International Journal
of
English Studies

IJES

UNIVERSITY OF MURCIA

www.um.es/ijes

KWICgrouper – Designing a tool for corpus-driven concordance analysis

MATTHEW BROOK O'DONNELL
*University of Liverpool**

ABSTRACT

The corpus-driven analysis of concordance data often results in the identification of groups of lines in which repeated patterns around the node item establish membership in a particular function meaning group (Mahlberg 2005). This paper explains the KWICgrouper, a concept designed to support this kind of concordance analysis. Groups are defined by sets of patterns that can be matched against the lines in a concordance. The central elements of the KWICgrouper are described in object-oriented terms and an experimental implementation described.

KEYWORDS: Concordancing, concordance analysis, software design.

**Address for correspondence*: Matthew Brook O'Donnell.School of English, University of Liverpool. Liverpool, L69 7ZR, UK. Tel: 0151 794 2298. E-mail: m.odonnell@liverpool.ac.uk

I. INTRODUCTION

The Key Word in Context (KWIC) concordance has long been a central tool in computerised text analysis. It is implemented in some form or another in virtually every corpus tool. The ability to sort and select lines from the concordance on the basis of positional criteria, for instance according to words to the right or left of the node item, facilitates the identification of lexicogrammatical patterns and the definition of meaning groups. In corpus approaches that identify themselves as 'corpus-driven' (Tognini-Bonelli, 2001) the use of the concordance in this way is of particular importance in the empirical process.[1.] Under this paradigm, comparative frequency lists, including those resulting from the Key Word analysis (Scott, 1997), provide candidates for detailed concordance investigation. Sorting of the KWIC display facilitates the identification of recurrent patterns that can be described in terms of collocation, colligation and semantic preference or association (Sinclair, 2004; Hoey, 2005; Mahlberg, 2005).

For words with a relatively low frequency or phrases with quite distinct patterns[2.] this method works well and is not too labour intensive. However, how can items with corpus frequencies in the hundreds, thousands or tens of thousands be analyzed?[3.] Sinclair (1999) proposes a method where 25 concordance lines are randomly selected and each line accounted for in terms of surface (collocations and colligations) and meaning patterns (see Sinclair, 2003). Next a further 25 lines are selected and analyzed, each line confirming or adjusting the classification groups from the first round. The procedure is repeated, 25 lines at a time, until no new information is yielded from new sets of concordance lines (see also Hunston, 2002: 52-65). The question of how many iterations are sufficient to provide representative coverage is an import one. It is of course always possible that the next 25 lines will contain new or previously under represented patterns. However, the use of random sampling or the 'every nth procedure'[4.] for selecting the concordance lines to be analyzed provides a similar level of confidence that allows news commentary television programmes to 'call' political elections for one candidate when only 10% of the votes have been counted.

Mahlberg (2005) develops the corpus-driven concordance analysis to incorporate a text-linguistic element and introduces the concept of 'local textual functions'. Her study makes a link between frequency, surface patterns and interpretation in the concordance analysis. The method involves assigning functional labels to lines that contain repeated patterns, semantically related lexical items or similar pragmatic functions. Mahlberg (2007) presents a comprehensive application of this method in an analysis of the all 368 occurrences of the term *sustainable development* (SD) in a corpus of newspaper texts from the 2002 archive of *The Guardian*. She explains that the first component of her method identifies:

> ...groups of examples that illustrate different aspects of meaning associated with SD. These meaning or functional groups collect examples whose concordance lines show similarities. The criteria for the groups described in the following section are a combination of repeated surface patterns and similarities in meaning that are not automatically visible through an exact repetition of a sequence of words (Mahlberg, 2007: 198-199).

Examples of the resulting groups include: (1) Conferences (lines include terms such as *World Summit on*, *meeting on*, *talks on* connected to SD), (2) Organisations (Lines in this group have names of organisations such as *Business Action for SD*, *SD Network*, *SD Commission*), (3) Education (lines in this group include words for qualifications such as *certificate*, *MSc.*, *Ph.D.* that can be obtained *in SD*) and (4) Approaches[5.] (Mahlberg, 2007: 200-201). As the quotation above suggests the criteria for inclusion of a concordance line in a group are a combination of formal features, semantic sets and also an analysis of the wider co-text. For example, the majority of lines in the Conferences group have the preposition *on* in L1 position, giving a pattern [*on* + SD].

However, this is insufficient to classify the lines. A preceding noun from a small set of semantically related words makes the pattern more precise: [{*commission*, *talks*, *meeting*} *on* + SD]. However, SD also functions as a pre-modifier of such nouns in phrases such as *a sustainable development conference*. Mahlberg examined all the lines in detail to place them into groups choosing to place each line in just one group although some could fit into two or more groups. She is careful to point out that ‛these categories cannot be not watertight, as the phenomenon they aim to describe is fuzzy, too’ (2007: 199).

In her larger study of general nouns in English, Mahlberg was unable to carry out an exhaustive analysis of the concordance lines and could not, therefore, present quantitative figures for the resulting functional groups. She states:

> As a consequence of the limited amount of data, and the variety of features that the concordances revealed, quantitative information could only play a minor role in the concordance analysis. On the basis of these limitations, the study has to be regarded as an exploratory study, indicating some initial results. The nature of the results and their implications, however, outweigh the quantitative limitations (Mahlberg 2005: 180).

Mahlberg’s corpus-driven method, the applications of it and the concept of local textual functions are important developments for corpus linguistics. They are the main inspiration for the concepts behind the tool outlined in this paper, which I have called KWICgrouper. This developing tool attempts to provide a framework to support the methodological cycle used in Mahlberg’s studies and reduce the effort required.

The following section extracts the elements of the method described above and presents them in a way that could form the foundation of an implementation. These elements are outlined in object-oriented terms. The next section describes a proof-of-concept of implementation of KWICgrouper in Python. Finally, an illustration of the how the tool can be used to compare the patterns connected to the adjective *fresh* in different parts of articles from a newspaper corpus.

II. CONCEPTS

The two components of the approach to concordancing discussed in the previous section are: (1) a concordance generated for an item from a given text or corpus and (2) a set of functional meaning groups defined in terms of patterns found in the lines of the concordance. A concordance is made up of a series of lines centred around the 'keyword', each with a left and right context. A concordance line may contain a number of representations of the line or stretch of text from the original file, including case normalized and display versions and various tokenized versions, depending on whether punctuation is to be considered as a token or not. Consider the sentence (1) below:

> (1) JACK STRAW triggered *fresh* controversy yesterday after calling for 'good neighbours' to tell the police about anti-social thugs and vandals.

Using whitespace to tokenize the string gives 20 words. The keyword is the fourth word, resulting in a structure show in Table 1. Each 'word' is separated by a • character and has a positional prefix indicating its distance from the keyword item.

left	{3}JACK • {2}STRAW • {1}triggered
centre	fresh
right	{1}controversy • {2}yesterday • {3}after • {4}calling • {5}for • {6}'good • {7}neighbours' • {8}to • {9}tell • {10}the • {11}police • {12}about • {13}anti-social • {14}thugs • {15}and • {16}vandals.

Table 1: KWIC line structure for an example concordance line

Issues relating to tokenization and the creation of word frequency lists have been discussed in considerable detail elsewhere (Barnbrook, 1996: 57-63; Manning & Schütze, 1999: 124-130; Scott & Tribble, 2006: 11-22) and will not be repeated here. However, for the purpose of pattern matching across concordance lines a consistent format should be adopted. For instance the first two words (L3 and L2), *JACK STRAW*, might be best transformed into lower case and the punctuation connected with R6, R7 and R16 removed from the representations used for matching.

The second component of the corpus-driven approach to concordance analysis is the meaning groups that bring together lines that have formal similarities. Theoretically, any number of groups could be defined for a particular concordance and these groups may be related in various ways that can be captured in set theoretical terms. Groups are defined by one or more 'patterns' where a pattern is one or more matching expressions, such as, '*over* in R' or '*face* in L1 to L6' meaning that the word *over* occurs in the right hand side context and that *face* occurs in the range of positions from L1 to L6, respectively.

A concrete example of these concepts is given below in section 5. But in summary here, imagine carrying out a concordance on the word *fresh* in a newspaper corpus that produces 500 lines. The resulting set of data would constitute a KWIC concordance made up of 500 lines, each containing various representations of the sentence or specified context around the matching item *fresh*, and also a normalized and tokenized structure with three components, left, centre and right, against which pattern matching can take place. Two possible meaning groups might be: (1) FACE that has a pattern to match the forms *face*, *faced*, *faces* and *facing* anywhere in the left context and (2) CONTROVERSY, which has patterns to match the words *controversy*, *row*, *embarrassment* and *blow* within the first 3 words of the right context.

III. OBJECT-ORIENTED REPRESENTATION

The purpose of this section to translate the concepts outline in the previous section into more formal terms using the concepts of object oriented design.[6.] Figure 1 shows a UML style object diagram for the central four objects. It should be stressed that this represents an initial (and incomplete) analysis of the KWICgrouper concepts and that the encapsulation and connections between the objects will most likely be refined.

1. *kwic*. The central object is the **kwic** object that represents a concordance. Among its attributes are a unique identifier (`id`), a list of files from which the concordance was built (`files`) and a pattern (`nodePattern`) used to select lines.[7.] The object contains a list of **kwicLine** instances (see below) and methods to add, sort, and select lines based on a matching pattern.

2. *kwicLine*. This object represents a concordance line and has a many-to-one relationship with a **kwic** instance. For the purpose of matching, the object has *left*, *right* and *centre* attributes which contain the tokenized words from the context (see Table 1). This object has a number of methods to carry out processes on a concordance line including access (getLine), display (printLine), transformation (highlight, which highlights words and patterns within a line). A crucial method for the KWICgrouper concept is matchLine that tests a concordance line against a pattern or group of patterns to ascertain its membership in a **group**.

3. *group*. The **group** object encapsulates the concept of a function/meaning group outlined in Section 2. A **group** instance belongs to a **kwic** instance (which may have many different **group** instances) and is defined by a list of **pattern** instances. A group instance can be passed to the matchLine method of a kwicLine instance to determine whether the line meets the criteria for inclusion in the group (addMatch method attaches a line to a group instance).

4. *pattern*. The final object represents a **pattern** that can be applied to a concordance line. The attributes of a pattern instance include: *pattern* – the string (or expression) to be matched, *side* – where the pattern should be applied, either to left or right context or to

either, *range* – the word position or range of positions from the central node where the pattern should be applied, type – whether the pattern is a positive or negative match, that is, are we interested in the occurrence or absence of the string specified in *pattern* and *operator* – which determines the boolean relation that should be used to combine the pattern with others in its group.

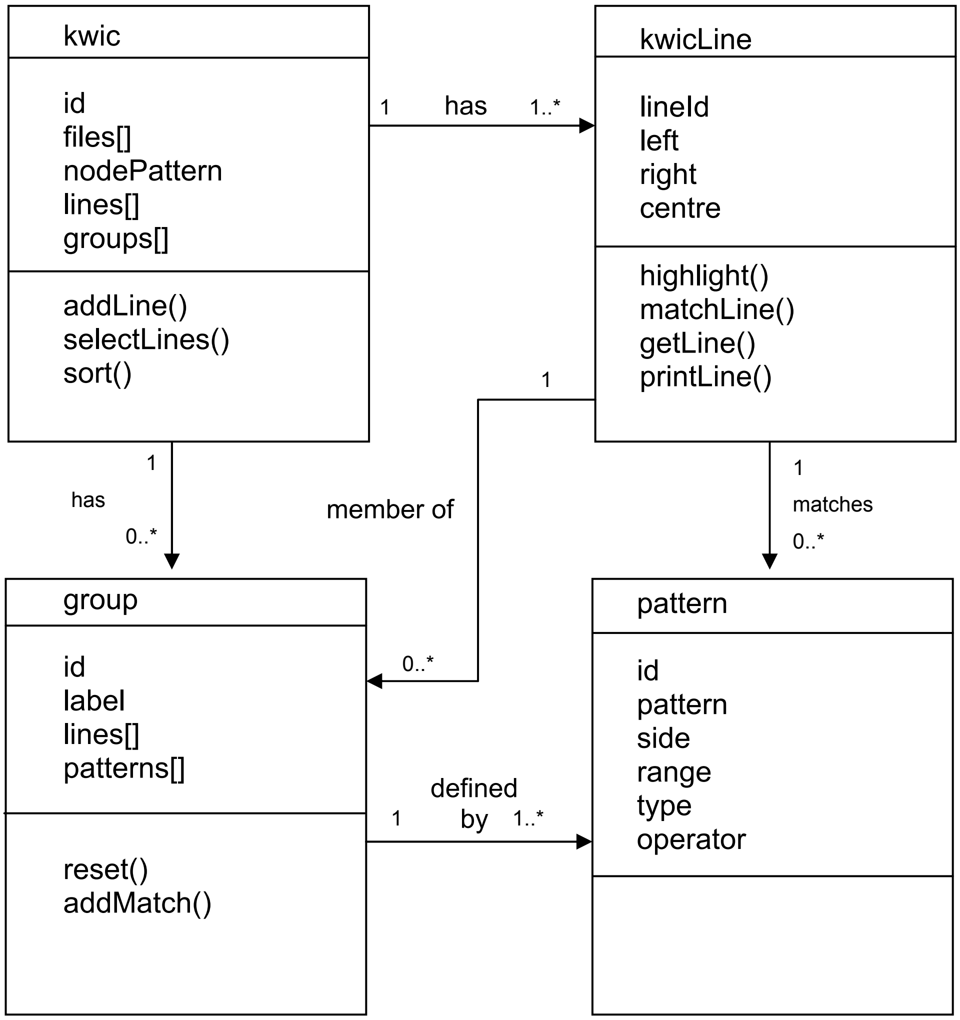


Figure 1: Objects in KWICgrouper

There are a number of areas in this object-based outline where further development is required, particularly in terms of the interaction between the objects and the placement of methods.[8.] However, this outline provides enough structure for an initial implementation.

IV. KWICGROUPER IMPLEMENTATION

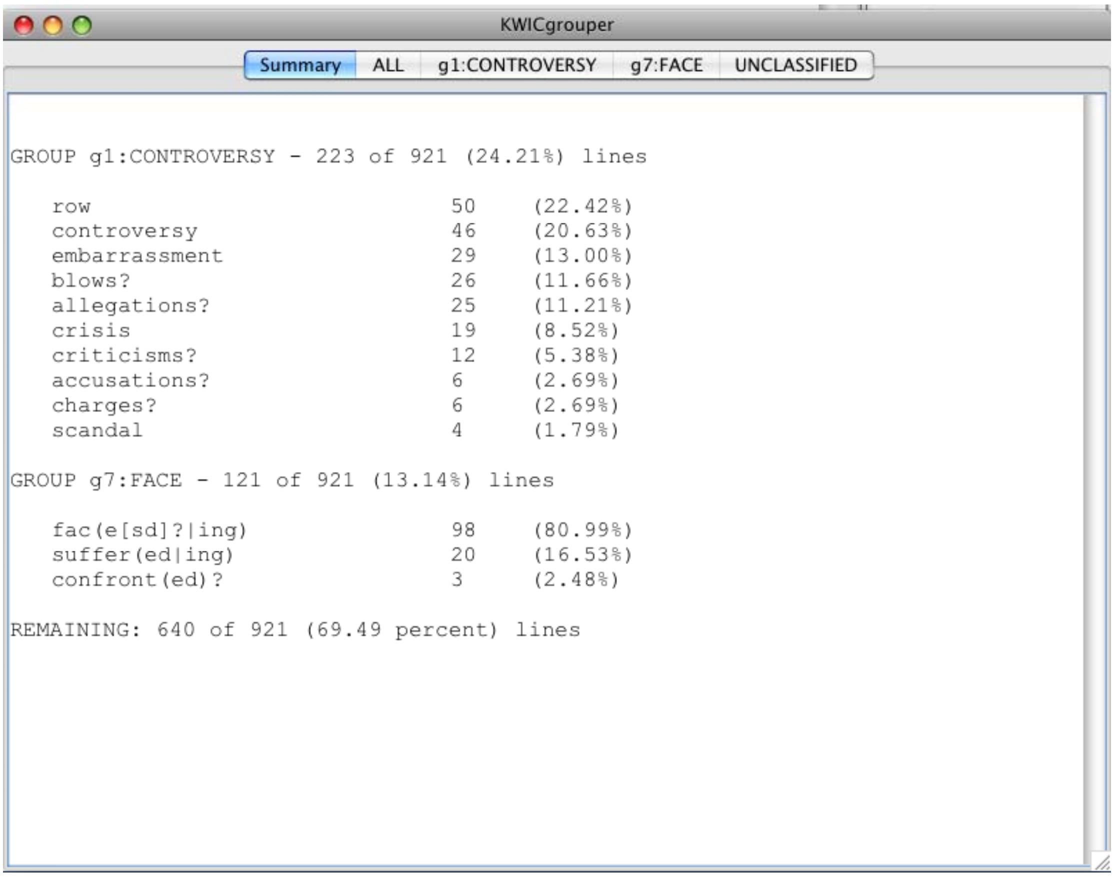
To test the KWICgrouper concept a partial implementation has been carried out in Python. Currently it serves just as a proof-of-concept. Python is well suited to text processing and natural language processing tasks (Mertz, 2003; Bird, Klien & Loper, 2008) with strong string and regular expression processing. It is a highly flexible language that supports object oriented, procedural and functional programming paradigms. The basic structure discussed in the previous section and shown in Figure 1 can be transferred into Python modules in a straightforward manner. The display and user interface could also be developed in a number of ways including a web-based framework or using a GUI toolkit. For the proof-of-concept implementation the wxPython toolkit was used (Rappin & Dunn, 2006). The details of the implementation are of limited interest in the present context. However, one component that warrants explanation is the serialized representation of for the functional meaning groups.

```
<groups kwicTerm=”fresh”>
      <group id="g1">
            <label>CONTROVERSY</label>
            <pattern>
                  <side>R</side>
                  <position start="1" end="4" />
                  <matches>
                        <match>controversy</match>
                        <match>row</match>
                        <match>blows?</match>
                        <match>embarrassment</match>
                        <match>crisis</match>
                        <match>allegations?</match>
                        <match>accusations?</match>
                        <match>charges?</match>
                        <match>criticisms?</match>
                        <match>scandal</match>
                  </matches>
            </pattern>
      </group>
            …
      <group id="g7">
            <label>FACE</label>
            <pattern>
                  <side>L</side>
                  <position start="1" end="4" />
                  <matches>
                        <match>fac(e[ds]?|ing)</match>
                        <match>suffer(ed|ing)</match>
                        <match>confront(ed)?</match>
                  </matches>
            </pattern>
      </group>
      …
</groups>
```

Figure 2: XML representation of groups for fresh concordance

Figure 2 contains a section of the XML used to represent two groups for a concordance created for the word *fresh*. The first group is given the label CONTROVERSY and has one pattern. The pattern applies to the right hand side context in R1 to R4 word positions. There are a series of matches including *controversy*, *row* and *embarrassment.* The current implementation of the matching method makes used of regular expression syntax, so `blows?` and `allegations?` match *blow*, *blows* or *allegation* and *allegations* respectively. In the same way, in the group with the label FACE the match pattern `fac(e[ds]?|ing)` will match *face*, *faces*, *faced* and *facing*. The XML file is read in by the KWICgrouper application and transformed into the matching **group** and **pattern** instances that are then used to match against the concordance lines in **kwic** instance.

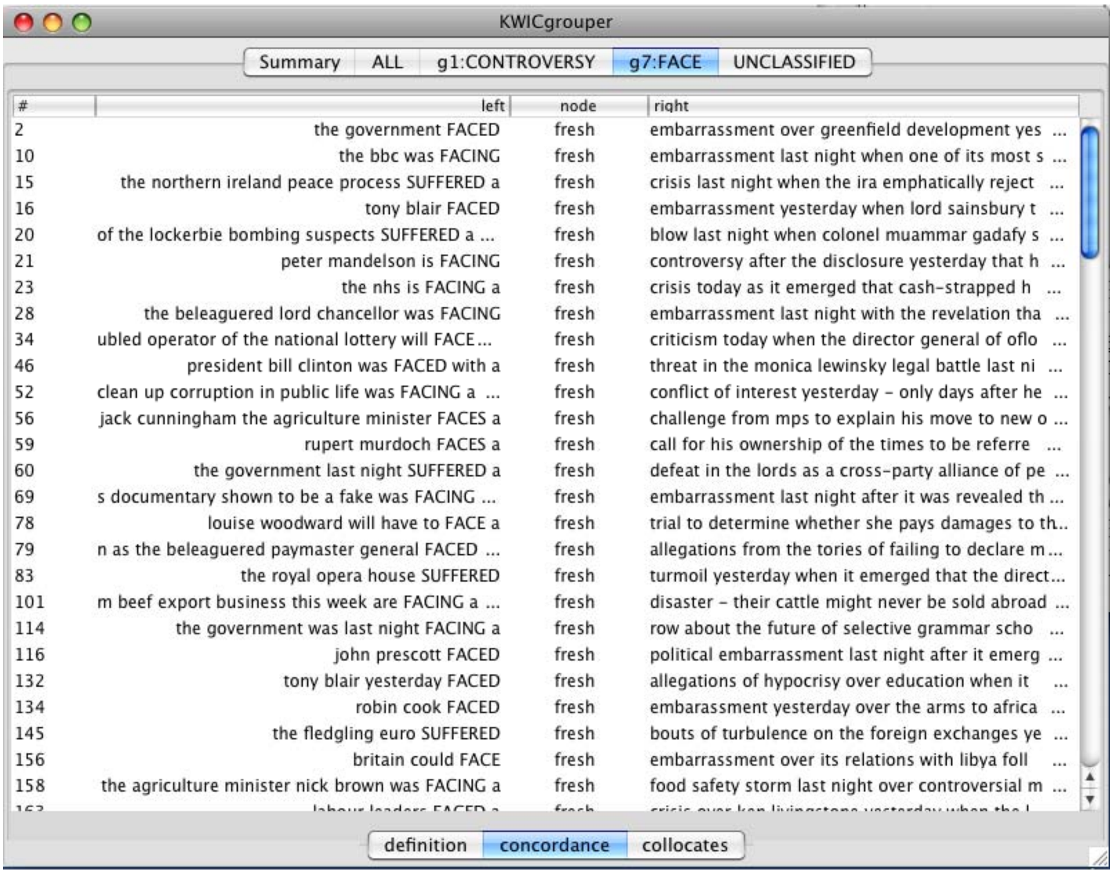
Figure 3 contains a screen shot of the summary screen produced for a concordance of *fresh* in a corpus of first sentences of news articles from the *Guardian* newspaper (see section 5 for more details). The two groups, CONTROVERSY and FACE, defined by the XML in Figure 2 are applied against the 921 concordance lines. The summary indicates that 223 lines (24.21% of the total concordance) have one of the match strings in R1 to R4 position. In each group the number of occurrences of each pattern are displayed, showing that *fresh row* and *fresh embarrassment* account for 96 of the 223 matches.



```
GROUP g1:CONTROVERSY - 223 of 921 (24.21%) lines

   row                          50    (22.42%)
   controversy                  46    (20.63%)
   embarrassment                29    (13.00%)
   blows?                       26    (11.66%)
   allegations?                 25    (11.21%)
   crisis                       19    (8.52%)
   criticisms?                  12    (5.38%)
   accusations?                 6     (2.69%)
   charges?                     6     (2.69%)
   scandal                      4     (1.79%)

GROUP g7:FACE - 121 of 921 (13.14%) lines

   fac(e[sd]?|ing)              98    (80.99%)
   suffer(ed|ing)               20    (16.53%)
   confront(ed)?                3     (2.48%)

REMAINING: 640 of 921 (69.49 percent) lines
```

Figure 3: *Summary screen for KWICgrouper displaying* fresh *concordance*

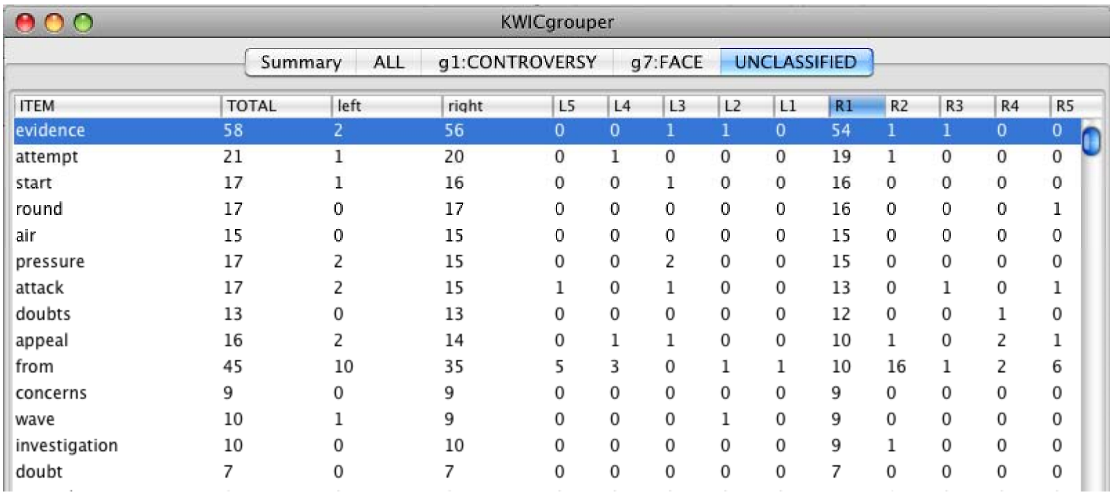
In the current implementation groups are treated as independent of each other and lines can belong to multiple groups. But as mentioned in Section 2 where the KWICgrouper concepts were outlined, it should be possible to order groups so that each line is only matched once or to carry out set-like operations to determine which lines overlap between different groups. The user-interface in the proof-of-concept implementation is somewhat limited. But it is possible to examine the lines that have been classified in each group. Figure 4 shows the lines matched by the FACE group.



#	left	node	right
2	the government FACED	fresh	embarrassment over greenfield development yes ...
10	the bbc was FACING	fresh	embarrassment last night when one of its most s ...
15	the northern ireland peace process SUFFERED a	fresh	crisis last night when the ira emphatically reject ...
16	tony blair FACED	fresh	embarrassment yesterday when lord sainsbury t ...
20	of the lockerbie bombing suspects SUFFERED a ...	fresh	blow last night when colonel muammar gadafy s ...
21	peter mandelson is FACING	fresh	controversy after the disclosure yesterday that h ...
23	the nhs is FACING a	fresh	crisis today as it emerged that cash-strapped h ...
28	the beleaguered lord chancellor was FACING	fresh	embarrassment last night with the revelation tha ...
34	ubled operator of the national lottery will FACE ...	fresh	criticism today when the director general of oflo ...
46	president bill clinton was FACED with a	fresh	threat in the monica lewinsky legal battle last ni ...
52	clean up corruption in public life was FACING a ...	fresh	conflict of interest yesterday – only days after he ...
56	jack cunningham the agriculture minister FACES a	fresh	challenge from mps to explain his move to new o ...
59	rupert murdoch FACES a	fresh	call for his ownership of the times to be referre ...
60	the government last night SUFFERED a	fresh	defeat in the lords as a cross-party alliance of pe ...
69	s documentary shown to be a fake was FACING ...	fresh	embarrassment last night after it was revealed th ...
78	louise woodward will have to FACE a	fresh	trial to determine whether she pays damages to th...
79	n as the beleaguered paymaster general FACED ...	fresh	allegations from the tories of failing to declare m ...
83	the royal opera house SUFFERED	fresh	turmoil yesterday when it emerged that the direct...
101	m beef export business this week are FACING a ...	fresh	disaster – their cattle might never be sold abroad ...
114	the government was last night FACING a	fresh	row about the future of selective grammar scho ...
116	john prescott FACED	fresh	political embarrassment last night after it emerg ...
132	tony blair yesterday FACED	fresh	allegations of hypocrisy over education when it ...
134	robin cook FACED	fresh	embarassment yesterday over the arms to africa ...
145	the fledgling euro SUFFERED	fresh	bouts of turbulence on the foreign exchanges ye ...
156	britain could FACE	fresh	embarrassment over its relations with libya foll ...
158	the agriculture minister nick brown was FACING a	fresh	food safety storm last night over controversial m ...
163	labour leaders FACED a	fresh	crisis over ken livingstone yesterday when the l

Figure 4: XML representation of groups for fresh concordance

Unclassified lines can be viewed, as a collocate table (similar to that created by the Concord program in WordSmith Tools [Scott 2008]). This allows the user to identify other potential groups that can then be added and the matching process repeated. 640 of the 921 lines remain unmatched after the FACE and CONTROVERSY group matching process. Figure 5 shows the top R1 collocates for these 640 unclassified lines. A further two groups DOUBTS (including *doubt(s), fears, concern(s)*) and APPEAL (*appeal(s), call(s), attempt(s), demand(s), impetus, hope(s)*) can be added to the XML (see Figure 6).



ITEM	TOTAL	left	right	L5	L4	L3	L2	L1	R1	R2	R3	R4	R5
evidence	58	2	56	0	0	1	1	0	54	1	1	0	0
attempt	21	1	20	0	1	0	0	0	19	1	0	0	0
start	17	1	16	0	0	1	0	0	16	0	0	0	0
round	17	0	17	0	0	0	0	0	16	0	0	0	1
air	15	0	15	0	0	0	0	0	15	0	0	0	0
pressure	17	2	15	0	0	2	0	0	15	0	0	0	0
attack	17	2	15	1	0	1	0	0	13	0	1	0	1
doubts	13	0	13	0	0	0	0	0	12	0	0	1	0
appeal	16	2	14	0	1	1	0	0	10	1	0	2	1
from	45	10	35	5	3	0	1	1	10	16	1	2	6
concerns	9	0	9	0	0	0	0	0	9	0	0	0	0
wave	10	1	9	0	0	0	1	0	9	0	0	0	0
investigation	10	0	10	0	0	0	0	0	9	1	0	0	0
doubt	7	0	7	0	0	0	0	0	7	0	0	0	0

Figure 5: Some collocates of fresh in the unclassified lines

These two new groups classify a further 107 lines. The cyclic process of refining groups and adding further groups continues until the user is satisfied that the main function meaning groups have been discovered.

```
<group id="g5">
      <label>DOUBTS</label>
      <pattern>
            <side>R</side>
            <position start="1" end="4" />
            <matches>
                  <match>doubts?</match>
                  <match>fears</match>
                  <match>concerns?</match>
            </matches>
      </pattern>
</group>

<group id="g6">
      <label>APPEAL</label>
      <pattern>
            <side>R</side>
            <position start="1" end="4" />
            <matches>
                  <match>appeals?</match>
                  <match>calls?</match>
                  <match>attempts?</match>
                  <match>demands?</match>
                  <match>impetus</match>
                  <match>hopes?</match>
            </matches>
      </pattern>
</group>
```

Figure 6 : XML representation of additional groups for fresh concordance

V. EXAMPLE: FRESH CONTROVERSY IN THE HEADLINES!

Mahlberg and O'Donnell (2008) present an examination of the distinctive collocational and text functional behaviour of the word *fresh* in text-initial sentences (i.e. the first sentence of a news article, which is labelled TISC) when compared to non text-initial sentences (NISC). Looking at collocates in R1 position they found that the words *controversy*, *row*, *embarrassment* and *blow* (that they labelled CONTROVERSY nouns) are distinctive collocates of *fresh* when it occurs in a text-initial sentence. They relate this finding to features associated with 'newsworthiness' and particularly the idea that negative events and issues hold particular appeal within the context of 'hard news' (Bell 1991). The focus on the adjective *fresh* comes from the fact that it occurs with greater relative frequency in text-initial sentences (TISC) than in non-initial sentences (NISC). The Key Word procedure (Scott 1997) was used to identify this and other such items with text-initial associations. Many of the collocates of *fresh* are also by themselves text-initial key words. One such example is *controversy*, which occurs 353 times in TISC and 533 times in NISC (NISC is nearly 7 times larger than TISC). The KWICgrouper can be used to explore the collocates and function meaning groups connected with *controversy* and to investigate whether the word behaves differently in text-initial sentences than in non-initial ones.

<table>
<tr><th colspan="3"><b>TISC</b> (<i>controversy</i> 353 occs)</th><th colspan="3"><b>NISC</b>(<i>controversy</i> 533 occs)</th></tr>
<tr><td colspan="3">GROUP g1:EMBROIL - 52 of 353 (<b>14.73%</b>) lines
[left context]</td><td colspan="3">GROUP g1:EMBROIL - 17 of 533 (3.19%) lines</td></tr>
<tr><td>embroil(s|ed|ing)</td><td>19</td><td>(36.54%)</td><td>embroil(s|ed|ing)</td><td>9</td><td>(52.94%)</td></tr>
<tr><td>plung(e|es|ed|ing)</td><td>16</td><td>(30.77%)</td><td>mir(es|ed)</td><td>5</td><td>(29.41%)</td></tr>
<tr><td>engulf(s|ed|ing)?</td><td>11</td><td>(21.15%)</td><td>engulf(s|ed|ing)?</td><td>1</td><td>(5.88%)</td></tr>
<tr><td>mir(es|ed)</td><td>5</td><td>(9.62%)</td><td>marred</td><td>1</td><td>(5.88%)</td></tr>
<tr><td>marred</td><td>1</td><td>(1.92%)</td><td>plung(e|es|ed|ing)</td><td>1</td><td>(5.88%)</td></tr>
<tr><td colspan="3">GROUP g2:SPARK - 27 of 353 (7.65%) lines
[left context]</td><td colspan="3">GROUP g2:SPARK - 16 of 533 (3.00%) lines</td></tr>
<tr><td>spark(s|ed|ing)?</td><td>10</td><td>(37.04%)</td><td>spark(s|ed|ing)?</td><td>11</td><td>(68.75%)</td></tr>
<tr><td>re-?ignit(e|es|ed|ing)</td><td>10</td><td>(37.04%)</td><td>defus(e|ed)</td><td>2</td><td>(12.50%)</td></tr>
<tr><td>ignit(e|es|ed|ing)</td><td>3</td><td>(11.11%)</td><td>re-?ignit(e|es|ed|ing)</td><td>2</td><td>(12.50%)</td></tr>
<tr><td>defus(e|ed)</td><td>3</td><td>(11.11%)</td><td>ignit(e|es|ed|ing)</td><td>1</td><td>(6.25%)</td></tr>
<tr><td>re-?kindl(e|es|ed|ing)</td><td>1</td><td>(3.70%)</td><td></td><td></td><td></td></tr>
<tr><td colspan="3">GROUP g3:CAUSE - 44 of 353 (12.46%) lines
[left context]</td><td colspan="3">GROUP g3:CAUSE - 74 of 533 (13.88%) lines</td></tr>
<tr><td>fac(es|ed|ing)</td><td>13</td><td>(29.55%)</td><td>caus(e|es|ed|ing)</td><td>47</td><td>(63.51%)</td></tr>
<tr><td>court(s|ed|ing)?</td><td>10</td><td>(22.73%)</td><td>court(s|ed|ing)?</td><td>15</td><td>(20.27%)</td></tr>
<tr><td>caus(e|es|ed|ing)</td><td>8</td><td>(18.18%)</td><td>provok(e|es|ed|ing)</td><td>7</td><td>(9.46%)</td></tr>
<tr><td>provok(e|es|ed|ing)</td><td>7</td><td>(15.91%)</td><td>generat(e|ed|ing)</td><td>3</td><td>(4.05%)</td></tr>
<tr><td>prompt(ing|ed)?</td><td>3</td><td>(6.82%)</td><td>fac(es|ed|ing)</td><td>2</td><td>(2.70%)</td></tr>
<tr><td>generat(e|ed|ing)</td><td>2</td><td>(4.55%)</td><td></td><td></td><td></td></tr>
<tr><td>(seek|sought) out</td><td>1</td><td>(2.27%)</td><td></td><td></td><td></td></tr>
<tr><td colspan="3">GROUP g5:FRESH - 77 of 353 (<b>21.81%</b>) lines
[in L1]</td><td colspan="3">GROUP g5:FRESH - 25 of 533 (4.69%) lines</td></tr>
<tr><td>fresh</td><td>37</td><td>(48.05%)</td><td>political</td><td>12</td><td>(48.00%)</td></tr>
<tr><td>new</td><td>15</td><td>(19.48%)</td><td>further</td><td>5</td><td>(20.00%)</td></tr>
<tr><td>political</td><td>7</td><td>(9.09%)</td><td>running</td><td>3</td><td>(12.00%)</td></tr>
<tr><td>growing</td><td>7</td><td>(9.09%)</td><td>fresh</td><td>3</td><td>(12.00%)</td></tr>
<tr><td>running</td><td>5</td><td>(6.49%)</td><td>new</td><td>1</td><td>(4.00%)</td></tr>
</table>

further	3	(3.90%)	growing	1	(4.00%)
renewed	2	(2.60%)			
repeated	1	(1.30%)			

Table 2: Comparison of some groups for controversy in text-initial and non-initial sentences

Table 2 shows a selection of the groups defined through the cyclic use of KWICgrouper outlined in the previous section. The groups were first defined against the 353 TISC concordance lines for *controversy*, with the results shown in the left hand column of Table 2. Then the same XML pattern definition file was used for running the 533 NISC concordance lines through the program. The reader is invited to examine the differences between the TISC and NISC lines in Table 2. Here attention is drawn to the EMBROIL and FRESH groups.

VI. CONCLUSIONS

The KWIC concordance will continue to be a central tool in corpus linguistic analysis. Concepts such as Mahlberg's Local Textual Functions and her methodology for assigning concordance lines to function meaning groups highlight the analytical power of the concordance. Many of these patterns are (currently) beyond the reach of a fully automated procedure. Not only do human beings possess the semantic knowledge necessary for the identification of such groups, they also have highly tuned pattern recognition abilities. Existing tools for concordance analysis go a certain way to supporting the analysis and grouping of a small to medium set of lines. But it becomes increasingly challenging as items of medium and high frequency are examined. The concepts and analytical procedures discussed in this paper, brought together in the notion of the KWICgrouper, are an attempt to explore how currently concordance tools might be extended and future ones developed to assist corpus-driven analysis.[9.]

NOTES

1. Contrasting ʽText' and ʽCorpus' views, Tognini-Bonelli suggests a difference in reading mode. Text is read horizontally—word to phrase to clause to sentence, and so on—while a ʽcorpus, *examined at first in KWIC format with node word aligned in the centre*, is read vertically, scanning for repeated patterns present in the co-text of the node' (2001: 3; emphasis added).
2. Sinclair's well known examples include *budge*, *naked eye* and *true feelings* (see Sinclair 2004), which can be demonstrated from under 50 concordance lines.
3. Collocation profiles using statistical measures, such as t-score, Mutual Information and log-likelihood (Church & Hanks 1989; Clear 1993; Oakes 1998) provide a useful way to abstract and summarize the most significant patterns within a set of concordance lines, thus in some was automating the task of pattern identification over concordance lines. Recent developments include the notion of the 'word sketch' implemented in the SketchEngine (Kilgarriff et al 2004; http://sketchengine.co.uk), described as 'a one-page, automatic, corpus-derived summary of a word's grammatical and collocational behaviour.'
4. The ʽevery nth procedure' takes into consideration the total number of concordance lines for a node item and the number of sample lines required. So 100 lines from an item with 54,750 hits in a corpus would be collected by extracting lines 1, 548, 1096, 1644, 2192 and so on. Mahlberg (2005: 68) discusses the challenges of analyzing each of the 62 nouns in her study using the procedure Sinclair suggests. She concludes that ʽfor reasons of feasibility, only 100 concordance lines are selected per noun' and that ʽconsequently, the results cannot be taken as a basis for a representative quantitative interpretation'.
5. Mahlberg outlines the criteria for this group: ʽSD occurs as a premodifier of nouns such as test, review, projects, report, criteria, strategy, which can be interpreted as an indication of systematic approaches of dealing with SD, or the complexity of SD that requires a well-organised approach, which also becomes obvious in noun phrases such as principles of sustainable development' (2007: 201).
6. See Mason (2000) on the use of object oriented design for text processing programming.
7. The file processing and matching functionality is currently placed outside of the kwic object, but a case could be made for their incorporation. That is, a kwic object is passed a file or series of files that it uses to create the lines it contains.
8. For instance, the method for matching lines could be placed within the **group** object with each of its **pattern** instances having a match method instead of being located with the **kwic** and **kwicLine** objects as currently shown in Figure 1. In terms of attributes, the current representation of the **pattern** object includes the *type* and *operator* attributes that arguably should be moved to the **group** object along with a more flexible way of combining relations between the various matching patterns of a group.
9. Some of the functionality has been incorporated into the latest version of Concord in WordSmith Tools Version 5.0 through the 'Follow-Up' feature. See http://www.lexically.net/downloads/version5/HTML/index.html?conc_follow_up.htm (accessed 6.6.08).

REFERENCES

Barnbrook, G.. (1996). *Language and computers*. Edinburgh Textbooks in Empirical Linguistics; Edinburgh: Edinburgh University Press.

Bell, A. (1991). *The Language of News Media*. Oxford: Blackwell.

Bird, S., E. Klein, & E. Loper. (2008). Natural Language Processing in Python. Available online at: http://nltk.org/index.php?title=Book&printable=yes (accessed 31.5.08)

Church, K. W. & P. Hanks. (1989). 'Word association norms, mutual information and lexicography' in *Proceedings of the 27th Annual Meeting of ACL*, Vancouver, pp. 76-83.

Clear, J. (1993). *From Firth principles: Computational tools for the study of collocation*. In M. Baker, G. Francis, and E. Tognini-Bonelli (eds.) *Text and technology: In honour of John Sinclair*. Amsterdam: John Benjamins.

Hoey, M.P. (2005). *Lexical Priming: A new theory of words and language*. London: Routledge.

Hunston, S. (2002). *Corpora in applied linguistics*. Cambridge: Cambridge University Press.

Kilgarriff, A., P. Rychly, P. Smrz, D. Tugwell. (2004). 'Sketch Engine,' in *Proceedings of EURALEX 2004*, Lorient, France.

Mahlberg, M. (2005). *English general nouns: A corpus theoretical approach*. Amsterdam: John Benjamins.

Mahlberg, M. (2007). 'Lexical items in discourse: identifying local textual functions of *sustainable development*'. In M.P. Hoey, M. Mahlberg, M. Stubbs and W. Teubert. *Text, Discourse and Corpora*. London: Continuum.

Mahlberg, M. & M.B. O'Donnell. (2008). 'A Fresh View of the Structure of Hard News Stories' in *Proceedings of The 19th European Systemic Functional Linguistics Conference and Workshop*, 23rd - 25th July 2007, Saarbrücken, Germany.

Manning, C.D. & H. Schütze. (1999). *Foundations of statistcal natural language processing*. Cambridge, MA: MIT Press.

Mason, O. (2000). *Programming for Corpus Linguistics: How to do Text Analysis in Java*. Edinburgh: Edinburgh University Press.

Mertz, D. (2003). *Text Processing in Python*. Upper Saddle River, NJ: Pearson Education.

Oakes, M. (1998). *Statistics for corpus linguistics*. Edinburgh Textbooks in Empirical Linguistics; Edinburgh: Edinburgh University Press.

Rappin, N & R. Dunn. (2006). wxPython in Action. Greenwich, CT: Manning Publications.

Scott, M. (1997). 'PC Analysis of Key Words—and Key Key Words'. *System*, Vol. 25, No. 1, pp. 1-13.

Scott, M. (2008). *WordSmith Tools Version 5.0*. Liverpool: Lexical Computing Ltd.

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

Sinclair, J.M. (1999). 'A way with common words'. In H. Hasselgard & S. Oksefjell (eds.). *Out of Corpora. A Study in Honour of Stig Johansson*. Amsterdam: Rodopi, pp. 157-175.

Sinclair, J.M. (2003). *Reading concordances: An introduction*. London: Longman.

Sinclair, J.M. (2004). *Trust the Text*. Language, corpus and discourse. London: Routledge.

Tognini-Bonelli, E. (2001). *Corpus linguistics at work*. Studies in Corpus Linguistics, 6; Amsterdam: John Benjamns.