Concordance Line Sorting in *The Prime Machine*

Stephen Jeaco

Xi'an Jiaotong-Liverpool University

The published Version of Record is:
Jeaco, Stephen. 2021. Concordance line sorting in The Prime Machine. *International Journal of Corpus Linguistics* 26:2 pp. 284–297. https://doi.org/10.1075/ijcl.18056.jea

Note: The copy-editing process from this version to the final publication was very helpful and many changes were made to the wording to (a) clarify and (b) trim the word count.

Abstract

The Prime Machine was developed to be a user-friendly English language concordancing tool. This short paper introduces some ways in which users of this corpus tool can sort concordance lines. It considers some possible needs of language learners in terms of the ranking of concordance lines and introduces two concordance line ranking methods which have been adopted and developed for this concordancer.

Key words: *concordance line ranking, concordance line filtering, data-driven learning*

1. Introduction

The Prime Machine (*tPM*) is a corpus tool for English language learning and teaching, and Author (2015, 2017a) reported on the development of Version 1. This corpus tool was designed to provide a user friendly interface for language learners and teachers. As well as being a gateway to a multitude of examples from corpus texts, it provides users with additional information about the context and contextual environment of the words and combinations of words which are being explored, drawing on the theory of Lexical Priming (Hoey, 2005). It was initially designed primarily for English for Academic Purposes (EAP)

students and teachers (pitched at intermediate to advanced levels), but additional features have been developed and added to facilitate corpus linguistic research projects for English majors and students of linguistics at undergraduate and postgraduate level. When the application (or app) is connected to the server, a range of pre-prepared corpora can be accessed and a sample of extended concordance lines along with summary data based on all the hits is downloaded to the user's own computer. The default setting is for this sample to contain up to a maximum of 200 concordance lines for the initial search and there is a button to request further batches of 200 lines. As described in Author (2017b) these concordance data include wider contexts and source information that can be displayed on the app's 'concordance cards'. Tools for importing and exploring the user's own texts as Do It Yourself (DIY) corpora are also provided. In May 2018, *tPM* Version 3 was publicly released[1] and this paper describes some of the concordance line sorting features now implemented.

tPM provides users with access to both online corpora and DIY corpora and in order to understand some background issues related to concordance line sorting for results from these two different kinds of corpora, some fundamentals of the software architecture need to be briefly outlined. The online corpora are prepared and pre-processed by the server's administrator. The databases for these online corpora have undergone a considerable amount of pre-processing, with scripts automatically going through every lexical item in each corpus, extracting concordance lines for these, calculating measures to be used for concordance line ranking and storing these scores for quick and efficient subsequent retrieval by users of the system. Pre-processing online corpora means the lead-in time for adding a new corpus is a matter of hours or days, but this work remains hidden in the sense that it is carried out before users of the software can see the corpus on their menu. After the pre-processing has been completed and when a user enters a query, the server is able to retrieve samples of concordance lines as well as information from these summary tables that provide scores based on all hits for the node in the corpus. DIY corpora in *tPM*, on the other hand, are stored on the user's own computer and only much simpler pre-processing is performed as texts are imported. However, the generation of concordance line scores for DIY corpora does not need any interaction with the server; the user's own computer has access to all the hits for the query, and the user's own computer can take the whole of the processing load for concordance line ranking.

For web-based concordancer interfaces as well as *tPM* as a server-client app, the way concordance results are ranked is very important because when there are a large number of hits, not all the results will be transmitted to the client browser or client app. As well as needing a means to determine how a selection of concordance lines will be made, the ordering of concordance lines can facilitate identification of patterns and/or usage in diverse contexts. *tPM* provides some of the more common ways of sorting concordance lines that are available in other concordancers, including using text order and sorting alphabetically by words in nearby columns. In this paper, however, some considerations of concordance line ranking for language users will be explored and then two of the special methods available in *tPM* will be described. Through effective and useful concordance line ranking, the aim is to provide more ways for language learners and linguistics students to notice and analyse features in the data.

2. Concordance Line Sorting

For researchers using concordancers to identify the range of ways in which a word is used, given the frequency of many words in corpora and the time it takes to analyse and categorize each line, it is usually not possible to examine in detail all the concordance lines for a particular search. Therefore, selection and ordering of a sample of concordance lines needs to be carried out in a systematic way (Sinclair, 1991). A very common way of reducing the number of concordance lines for analysis in concordancing software is to provide random sampling. Sinclair (1991) proposes a sample retrieval–analysis cycle ending when no new patterns emerge. When hits are in the thousands or tens of thousands (or more) a researcher needs to be particularly mindful of the way results have been selected (whether automatically by the system or through user-initiated sampling) and how these are ordered. For language learners, the number of unsorted instances required to provide a good overview for medium to high frequency items is likely to be well beyond their patience or skills, particularly at intermediate levels and beginners. With low frequency words or highly specific search patterns, selection and ordering of concordance lines remains important as this can assist researchers find regularity and notice patterns in the data. However, it becomes an issue much greater importance for sets of results that do not appear on a single screen. For language learners working with results of all sizes, it is especially desirable to sort concordance results to move more “useful” patterns to the top.

Some researchers have looked at methods to order concordance lines so that those containing examples that are easier for language learners to understand appear first. Wible et al. (2002) proposed a method of scoring concordance lines based the percentage of words outside a certain vocabulary profile. However, one of the lessons the Data Driven Learning (DDL) method seeks to instil in learners is that comprehension of everything is not necessary for them to learn something (Johns, 1988). Filtering according to vocabulary frequency using a system like Wible et al. (2002) propose may be appropriate in some contexts, but it is likely to hide collocations and patterns of use that include words which are less frequent in the language as a whole, but are often used or important in more specific contexts or domains. It is also questionable whether lines containing company names should be penalized in the same way as low frequency vocabulary items, particularly if capitalization of initial letters indicates that they are proper nouns.

The most advanced concordance line ranking algorithm currently implemented in any mainstream English concordancer seems to be *Sketch Engine's* GDEX, which is introduced and explained through its application to the extraction of example sentences for collocations by Kilgarriff et. al (2008). Given that the name of the algorithm comes from "Good Dictionary EXamples", it is not surprising that it was initially developed to provide lexicographers with easy access to corpus examples which contain fewer low frequency words, and use a fairly restricted vocabulary, and it seems to have been strongly oriented towards lexicographers as its early target users. With the expansion of corpora from carefully selected and balanced datasets to large automatically harvested collections from the Web, it is obvious that a conflict would arise between displaying the strangeness of internet text and providing the lexicographers with neat examples which can be used and advertised as being "corpus derived", while still upholding expectations of being well-formed. Since dictionary examples usually appear as single sentences with no further context, the penalties for proper names, long sentences and unusual words are easy to understand.

Scoring concordance lines by scores based on vocabulary profiles (Wible et al., 2002) or using a series of requirements (as with GDEX) means that concordance lines can be ranked and lower scoring lines are either relegated to the very end (and perhaps never summoned to the screen) or filtered out completely. However, both these methods prioritise penalties for what are considered undesirable features over ordering according to patterns in concordance line results.

The sorting of concordance lines according to the words in the nearby environment on a very basic level is provided in most concordancers, and while sorting alphabetically by the words in specific slots (L1, R1, etc.) can help users notice common patterns, during the development of *tPM*, an alternative method was sought. This was considered important because unless language learners stay alert to the range of letters of the alphabet visible on each screenful of results, they can easily overlook the fact that their examination may be dominated by As and Bs, for example. A preferred method might be based on repeated patterns of nearby words without needing to prioritize words beginning with letters from the beginning of the alphabet over those beginning with letters from the end.

Very recently, Anthony (2018) introduced some important issues in concordance line sorting (and other corpus data visualizations) and has pointed out that alphabetic ordering depends heavily on the slot (or slots) chosen and has also noted that alphabetic order has little to do with quantitative representativeness. He describes a new system in his *AntConc* software (called 'KWIC patterns') that sorts concordance lines according to the frequency of repeated words specified slots to the left and right of the node. This method is a new addition to the sorting capabilities of *AntConc*, and offers multiple ways for users to sort and resort results according to different slots. However, using raw frequencies means that very high frequency words may dominate the results, and users of the system need to decide on the sort parameters, possibly overlooking patterns in slots that have not been selected. In contrast, the two alternative methods in *tPM* as presented in this paper use similarities between concordance lines and a collocation measure to sort results in a way that displays patterns of use across the 4 word window to the left and right of the node.

2.1. Ranking Concordance Lines Using Links Across Texts

Working with *The Bank of English* during the 1990's, Collier developed a system to rank concordance lines (1994, 1999). His system applied the lexical cohesion measures for text abridgement developed by Hoey (1991) to a set of concordance lines rather than sentences from a single text. In Hoey's text abridgement system, two levels of relationship between each pair of sentences in a complete text are measured. The first level is called a "link", and is established through finding lexical items which are common to both sentences[2]. If a

threshold number of links between a pair of sentences is reached, both sentences in the pair are marked as forming a "bond". The number of bonds each sentence has is the second level of measurement. While acknowledging important differences between sentences from a single text and concordance lines from a whole corpus, Collier worked through the two level system of links and bonds which Hoey developed as a means of measuring the strength of lexical cohesion between two sentences, and with some modifications successfully applied this to concordance lines. Two of the important changes were to restrict the identification of links to repetitions in a specified window of words to the left and right of the node and to allow the user to specify whether repetitions needed to occur in the same slot within the window or not. In order to evaluate this method (and a range of settings for different parameters in the system), he enlisted twelve Cobuild lexicographers to analyse a sample of 200 concordance lines and to identify 20 lines they considered to be most representative of the word's usage and 20 lines they considered to be the most useful as examples for a dictionary. His evaluation showed that different settings will yield different results, but overall the set of lines at the top of the ranking are not very dissimilar to those which would be selected by the dictionary experts (Collier, 1999). He concludes that when humans rate concordance lines for usability over representativeness "the informants are making use of features which are more closely-defined positionally and heavier in grammatical items than those which occur in lines which are chosen as representative" (Collier, 1999, p. 207).

Collier's approach does not seem to have been implemented in any other concordancers available today. However, it provides a way of ranking lines without too many assumptions, and the parameters can allow high ranking for both colligation (through word-order and "function" words) and collocation (based on repeated forms or lemma). Given in his study the system's rankings could more closely approximate dictionary experts rankings for usability over representativeness, it follows that lines which are highly ranked using this system should help guide learners to "notice" patterns in the usage of a node and the method should work more towards providing usage-oriented rather than meaning-oriented examples. In *tPM*, the Links Across Texts score uses this method, and since it counts grammatical items and compares items in fixed positions, it complements the other ranking methods very well.

For the user's DIY corpora, all the concordance lines are available to the app locally, and so ranking scores for a set of concordance lines are calculated on the fly. For small to medium sized corpora stored on the server, results can be generated before the corpora are made available through pre-processing scripts that take less than 24 hours to provide ranking

scores for every individual word in the corpus. These scores are attached to the concordance line data in the database to give "instant" ranking in the user interface. A strength of the system is that it is not based on raw frequencies, but measures the number of matches made with other concordance lines in the set. However, the need to compare each concordance line with each of the other lines leads to a problem of processing speed which increases in orders of magnitude[3], so higher frequency items are not processed in this way in *tPM*. Collier's system is very good, however, for nodes which do not have clear collocation patterns based on statistical measures; it adds more fine-grained ranking to medium frequency data and provides a means of ranking low frequency items for which little collocation information is available.

2.2. Ranking concordance lines using Collocations

Another way of ranking concordance lines is to weigh lines according to the strength and/or number of collocations which they contain. The idea of using collocations as a way into concordance lines analysis is not new. In the paper introducing mutual information collocations, Church and Hanks suggested that they could be "an index to the concordances" (1990, p. 29). *Sketch Engine*'s GDEX algorithm also includes a score for collocates, but the actual weightings used to combine a collocate score with the other measures are not provided in Kilgarriff et al (2008), and it is reported that the most important measures are sentence length and penalties for low frequency items. The concept of collocation is widely accepted as important in English language teaching, and promoting concordance lines which hold examples of strong collocations should help learners. Showing concordance lines which hold examples of strong collocations should provide rich input for learners, and several ways of operationalizing this would seem to be possible: ranking according to the raw number of collocations; ranking according to the total frequencies of all the collocations represented; or ranking according to the statistical strength of the collocations. When a search is made on the online corpora in *tPM*, other summary data based on all hits in the database are returned with the first batch of 200 concordance lines. These other summary data include collocations that are displayed in a word cloud or table on another tab of results – the Collocation Tab. For the Collocations ranking in *tPM*, it was thought desirable to try to make the Collocation Tab and the Lines/Cards Tab mutually supportive. For the collocation clouds on the Collocation Tab, the cube root of the log-likelihood score is used to determine the size of each item[4]. For a

cloud, the difference in size between items needs to be fairly close, otherwise other items in a cloud containing a very strong item would be too small to see. For concordance line ranking, using raw frequency or the raw log-likelihood value would rank a very high frequency or very strong item too highly in comparison with the others and an entire page of concordance lines could be filled with just one collocation. By using the cube root of the log-likelihood, the differences between values are compressed. In purely statistical terms, a fuller evaluation and exploration of alternative ways of obtaining this kind of measure would be needed. However, as a way of ranking results so that collocations strong on the Collocations Tab can be seen in the top results, this pragmatic approach seems reasonable.

2.3. Combining scores for Links Across Texts and Collocations

However, there are some problems with basing ranking only on collocation. Because some words will have very few collocations or none at all, the rankings can be very flat[5]. To work around this, *tPM* combines the results of Collier's system and the collocation rankings, so users can get the benefits of both systems. Collier's system is applied to lexical items with a frequency equal to or lower than 1,000. This cut-off was determined through experimenting with scripts running on the database system in an attempt to balance coverage against overall pre-processing time. Items with a frequency over 1,000 will therefore have a Collier-style ranking of 0, but are likely to have many collocations and therefore values for the other measure are more widely dispersed. Table 1 shows how these measures are combined. As can be seen, the name of the sorting method (used in the app's drop-down menu) matches the first level of sorting. The second level offers a sorting based on the other method, giving the fine-grained sorting for ranks that are tied at the first level. Finally, any results with matching scores for the first and second levels are sorted by the fixed random number. This last level of sorting is simply to ensure that sorting and re-sorting by different methods does not affect the results obtained for each individual method; if a student sorts the lines one way, tries another method and then reverts back, the order of lines will always be the same. Additional ranking options are also available prioritising collocations and links to the left or to the right of the node.

Table 1. How sorting methods are combined in *tPM*

Method	**First level**	**Second level**	**Third level**

Links Across Texts	Method based on Collier (1999)	Collocations (presented here)	Fixed Random Order
Collocations	Collocations (presented here)	Method based on Collier (1999)	Fixed Random Order

To demonstrate the differences between these two approaches, Figure 1 shows how repeated patterns in specific slots contribute to a score for a set of concordance lines using the method based on Collier's system (Links Across Texts). The underlining has been added for the purpose of illustration to show how repeated items in specific slots could be contributing to a link; however, no underlining appears in the KWIC display in the software. Figure 2 shows how collocations contribute to a ranking for the method based on collocations. Again, underlining has been added manually to indicate collocates which would contribute to the ranking score. Figure 3 shows the collocation cloud from the Collocations Tab which would also be downloaded with these concordance data, and it can be seen that many of the prominent collocations are visible in the top few concordance lines from Figure 2. Other collocations in the sub-corpus for the node that are not visible in the cloud include *pilot .. conducted* and *pilot study carried out*. When directly using the software, the number of lines visible on one screen will vary with the size of font and the size and screen resolution, but with 200 lines downloaded at a time, users can scroll down and see other collocation patterns in the concordance line results too.

Figure 1. Top 12 concordance lines for the node *pilot* in the BNC: Academic Sub-corpus, sorted using Links Across Texts.

Figure 2. Top 12 concordance lines for the node *pilot* in the BNC: Academic Sub-corpus, sorted using Collocations.

Figure 3. Log-likelihood collocation cloud for the node *pilot* in the BNC: Academic Sub-corpus.

2.4. Concordance line ranking in tPM Version 3

Before Version 3, each time the user changed the ranking method, a new set of concordance lines would need to be retrieved from the server, with only the top 100 highest ranked lines for the ranking method being available at one time. However, through evaluation and observations of users in my own teaching sessions, it became evident that the range of patterns could be too narrow, and the delay in obtaining results each time students sorted and resorted was a problem. In Version 3, the initial download of concordance lines has been increased to 200 and these always comprise the same fixed random sample[6]. The ranking methods are then applied to this sample, without retrieving additional data from the server. Almost all the sorting methods available for online corpora in *tPM* are also available for user-created DIY corpora. Generating concordance lines for a word in a DIY corpus includes the automatic calculation of Collocations and Links Across Texts ranking scores.

In the author's own institution, students appear to have found it helpful to sort concordance lines using the Collocations method, especially because collocations in the nearby context are also highlighted at the top of each concordance card. Anecdotally, this has been evident through workshop sessions and coursework assignments where individual students try sorting their results in different ways, and often include several different methods in coursework assignments overall, yet typically present concordance lines using collocation ranking.

A summary of the advantages of *tPM*'s Collocations ranking method that have been identified in this paper is as follows:

- Ranking is based on all the hits in the corpus, not just the sample that has been downloaded;

- Strength of collocations in the 4 word window to the left and right of the node are used for the primary ordering of results;
- The ordering is fine-turned according to "links" and "bonds" between each line and the other concordance lines for the node in the entire corpus based on matching words in the 4 word windows either side of the node;
- Through pre-processing the methods are scalable for server delivery, while for DIY corpora the methods can be computed on the fly.

4. Conclusion

This paper has described some of the concordance line sorting features available to users of *The Prime Machine*. It has been argued that sorting concordance lines according to matching items in specific positions and according to collocational measures is likely to be more effective than alphabetical ordering or ranking based on vocabulary profiling. It is hoped that other users of *tPM* will also find these concordance line sorting methods helpful. Further research is needed both to evaluate the methods, particularly considering their effectiveness for different kinds of uses of concordance lines and different kinds of textual analyses. Teachers and researchers in the wider community are encouraged to try out the corpus tool and its ranking methods in different teaching contexts.

Notes

1. For information about *tPM* see www.theprimemachine.net

2. In a text abridgement system these links may also be based on other types of "complex repetition", paraphrase and other relationships (see Hoey, 1991, pp. 51-75).

3. For further details, see Author (2015)

4. Details of the parameters for the log-likelihood collocation measure can be found in Author (2015), Author (2019).

5. See Author (2015) for more details about how this measure works for very high frequency words.

6. The use of a fixed random as opposed to generating new random samples for each request has both advantages and disadvantages. Advantages for language learners are that the results appear more stable or consistent; looking up words a second time will display the same results. The disadvantage, however, is that unless more batches of results are requested, much of the data for higher frequency items may never be displayed as concordance lines. Summary data for the online corpora, however, do draw on all the results (even if the lines are not downloaded). When there are more than 500 hits for results in DIY corpora, options are available to display all the results or to choose a fixed random sample or a new random sample.

References

Anthony, L. (2018). Visualization in Corpus-Based Discourse Studies. In C. Taylor and A. Marchi (Eds.) *Corpus Approaches to Discourse: A Critical Review*. Abingdon: UK. Routledge Press.

Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics, 16*(1), 22-29.

Collier, A. (1994). A system for automating concordance line selection. Paper presented at the NeMLaP Conference, Manchester.

Collier, A. (1999). The Automatic Selection of Concordance Lines. Unpublished Ph.D. dissertation, University of Liverpool.

Frankenberg-Garcia, A. (2014). The use of corpus examples for language comprehension and production. *ReCALL, 26*(2), 128-146. Hoey, M. (1991). *Patterns of Lexis in Text*. Oxford: Oxford University Press.

Hoey, M. (1991). *Patterns of Lexis in Text.* Oxford: Oxford University Press.

Hoey, M. (2005). *Lexical Priming: A New Theory of Words and Language*. London: Routledge.

Jeaco, S. (2015). The Prime Machine: a user-friendly corpus tool for English language teaching and self-tutoring based on the Lexical Priming theory of language. Unpublished Ph.D. dissertation, University of Liverpool. Retrieved from https://livrepository.liverpool.ac.uk/2014579/

Jeaco, S. (2017a). Concordancing Lexical Primings: The rationale and design of a user-friendly corpus tool for English language teaching and self-tutoring based on the Lexical Priming theory of language. In M. Pace-Sigge & K. J. Patterson (Eds.), Lexical Priming: Applications and Advances (pp. 273–296). John Benjamins.

Jeaco, S. (2017b). Helping language learners put concordance data in context: Concordance Cards in The Prime Machine. International Journal of Computer-Assisted Language Learning and Teaching, 7(2), 22–39.

Jeaco, S. (2019). Exploring collocations with The Prime Machine. International Journal of Computer-Assisted Language Learning & Teaching, 9(3), 29–49.

Johns, T. (1988). Whence and whither classroom concordancing? In T. Bongaerts (Ed.), *Computer Applications in Language Learning* (pp. 9-27). Dordrecht: Foris.

Kilgarriff, A., Husak, M., McAdam, K., Rundell, M., & Rychlý, P. (2008). GDEX: Automatically finding good dictionary examples in a corpus. Paper presented at the Euralex, Barcelona.

Li, J., & Schmitt, N. (2009). The acquisition of lexical phrases in academic writing: A longitudinal case study. *Journal of Second Language Writing*, 18(2), 85-102.

Durrant, P., & Schmitt, N. (2010). Adult learners' retention of collocations from exposure. *Second Language Research*, 26(2), 163-188.

Sinclair, J. M. (1991). *Corpus, Concordance, Collocation*. Oxford: Oxford University Press.

Wible, D., Kuo, C.-H., Chien, F.-y., & Wang, C. C. (2002). Toward automating a personalized concordancer for Data-Driven Learning: A lexical difficulty filter for language learners. In B. Kettemann, G. Marko & T. McEnery (Eds.), *Teaching and Learning by Doing Corpus Analysis* (pp. 147-154). Amsterdam: Rodopi.