Issues of Large-scale Collocational Analysis

Alex Collier

Research & Development Unit for English Studies
University of Birmingham

1. Introduction

This paper will introduce the reader to the basic concepts and procedures which have so far formed part of the analysis of collocational patterns in corpora. It will then go on to identify some of the problems which can arise when these techniques are applied to much larger corpora. Finally, a few ideas on how these difficulties might be overcome will be presented.

2. First Steps towards Collocation

The typical starting point for most corpus users interested in the collocational behaviour of a particular **type** (discrete word) will be to extract all the concordance lines for that *type* from their corpus. Concordance lines show a type in a fixed amount of context, perhaps 80 characters if it is to be displayed on a computer screen. There are now a number of packages available which will extract concordance lines from a corpus in this way, but the display is generally something like the figure below.

Here is a sample of concordance lines for a relatively infrequent word, *kin*. In the context of such a set of lines the word in the middle (*kin*) is generally referred to as the **node word**.

```
 his own name or where to find his own kin. An Enqlish boy - a feringhi. He wa
 kin, and very frequently the kith and kin we  were most happy to have postcar
 she knows me better than do any of my kin, and not to lose courage: she hasn'
 son's wife left her husband and other kin shunned Bo's camp leaving Bo a succ
, place of birth, religion and next of kin of each of the three dead men, repl
You require the consent of the next of kin and how are you going to get that i
ake impossible the feeling of kith and kin crucial to members' derivation of s
ces of supply and to 'our Commonwealth kin". The NF manifesto declared, "we mu
e a narrow view of who is our kith and kin. Religion very properly tends to em
esn't out go so much into the extended kin, but is still very much inside the
```

Figure 1: Concordance lines for *kin*

In order to perform an analysis of the collocational patterns of *kin*, there are a number of steps to go through.

It can be seen from the above concordance lines that the number of words present on each line varies quite a lot depending on the length of the words: even in the small sample of lines shown above the number of words per line ranges from twelve to nineteen. It is therefore necessary to regularise these concordance lines in some way. Much of the research to date in the field of collocational analysis has involved the definition of a limited amount of context around the node word, in this instance, *kin*. This amount of context is generally referred to as the **span**. The span is usually defined in terms of the number of running words (**tokens**) it contains taken from either side of the node word. Thus a ±4 span would consist of four running words, or tokens, from the left of the node word, the node word itself and four tokens from the right of the node word.

If ±4 spans are extracted from the first few lines of the above concordance lines the following is produced:

```
to find his own kin an english boy a
frequently the kith and kin we were most happy
do any of my kin and not to lose
her husband and other kin shunned bo's camp leaving
religion and next of kin of each of the
of the next of kin and how are you
feeling of kith and kin crucial to members' derivation
and to our commonwealth kin the nf manifesto declared
is our kith and kin religion very properly tends
much into the extended kin but is still very
```

Figure 2: ±4 span for *kin*

The definition of a span serves a number of purposes. Firstly, it clearly delimits the scope of any analysis that is performed on the span, since for a given number of concordance lines it is possible to calculate exactly how many running words will be encountered. Knowing how many words will be encountered is often a bonus if one is trying to develop a piece of software which will store the spans, or information about them, as they pass through the software. Secondly, the span removes the artificial boundaries of the concordance line, since language (English, at least) does not consist of 80-character chunks, but rather of running words with a familiar start and finish. By removing extra characters and words in order to arrive at a span of ±4, ±5 etc we are attempting to define the span in terms of naturally-occurring units. Of course, there has been some debate as to what the optimal span size might be, but, in collocational terms, ±4 seems to be a good starting point. Thirdly, and related to the above point, any non-words such as 'wa' (at the end of the first line of the set of concordance lines) and punctuation such as commas, quotes and so on, will generally be removed by the creation of spans, with the result that only whole words free of punctuation are left, which can be more easily matched against each other.

Most importantly, the 'regularisation' of the original concordance lines makes it possible to apply statistical measures to the data more easily, since it is possible to calculate exactly how many tokens are present and thus how many times (given the frequency of each type in your corpus as a whole) a particular type should be present in the set of spans. To take an example: we have extracted the concordances for *kin* and there are 20 of them. Knowing that we are using (±4) spans instead of raw concordances we can say that there are 20 × (4 + 4 + 1) = 180 tokens in our sample. If it is known that the word *kith* occurs 10 times in a corpus of, say, one million words, it ought to be possible to calculate how many times it ought to be present in a sample of known size such as the one we have for the spans of *kin*. This **expected** frequency is roughly equal to

$$\textit{frequency in corpus} \times \frac{\textit{size of sample}}{\textit{size of corpus}}$$

so we could therefore expect the word **kith** to be present

$$10 \times \frac{180}{1000000} = \mathbf{0.00180} \text{ times}$$

By comparing this expected frequency with the actual or **observed** frequency it is possible to calculate the statistical significance of a type being present in the sample spans. To go back to our real example: *kith* has an observed frequency of 4 - many times its expected frequency. A number of statistical measures can be applied in order to quantify this ratio of observed to expected frequency. One such measure, the **Z-score** is built into the Tact† concordancing package, enabling the user to extract concordances and perform collocational analysis from within the same piece of software.

If the above calculation is performed for every type which occurs in the set of spans, it is possible to build up a profile of the **collocates** of the node word. Such a profile can be seen in the next figure.

† See ALLC/ACH Proceedings 1992, p. 45ff for more information. Tact is available from: Centre for Computing in the Humanities, University of Toronto, Robarts Library, 130 St George Street, Toronto, Ont M5S 1A5 Canada. It is also distributed on the ICAME CD-ROM.

```
kin 35842.3
kith 34188
next 53.6963
our 14.3388
their 6.30728
not 4.7851
and 2.3409
of 2.02009
to 1.8058
the 0.598797
```

Here each word is presented along with the ratio between the observed and the expected frequencies. The expected frequency was calculated on the basis of each type's frequency in the Birmingham 20 million word Corpus, from which the sample of concordances was drawn. The list has been sorted by ratio, which can approximate to ranking the types in order of 'unusualness' - the higher the number, the more unlikely that the type is present by chance. To cut down the list, only types with an observed frequency of greater than 2 were included. It is fairly hard to base any statistical measure on frequencies less than this.

If the items on the list are compared with the original concordance lines, it can be seen that even this fairly simple approach to collocational analysis has achieved quite good results, since the collocates listed seem to tally with the recognisable patterns in the concordances. As one might expect, *kin* occurs much more frequently in this sample than in the Corpus as a whole, but this is solely attributable to the fact that it is the node word of the concordances and is therefore present in every line. For the majority of types the node word itself can be suppressed, but it is worth bearing in mind that some types collocate with themselves, for example in a phrase like 'day after day'. *Kith* has acquired an extremely high ratio. This is because it is very rare overall (13 occurrences in the Corpus) **and** it occurs 4 times in only 180 tokens (the size of the spans) thanks to the phrase 'kith and kin'. *Next* scores quite highly because of 'next of kin', which also includes *of*, which, despite its overall frequency (second most frequent word in the Corpus), still receives a score indicating that it is occurring more frequently than one would expect. The other collocates thrown up by this approach are *our* and *their* and this too can be corroborated by reference to the concordances.

3. Moving Toward Large-scale Collocation

The basic approach to collocational analysis outlined above works acceptably well on a small scale, but it does have its limitations. Several factors have combined which make it necessary and possible to increase the sophistication and scope of the process.

Over the past few years, improvements in computer technology, coupled with a drop in hardware prices, have made it possible to handle ever-larger text corpora. Accompanying the increase in scale of the corpora, the operations which can be performed on them have become more and more complex. These changes have, for example, enabled corpus researchers to progress from looking at a simple frequency-order word list of a corpus to examining the collocational patterns of each type in the list. There would seem to be little reason for not progressing still further as the price/performance ratio continues to fall.

Furthermore, corpus material is now more easily obtained than when the first corpora were built. Whereas previously, special arrangements had to be negotiated with a newspaper house to receive data via a telephone line and modem, newspaper output on CD-ROMs is now available 'off-the-shelf', giving anyone with a PC and CD drive access to many millions of words. Many modern books are computer-typeset, greatly facilitating the capture of electronic versions of the text.

4. Problems of Large-scale Collocation

4.1. Scale of Output

On the face of it there would seem to be nothing to prevent the kind of analysis shown above being performed on ever larger corpora. The technology can almost certainly be relied upon to keep pace with the amount of data available. Indeed, there is nothing to stop us continuing to look at collocation in this not-so-time-honoured fashion and the technology will almost certainly improve to fit the available data. But the

big problem will be in dealing with all the information that comes flooding out of this kind of analysis once it is performed on corpora consisting of many millions, dare one say, billions of words. At least two institutions in the UK have pledged to create corpora a power of ten bigger than existing corpora - Oxford University Press with their 'British National Corpus' and HarperCollins' 'Bank of English' - and these are to be 'heterogeneous' corpora; that is, not created solely from one source or text genre, but built with an idea of variety in mind.

In order to prevent the amount of output from the analysis of such large-scale corpora getting out-of-hand, careful pruning of the results will be necessary. This can be achieved by establishing robust statistical routines for the identification of collocational patterns.

When dealing with a dynamic, or a constantly growing, corpus it also becomes important to develop algorithms to prevent old information being re-presented to the user time after time. To take one instance: it is likely that *kith* is going to be a collocate of *kin* for some time yet, so collocate-analysts must have the option of suppressing that particular piece of information, so that only new information is supplied to them. The converse of this is that researchers must be informed of any decrease in significance or even **disappearance** of such items from the most recent additions to their ever-growing corpus.

The previous point highlights another difficulty which can be encountered, especially when one has to deal with data which is regularly updated. If an analysis has been performed of the collocational patterns of a certain type in one corpus, how can that be compared with the patterns of the same type in a different, eg newer, corpus? For the same reasons, how does one go about comparing the collocational patterns of two types in the same corpus?

4.2. Scale of Input

The following issues concern both 'dynamic' corpora and static corpora which are periodically updated.

As the size of a corpus increases, the methods used to access it (extract concordances, retrieve the frequency of a type etc) will become more and more important. It is vital that the basic tools of the corpus-user are kept up-to-date, otherwise the time spent just waiting for a set of concordances to be extracted will become intolerably long. To this end, indexing methodology must keep pace with the size of the data. Indexing helps to speed up tasks such as creating concordances by storing every location of each of the types in a corpus in a list or **index**. Thus, instead of searching through a corpus from the beginning each time concordances for a word are requested, an indexing system can move straight to each location of the word and display it in the required context. This saves much time, since comparing two words is a substantial task in computing terms. Indexing can therefore be thought of as a means of pre-processing a corpus in order to avoid such word-by-word comparisons. Once a corpus has grown to several million words it becomes nearly impossible to access it by any other means than by indexing, since the response is otherwise far too slow, but even indexed systems have their limitations and one must think carefully in advance of creating such a system so that built-in limitations are not allowed to enter into the design. An example of this might be limiting the number of items in the word list (perhaps because it is desirable to store it in a fixed-size space in a computer's memory). In this case, the system will function perfectly for the first few million words, but then suddenly fail when a new chunk of corpus data is added which pushes the type count beyond the capacity of the word list.

Another way to improve the response time of a piece of software is naturally to run it on better hardware. A concordancing package which can only deal with five million words of corpus data on one machine might quite happily process ten million on another, more powerful, machine. This is, however, a rather expensive solution, for whereas the amount of corpus data available and the capacity of computers is constantly growing, (most) individual researchers' budgets are not. The answer to this problem must therefore lie with improved software techniques. To illustrate this, it is necessary to return to the procedure outlined above for the analysis of the *kin* concordances.

In this example we were dealing with ten concordance lines. From each of those ten lines, the computer had to identify and isolate nine words. Each time it did that it had to perform a word matching operation, take steps to exclude punctuation marks and convert all the words to lower case. Imagine now that instead of ten concordance lines there were ten thousand. Multiply **that** number by the number of words in the span and by the number of operations performed on each word in the span.

As the number of concordances grows, so too does the number of individual types that are involved (there were over 100 in the spans from the example lines). Another example will serve to illustrate the scale of the problem. The type *new* occurs just over nineteen thousand times in the Birmingham Corpus. In the ±4 spans extracted from the concordances of *new* there are over 17,000 types. Ignoring those types which occur less than three times still leaves well over 5,000 to look at. For each of those types, the overall frequency of the type in the corpus has to be retrieved from the corpus word list. This is then used to calculate an expected frequency which is subsequently compared with the observed frequency in the original type list of the spans in order to arrive at a ratio value.

5. Solutions

In order to prevent the amount of corpus data and output from the analysis thereof from overwhelming the corpus researcher, ways must be found to systematise and optimise the operations which we wish to carry out on the corpus.

Making improvements in the hardware used has already been mentioned, but I do not wish to dwell on this point. Many researchers only have limited computer resources available; thus, suggesting that they acquire a bigger and better machine serves little purpose. I will therefore concentrate on software and procedures, since these are issues which can be addressed more easily - people may wish to implement their own systems and new software is generally more affordable, and more justifiable, than new hardware. Being able to do the same thing faster is all very well, but being able to do new things is more attractive.

A range of storage and access techniques is available, the choice of which will depend upon how frequently it is intended to perform collocational analysis on the data. Each successive method outlined below makes the retrieval of collocational information easier and faster.

5.1. Indexing

If collocational analysis is only carried out occasionally and a corpus of a million words or less is being used, then a concordancing package which makes a sequential run through the corpus each time concordances are required is probably adequate. One run is required for each type in each corpus, however, which can make the task of comparing collocates a slow process.

If concordances are required more frequently, or if the corpus is rather larger than a million words, pre-indexing should be considered. Concordances are obtained more quickly, but it will still be necessary to extract spans from them.

5.2. Integerisation

If gaining access to the collocational information in the corpus is important, it is possible to transform each token in the corpus into a number which corresponds to a particular item in the word list. The result of this transformation is that all the tokens in the corpus become the same size, which, combined with indexing, makes extracting spans extremely easy. For example, if the index holds the information that an occurrence of *kin* is at a certain location in the corpus, then the start of its left-hand span will be known exactly - it will be four 'tokens' to the left, but now all tokens are the same size, it is possible to go to precisely the correct location in the corpus to read off the left-hand span, the node word and the right-hand span. The overheads of extracting spans from a set of conventional concordances are therefore avoided.

5.3. Collocates Only

If identifying collocates is very important, then the best possible response time could be obtained by creating a databank consisting of **only** the collocational information from the corpus. With such a bank of collocates, it would be very simple to perform comparisons between types or between corpora.

This approach takes the concept of the dynamic corpus to its extreme, since the actual corpus text can be discarded, leaving only the collocational profiles and the information derived from comparing them. The data retrievable from the bank for the word 'kin' would therefore consist of the word and its frequency in the corpus and a list of all the words which collocate with it, each with its frequency of collocation and its frequency in the corpus:

```
kin 10 ->
a [1 430681] an [1 60769] and [8 522116] any [1 24247] are [1 80507]
bo's [1 1] boy [1 3908] but [1 98522] camp [1 1255] commonwealth [1 291]
crucial [1 610] declared [1 701] derivation [1 22] do [1 38774]
each [1 11541] english [1 3733] extended [1 649] feeling [1 3642]
find [1 9480] frequently [1 1053] happy [1 2455] her [1 71421]
his [1 102929] how [1 21727] husband [1 2555] into [1 36029] is [2 171384]
kin [10 62] kith [3 13] leaving [1 1971] lose [1 1466] manifesto [1 190]
members' [1 81] most [1 21679] much [1 21570] my [1 47326] next [2 8277]
nf [1 1001] not [1 92881] of [7 550031] other [1 29892] our [2 23247]
own [1 16806] properly [1 931] religion [2 927] shunned [1 32]
still [1 16626] tends [1 521] the [5 1113344] to [4 492240] very [2 29666]
we [1 66034] were [1 64471] you [1 133472]
```

Knowing the frequency of the node word and its collocates and the size of the corpus, it is then possible to calculate a significance value for each collocate and rank the collocates by that value.

6. Conclusion

In the AVIATOR Project, where we are specifically interested in tracing the changes in a word's collocational patterns, we have chosen to create collocate banks for the data which we need to examine. We maintain two main collocate banks at present, a static one based on the Birmingham Corpus and another, regularly updated, based on output from the Times newspaper. These two banks together represent about fifty million words of corpus data at the time of writing, yet they occupy no more disk space than the corresponding pre-indexed corpora. Used interactively, this storage method returns a list of collocates (with attached frequencies or significance of collocation) almost instantaneously in response to a word entered by the user.

A further facility involves a batch process which enables us to compare the collocational profiles of all types across two banks (for example of 'general' text and of journalism). In addition, the collocate banks are easily updateable, allowing us to make comparisons across time (for example comparing the latest issue of the Times with all previous issues) while simultaneously merging the new collocational information into the existing bank. The output from these comparisons can then be used to monitor the occurrence of new word combinations, where a type has acquired a new collocate, and also cases where an existing type is used in a completely new collocational context, which would suggest a shift in its meaning. These facilities complement the New Words filter (Filter 1 – see the papers by Blackwell and Renouf in this volume) and are employed in Filter 2 (detecting New Word Combinations) and Filter 3 (finding New Uses of Existing Words) as part of the AVIATOR software suite.