APHRICA: A PHRase In Context Algorithm

John L. Dawson

Literary & Linguistic Computing Centre, Sidgwick Avenue, Cambridge CB3 9DA, England

KEYWORDS: phrase repetition, literary stylistics

AFFILIATION: LLCC, University of Cambridge, UK

E-MAIL: JLD1@cus.cam.ac.uk
FAX NUMBER: UK + 1223 335062
PHONE NUMBER: UK + 1223 335029

APHRICA: A PHRase In Context Algorithm

This paper was in part inspired by a talk given at the ALLC/ACH92 conference at Oxford by Ian Lancashire. His talk was entitled "Phrasal Repetends in Literary Stylistics" and included the remarks:

> Phrasal repetends also increase in complexity combinatorially as they grow longer. For example, every 5-fixed-word repetend occurring 7 times contains 9 smaller fixed subphrases as well as a larger number of two-word, three-word, and four-word unfixed (collocating) combinations also occurring at least 7 times. (Any of these sub-phrases may occur more than 7 times, depending on the text.) Any phrasal repetend larger than two words, as a result, sits on the top of a pyramid of other phrasal repetends.

He also commented that a program searching any text for all possible collocations of a certain degree of significance appears to be computationally demanding. This is a consequence of the combinatorial effect mentioned above. Certainly, extracting all general repeated collocations from a text would be extraordinarily time-consuming, so the present paper limits itself to the problem of extracting all repeating fixed phrases and sub-phrases, where the words concerned are adjacent.

The program Tact [1], which is widely available, provides some of the facilities for achieving this kind of analysis.

Figure 1 shows part of a key-word-in-context concordance to T. S. Eliot's plays [2, 3], sorted by right-context to show repeating phrases, and the frequencies of phrases and sub-phrases which can be derived from this. It is quite clear that there are several such phrases, and that phrases longer than two words contain sub-phrases which themselves may occur more often than their parent phrases.

The APHRICA Algorithm

Looking at key-word-in-context concordances gives the clue to extracting all possible repeated phrases, and the method is expressed as an algorithm below.

(1) Make a concordance of all words which appear only once in the text. The context of the word is not of interest and may be discarded; what is required is the word and the reference showing the line on which it appears.

(2) Sort the headword/reference pairs back into their original textual order.

(3) Convert the sorted headword/reference pairs into editing commands which replace the singly-occurring words by some non-alphabetic symbol such as <>. A singly-occurring word can obviously not form part of a repeating phrase.

(4) Replace those punctuation marks which are not to be included within repeated phrases by <> as well. Replace multiple <> by a single <>. Extract the sequences of words occurring between <> as single lines. This divides the text into variable-length sections which comprise the longest possible repeating phrases.

(5) Make a keyword-in-context concordance sorted by right-context, but with no left-context.

(6) Because of the way in which the concordance has been sorted, any phrase A B C which appears sorted under the word A will also produce the phrase B C sorted under the word B, and this solves part of the problem of combinatorial effects mentioned at the beginning. Now for any three lines of this concordance, **X**, **Y**, and **Z**, we need a program which for line **Y** outputs

> longest-match (longest-match (**X**,**Y**),
> longest-match (**Z**,**Y**))
>
> working in whole words only, starting from the left. Any output line which then consists only of a single word can be discarded, as in that particular line of the concordance it does not form part of a repeating phrase.

(7) Process the resulting set of repeating phrases as follows: for any phrase which contains more than two words, output the phrase itself and all possible sub-phrases which contain two or more words, again starting from the left. Thus the phrase A B C D would be output together with the sub-phrases A B C and A B.

(8) Make an alphabetic frequency list of the resulting phrases and sub-phrases.

These steps are certainly time-consuming in computational terms, especially the concordance steps (1) and (5), but it is exactly those steps which reduce the processing time in later stages. The

same technique, with suitable modifications, could be used on lemmatized or otherwise normalized texts, or on a grammatical coding of a text.

Using the Repeated Phrases List

One obvious technique to try with the alphabetic frequency list of phrases is to construct a frequency distribution.

Plotting the absolute frequency of n-word phrases is not very informative. We know, by its very nature, that any n-word phrase with frequency f must contain two phrases of length (n-1) words, each with frequency greater than or equal to f, and so on iteratively. In particular, an n-word phrase with frequency f will contain (n-1) two-word phrases with frequency at least f.

A more interesting statistic is the ratio of the frequency of (n-1)-word phrases to that of n-word phrases. If this ratio is very low for all values of n, the implication is that most repeated phrases are contained in longer repeated phrases, but do not occur separately to any great extent. In other words, the author tends to write in relatively long phrases and avoids the standard juxtapositions of function words which are inherent in most texts. Such a text is likely to resemble (or be) poetry, where the constraints of normal sentence construction are considerably relaxed. It seemed to me that a good test of this would be to apply the method to T. S. Eliot's poetry and plays, because the plays themselves are by no means in standard prose, and abound with poetic features. It turns out that the patterns of phrase frequency ratios are indeed sufficiently different to mark the distinction between Eliot's plays and his poetry.

Examples will be shown of the results of applying these techniques to the poetry and prose of Robert Graves and T.S. Eliot.

References

[1] John Bradley, "TACT Design", in T.R. Wooldridge (ed.), *A TACT Exemplar*, CCHWP 1 (Toronto: Centre for Computing and the Humanities), pp. 1–4.

[2] T.S. Eliot, *The Complete Poems and Plays of T.S. Eliot* (London: Faber and Faber, 1969).

[3] J.L. Dawson, P.D. Holland, & D.J. McKitterick (eds), *A Concordance to "The Complete Poems and Plays of T.S. Eliot"* (Ithaca: Cornell University Press, 1995).

Figure 1

Keyword-in-Context Concordance of Phrases	Phrase Counts	
a different		
a different form		
a different kind of pain from prison		
a different landscape		
a different matter	a different	(21)
a different meaning	a different meaning	(2)
a different meaning Or so it seemed	a different person	(3)
a different name	a different sense	(2)
a different occasion	a different way	(2)
a different person		
a different person when you’re talking it		
a different person Whom you must get to know		
a different play		
a different quarter		
a different sense and		
a different sense of		
a different type of person		
a different view		
a different vision		
a different way from me And you are so much older		
a different way That		
...		
a few		
a few came		
a few candles		
a few days ago		
a few days at Wishwood Among his own family	a few	(22)
a few days later Alone at a concert	a few days	(4)
a few days more	a few friends	(2)
a few dying natives	a few minutes	(5)
a few friends	a few minutes ago	(2)
a few friends	a few years	(2)
a few hours ago		
a few minutes		
a few minutes ago		
a few minutes ago I was		
a few minutes alone with you		
a few minutes brooding		
a few moments		
a few questions		
a few weeks later		
a few worth keeping		
a few years abroad		
a few years out of England In one of the Dominions		

Note: This concordance has been abbreviated for the purpose of illustration