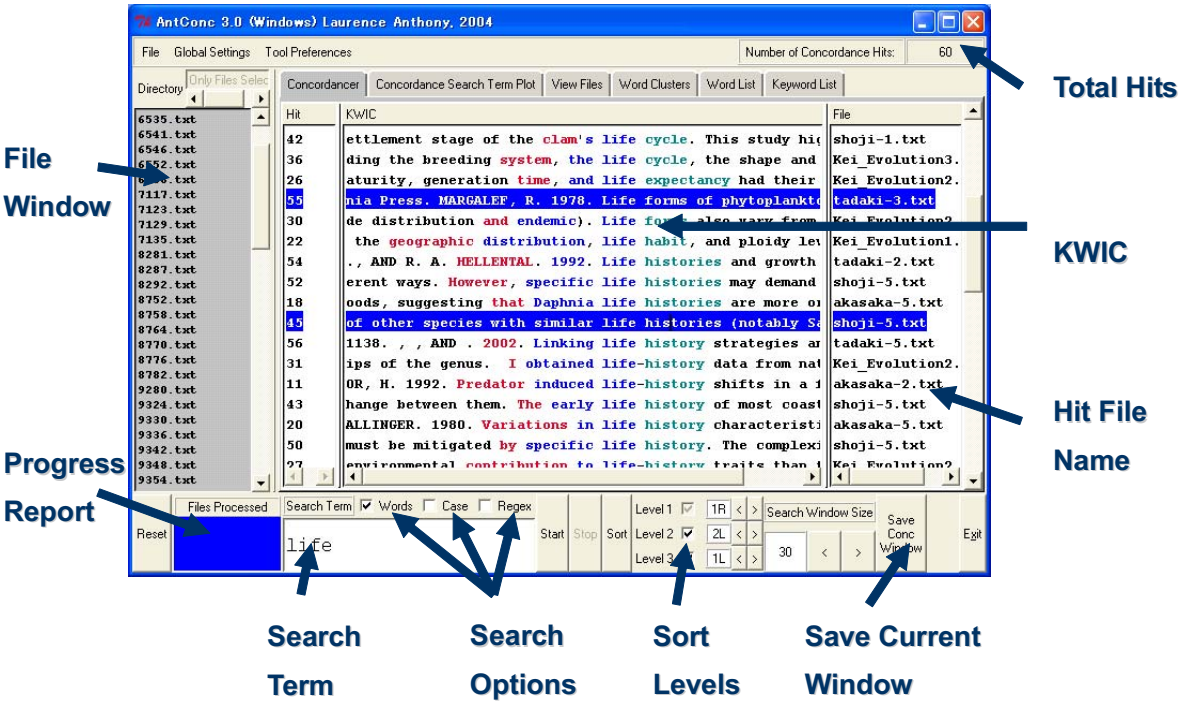

Figure 1. KWIC Concordancer Tool

user is operating the Concordancer Tool. As with all other tools in the program, the Concordancer Tool is designed so that the most common operations are accessible directly on the main screen. Lonfils & VanParys [9] explain that this is an essential feature of good software design as it avoids confusing pull-down menus and additional windows. It can also be seen that the program's interface widgets, such as check buttons, lists, and window adjusters, have the native look and feel of the operating system, and the same functionality as standard software applications. This improves the ease-of-use and intuitive operation of the program as it will match with the learners' habits and expectations.[9]

The Concordancer Tool of *AntConc* has a wide range of features that make it an effective tool not only for learners, but also teachers and researchers. These are summarized as follows:

1. Search terms can be either substrings, words, or phrases, and can be either case sensitive or insensitive. They can be embedded with a wide range of wildcards that the user can assign to any particular character or string of characters via a menu option.

2. Search terms can be defined as full regular expressions (REGEX), offering the user access to extremely powerful and complex searches.

3. Three levels of sorting of KWIC (Key Word in Context) lines are possible, with user definable highlight colors at each level.

4. If a user clicks on any search term in the KWIC results display, the program will automatically open the View Files tool (described later) and show the search term hit embedded in the original file.

5. The KWIC results display is divided into columns, in which the hit number, KWIC line, and file name are shown separately. As in all other tools, each column can be either displayed or hidden, and standard selection methods can be used to save data in the columns or rows to the clipboard or a text file.

4 Concordance Search Term Plot Tool

The main purpose of the Concordancer Tool is to show *how* a search term is used in a target corpus. For users who want to see *where* a search term appears, *AntConc* offers the Concordancer Search Term Plot Tool, shown in Figure 2.

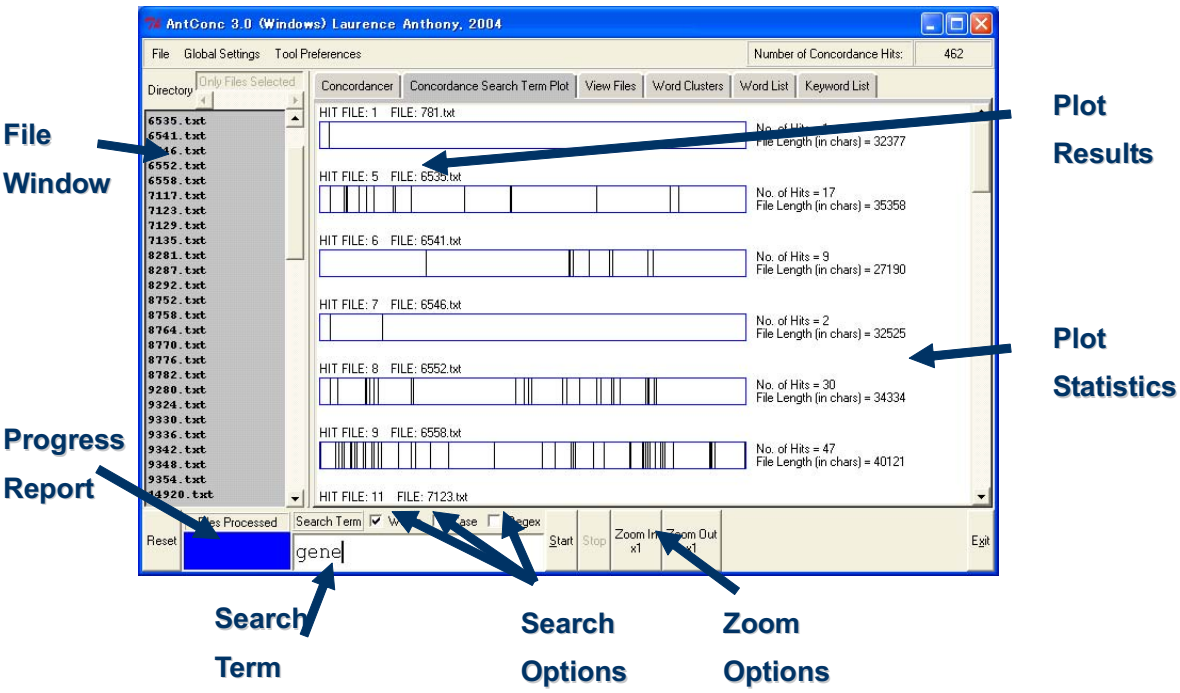

Figure 2. Concordance Search Term Plot Tool

The Concordance Search Term Plot Tool offers the same functionality as the Concordancer Tool in terms of search term options. However, the results are displayed in a quite different way. Here, each box represents a file in which multiple lines represent the relative positions at which search term hits can be found. From this display, it is easy to see where and in what distribution a search term appears in the file. This can be an effective aid, for example, in determining where phrases such as "we" or "in this paper" are used in research articles, or determining which research articles use a particular keyword or phrase.

5 View Files Tool

The View Files Tool of *AntConc* is shown in Figure 3. As described above, when a user clicks on a search term in the results display of the Concordancer Tool, the View Files tool is used to display the search term in the original file. However, the View Files Tool can be used independently to search for any substring, word, phrase or regular expression in a target file, offering the user a very powerful text search engine.

All resulting hits are displayed in a user-definable highlight color, and buttons and keyboard shortcuts can be used to jump to a specified hit anywhere in the file. If the user clicks on one of the highlighted search terms, all KWIC lines based on the term are automatically shown using the Concordancer Tool.

6 Word List / Keyword List Tools

One of the first things that a user will do when analyzing a new corpus is to generate a list of all the words in the corpus. Word lists are useful as they suggest interesting areas for investigation and highlight problem areas in a corpus. Bowker & Pearson [10] describe how word lists can also be used to find families of related word forms and lemmas in a corpus. The Word List Tool is shown in Figure 4.

Hockey [11] states that an ideal word list generation program should be able to sort words into alphabetical or frequency order. The Word List Tool offers these features and the added features of reverse ordering and the ability to count words based on their 'stem' forms. Usually, it is important to use a stop list to avoid counting high frequency function words when generating a word list. In the Word List Tool, this can be done via the preferences window. In addition, users can specify the reverse of a stop list, i.e., a list of only the words that should be counted. These can be specified either by direct input from the keyboard or from a separate file.

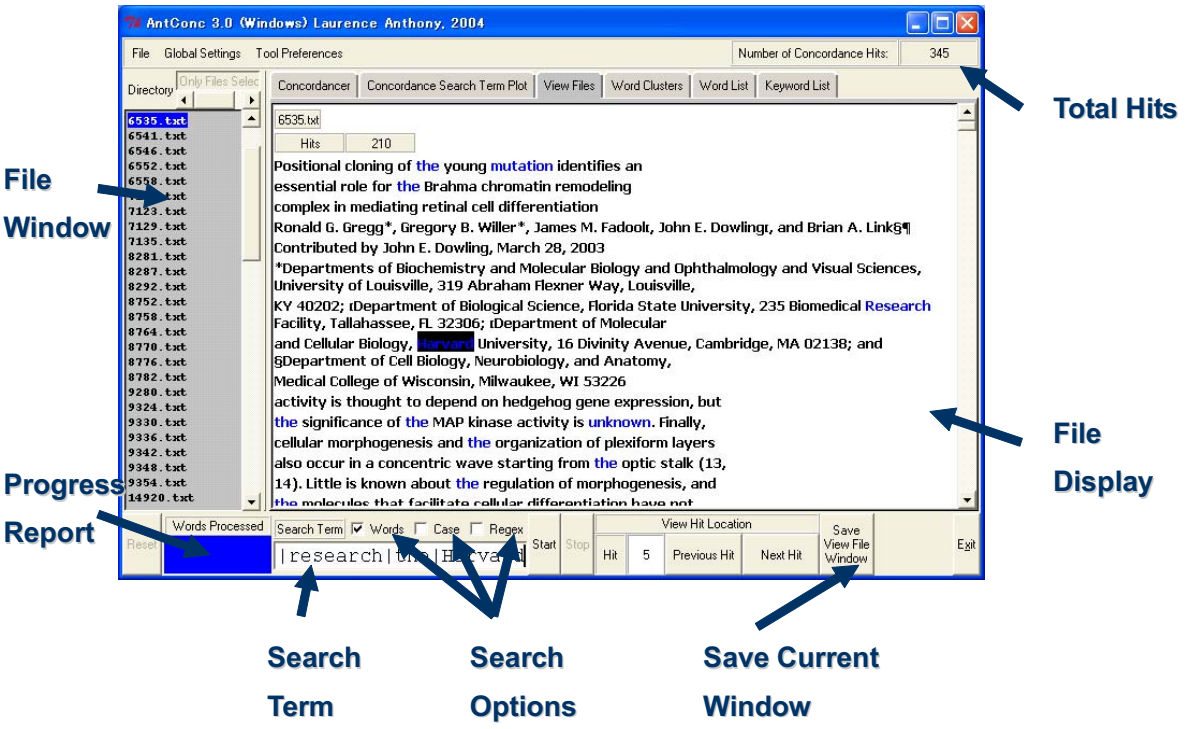

Figure 3. View Files Tool

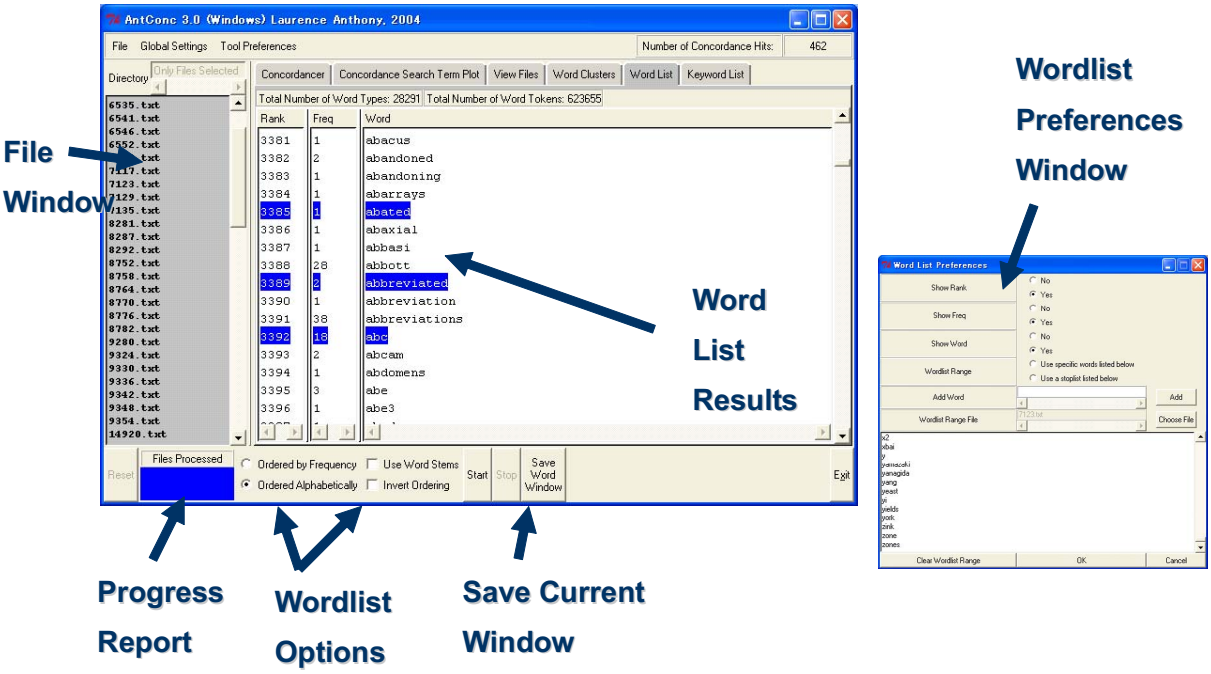

Figure 4. Word List Tool

Experienced users of corpus analysis tools will know that word lists usually tell us little about how important a word is in a corpus. Therefore, *AntConc* offers a Keyword List Tool (Figure 5), which finds which words appear unusually frequently in a corpus compared with the same words in a reference corpus that must also be specified by the user. The Keywords Tool operates in an almost identical way to the KeyWords tool in

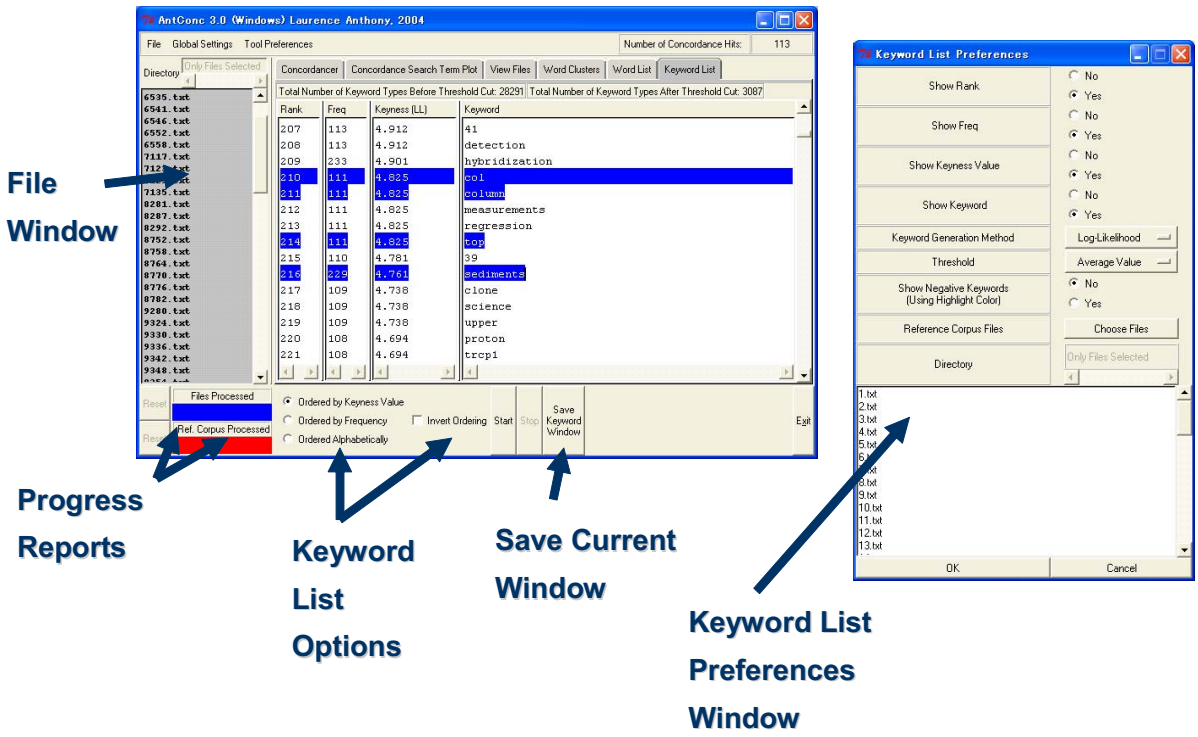

Figure 5. Keyword List Tool

WordSmith Tools, calculating the 'keyness' of words using either the chi-squared or log likelihood statistical measures [12], and offering the user the option of displaying or hiding unusually infrequent keywords (or negative-keywords) in the preferences window.

7 Word Clusters / Bundles Tool

Research has shown that collocations and other multi-word units such as phrasal verbs, and idioms are particularly difficult for learners to acquire.[13] Their importance is even greater if the learner is working with texts in a highly technical or scientific field, as the lexical unit is very often longer than a single word.[10] Surprisingly, collocations and so on have received little attention in most CALL programs [13], perhaps due to the difficulty in identifying and ordering them in a systematic way for the learner.

In *AntConc*, multi-word units can be investigated using the Word Clusters Tool (Figure 6). This tool displays clusters of words centered on a search term and orders them alphabetically or by frequency. The search terms can be specified as a substring, word, phrase or regular expression as in the Concordancer, Plot and View File tools, and the number of additional words to the left and right of the search term can also be specified. It is also possible to set a minimum frequency threshold for the clusters generated.

An alternative way to search for multi-word sequences is to find lexical bundles [14], which are equivalent to n-grams, where n usually varies between two and five words. Few corpus analysis programs offer this feature [1], but *AntConc* includes lexical bundle searches as an option in the Word Clusters Tool. Of course, calculating all the lexical bundles for a particular set of criteria can take a great deal of time. Therefore, as in all other tools in the program, the processing can be halted by clicking on the 'Stop' button at any time.

8 Limitations of *AntConc*

Concordancers can be divided into two main types; 1) those that first build an index which is used for subsequent search operations, and 2) those that act directly on the raw text.[11] The first of these has the advantage that they can operate on large corpora. On the other hand, they tend to be less flexible than the second type, especially if the user is often switching or modifying the target corpus

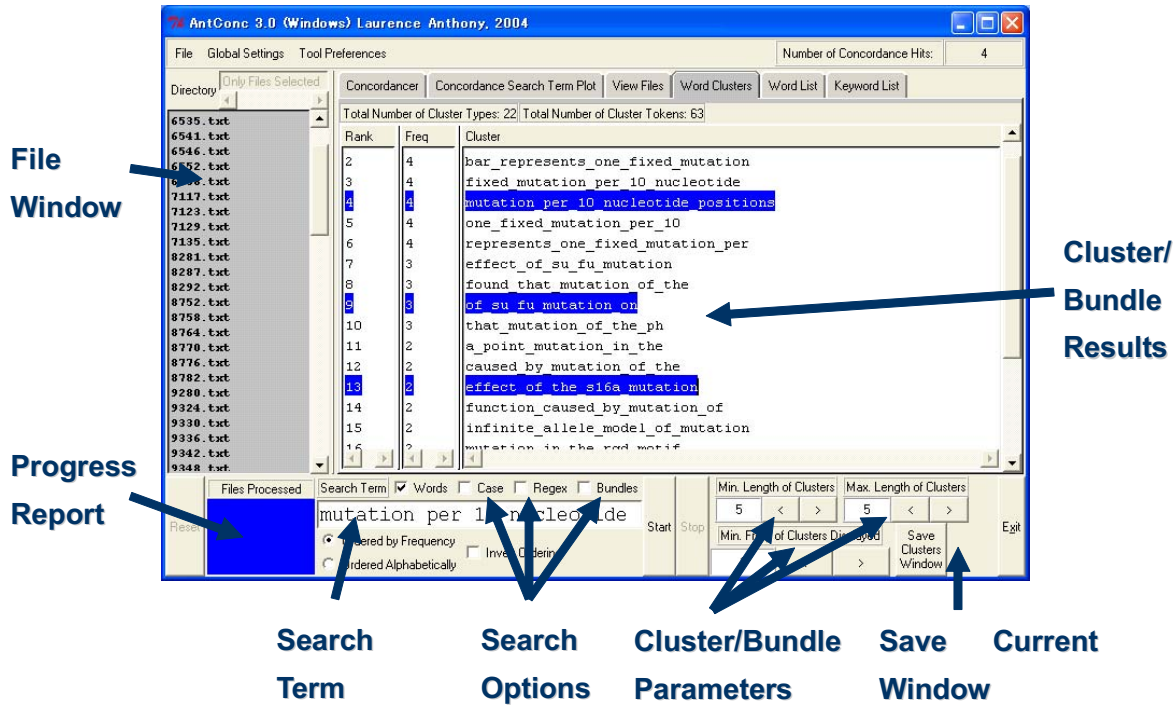

Figure 6. Word Clusters / Bundles Tool

for a particular need. *AntConc* fits into the second category, performing all processing on the raw data

files, and storing results in active memory. For this reason, it is limited to use with small specialized corpora. Nevertheless, as McEnery & Wilson [15] note, one of the major trends in corpus linguistics over the past few years is the increased interest in very small, highly specialized corpora. Small corpora can be used for a great many different purposes, as exemplified by Ghadessy et al. [16], and are particularly effective when teaching technical writing, as demonstrated by Noguchi [17].

Most corpus analysis programs offer users the ability to see the collocates of a search term in a table, where the frequency of the most common words to the left or right of the search term are indicated. Learners often find such tables difficult to interpret and so the current version of *AntConc* offers no implementation of this feature.

Some programs also offer detailed statistics related to the corpus and search results. Again, it was felt that these would overwhelm many learners and so the advice given by Hockey [11] was followed. The program should not include such statistics but instead offer an easy way to copy and paste results into a spreadsheet program for analysis later. As mentioned earlier, the results in all display

windows of *AntConc* can easily be copied and pasted directly into a spreadsheet program using simple keyboard shortcuts.

One of the weakest areas of *AntConc* is in the handling of annotated data such as data encoded in HTML/XML format. Although *AntConc* offers a simple way to view or hide embedded tags used in HTML/XML and other annotation methods, much more sophisticated methods need to be implemented if the full power of annotated data is to be realized.

9 Conclusions and Future Developments

AntConc is a lightweight, simple and easy to use corpus analysis toolkit that has been shown to be extremely effective in the technical writing classroom.[17] Although it does not include all the tools and features of the popular commercial applications, it offers many of the essential tools needed for the analysis of corpora, with the added benefit of an intuitive interface, and a freeware license.

To date, there have been 19 releases of the program since its launch in 2002, including three major upgrades. There are also plans to release a

new version of the software in the near future that addresses some of the limitations described in the previous section. The first improvement will be a redesign of the View Files Tool making it operate with far greater speed. The current tool is able to handle files with ambiguous line endings but this comes with a heavy loss in speed. The next release will also include a tool to view collocates, and the ability to sort word lists alphabetically from both the beginning and end of words, which is a feature recommended by Hockey.[11]

In a later release, it is hoped that *AntConc* will be improved to handle annotated data, in particular XML, in a much more powerful and intuitive way. XML data includes header definitions that if extracted, can be used as part of search criteria. If this extraction can be carried out automatically, it will enable users to access these definitions without any knowledge of the annotation method.

Finally, a detailed user manual and accompanying tutorial video are planned for the software, where the operation of each tool will be explained with concrete examples and a step-by-step guide.

Acknowledgements

This research was supported by a Grant-in-aid for Scientific Research by the Japan Society for the Promotion of Education, Science, Sports and Culture, Japan (No. 16700573), and by a Waseda University Grant for Special Research Projects, Japan (No. 2004B-861).

Notes

1. Information and download instructions available at: http://www.lexically.net/wordsmith/
2. Information and download instructions available at: http://www.monoconc.com/
3. Information and download instructions available at: http://home.ust.hk/~autolang/whatis_WP.htm
4. Information and download instructions available at: p://vlc.polyu.edu.hk/concordance/aboutweb.htm
5. Information and download instructions available at: http://www.antlab.sci.waseda.ac.jp/
6. Information and download instructions available at: http://morphix-nlp.berlios.de/

References

[1] D. Coniam, "Concordancing oneself: Constructing individual textual profiles," *International Journal of Corpus Linguistics*, vol. 9, no. 2, pp. 271–298, 2004.

[2] C. A. Chapelle, *Computer applications in second language acquisition: Foundations for teaching, testing, and research.* Cambridge, England: Cambridge University Press, 2001.

[3] S. Hunston, *Corpora in Applied Linguistics*. Cambridge, England: Cambridge University Press, 2002.

[4] T. Johns, "Contexts: the Background, Development and Trialling of a Concordance-based CALL Program," in *Teaching and Language Corpora*. A. Wichmann, S. Fligelstone, T. McEnery, and G. Knowles. London, England: Longman, 1997, pp. 100-115.

[5] J. M. Swales, *Research Genres*. Cambridge, England: Cambridge University Press, 2004.

[6] Y. C. Sun and L. Y. Wang, "Concordancers in the EFL Classroom: Cognitive Approaches and Collocation Difficulty," *Computer Assisted Language Learning*, vol. 16, no. 1, pp. 83-94, 2003

[7] T. Cobb, "Breadth and depth of lexical acquisition with hands-on concordancing," *Computer Assisted Language Learning*, vol. 12, no. 4, pp. 345-360, 1999.

[8] K. E. Nitsch, "*Structuring decontextualized forms of knowledge,"* Unpublished Ph.D., Vanderbilt University; Nashville, TN, 1978.

[9] C. Lonfils and J. Vanparys, "How to design user-friendly CALL interfaces," *Computer Assisted Language Learning*, vol. 14, no. 5, pp. 405-417, 2001.

[10] L. Bowker, L. and J. Pearson, *Working with Specialized Language: A Practical Guide to Using Corpora*. London, England/New York, NY: Routledge, 2002.

[11] S. Hockey, "Concordance Programs for Corpus Linguistics" in *Corpus Linguistics in North America: Selections from the 1999 Symposium*. R. C. Simpson and J. M. Swales. Ann Arbor, MI: University of Michigan Press, 2001, pp. 76-97.

[12] A. Kilgarriff, "Comparing corpora," *International Journal of Corpus Linguistics*, vol. 6, no. 1, pp. 97-133, 2001.

[13] N. Nesselhauf and C. Tschichold, "Collocations in CALL: An investigation of vocabulary-building software for EFL," *Computer Assisted Language Learning*, vol. 15, no. 3, pp. 251-279, 2002.

[14] D. Biber, S. Johansson, G. Leech, S. Conrad, and E. Finegan, *Longman grammar of spoken and written English*. London, England: Longman, 1999.

[15] T. McEnery and A. Wilson, *Corpus Linguistics. An Introduction. (Second edition*). Edinburgh, Scotland: Edinburgh University Press, 2001.

[16] M. Ghadessy, A. Henry and R. L. Roseberry, *Small Corpus Studies and ELT: theory and practice*. Amsterdam, Holland: John Benjamins, 1996.

[17] J. Noguchi, "A genre analysis and mini-corpora approach to support professional writing by nonnative English speakers," *English Corpus Studies*, vol. 11, pp. 101-110, 2004.

About the Author

Laurence Anthony is Associate Professor in the School of Science and Engineering at Waseda University, Japan, where he teaches technical reading, writing and presentation skills, and is coordinator of the technical English program. He received the M.A. degree in TESL/TEFL, and the Ph.D. in Applied Linguistics from the University of Birmingham, UK, and the B.Sc. degree in mathematical physics from the University of Manchester Institute of Science and Technology (UMIST), UK. His primary research interests are in corpus linguistics, computer assisted language learning (CALL), educational technology, genre analysis, and natural language processing (NLP).