SCoRE: A Tool for Searching the BNC

Matthew Purver

November 7, 2001

Abstract

This paper describes the design and implementation of SCoRE[1], a new tool for searching the British National Corpus (BNC). SCoRE has been created at King's College London (KCL) to provide functionality required for dialogue research. The paper describes the need for new functionality above that provided by SARA, the search software supplied with the BNC, and gives a description of the new functionality of SCoRE. Instructions for use are given in full, together with a brief description of the implementation. A comparison is given with SARA.

1 Introduction

1.1 The British National Corpus and SARA

The BNC (see Burnard, 2000) is a ∼100 million word SGML-encoded corpus of current British English compiled by the Oxford University Text Archive, of which ∼10M words are transcriptions of spoken language. It is available under licence for academic research, and a copy is now available within the Department of Computer Science at KCL.

SARA, the search software supplied with the BNC, provides many functions including the ability to search for strings within the corpus, view results in context, and browse the corpus. A full specification for SARA is given in (Dodd, 1997). A SARA server has been set up at `bach.dcs.kcl.ac.uk`, and the client software is available in `/dcs/research/nlp/corpora/sara/`.

[1] Originally, "**S**earch a **Co**rpus for **R**egular **E**xpressions".

1.2 Dialogue Research

Of particular interest to current research at KCL is a sub-corpus of spoken dialogue. This corpus is being used within the SHARDS[2] and ROSSINI[3] projects to obtain data concerning various elliptical and clarificational phenomena in dialogue. These phenomena are difficult to detect using straightforward string search methods, as they tend not to be characterized by occurrences of particular strings, but rather by particular patterns such as repeats at varying separations.

2 Design and Implementation

2.1 Requirements

A set of requirements for a dialogue search tool were defined as follows:

- Ability to search for word strings (including punctuation)
- Ability to search for repeated strings of user-specifiable length and separation
- Ability to search on a part-of-speech (PoS) basis
- Ability to search on the basis of sentences or speaker turns
- Ability to restrict searches/results to files of particular types (e.g. written/spoken)
- Ability to view matches in context of user-specifiable depth
- Ability to browse dialogue files via a user-friendly interface
- Compatibility with Linux and Windows

SARA provides only the first and fifth of these requirements (PoS searches are available, but only as refinements of prior word searches).

2.2 Implementation

2.2.1 Overview

The BNC is very large (~2Gb), necessitating a client-server design, with the corpus files installed on a server which performs searches and returns results to a client. In order to ease client compatibility with more than one operating system and provide a user-friendly interface, a web-based approach was taken.

[2]SHARDS, a **S**emantically-Driven **HPSG** **A**pproach to the **R**esolution of **D**ialogue Fragment**s**, is a project investigating the resolution of elliptical expressions in dialogue.

[3]ROSSINI is a project investigating the **R**ole **O**f **S**urface **S**tructural **IN**formation **I**n Dialogue.

The server consists of a set of Perl scripts, which perform the main search functions, together with CGI scripts (also written in Perl) which provide the web-server interface - it can therefore be installed on any operating system which has an implementation of Perl and a web server. The client interface is a collection of HTML forms which call the CGI scripts, so can be run by any table-capable web browser.

To allow maximum flexibility, all searching is performed by constructing a Perl regular expression which matches the desired phenomena. Use of variables, together with expressions which match the SGML markup of the BNC, allows such phenomena as repeats across sentence/speaker turn boundaries to be matched.

The search process is performed in two stages. The first stage, search, constructs the regular expression, then examines each file in the corpus in turn, checking it conforms to the required file type and counting the number of matches. This stage produces a list of results which doubles as an interface to the second stage. This stage, display, examines a specific file, converting it to a user-legible format and displaying any matches, together with surrounding context.

2.2.2 Search

Perl-style pseudocode for the search process is shown in figure 1. The regular expression is built using knowledge of the BNC markup scheme (see Burnard, 2000, for details). To allow matching across many sentences/-turns, a FIFO string buffer is used to hold the relevant number of sentences. As each new sentence is added, the oldest one is discarded before matching is attempted.

The process is slow as each corpus file must be examined (at least to the extent of examining the header). Due to the unlimited length of possible search strings and the inclusion of variables as valid search terms, it is difficult to avoid this (for instance, by pre-compilation of a database of words indexed to files, as used by SARA), but possible approaches are being investigated.

2.2.3 Display

The display process follows a similar procedure to the search process, operating on just one file. As well as searching for matches, the raw SGML from the BNC file is converted into a more legible HTML format.

To allow backward context to be displayed, a FIFO string buffer is used to hold the relevant number of most recent sentence/turns, and displayed when a match is found.

```
# initialise
getCommandLineSwitches( $fileType, $sentType, $matchString );
$regExp = compileRegExp( $sentType, $matchString );
$numSents = numSents( $sentType, $matchString );
@files = ls( $BNC_corpus_directory );
startHTMLForm;

# loop through files
foreach $file ( @files ) {
   unless directory( $file ) {

      # check file
      open( $file );
      initialiseBuffer( $buffer );
      if ( checkFileHeaderType( $fileType ) ) {

         # start buffer
         for ( $i = 0; $i < $numSents; $i++ ) {
            push( $buffer, getSentence( $sentType ) );
         }

         # loop through file, adding new sentence to buffer
         while ( push( $buffer, getSentence( $sentType ) ) ) {
            # test match
            if match( $string, $regExp ) {
               $matches++;
            }
            # and lose oldest sentence from buffer
            shift( $buffer );
         }

         # print out & store results for this file
         if ( $matches ) {
            printHTMLFormEntry;
            $res += $matches;
         }

      }
      close( $file );

   }
}

# tidy up
printResultsSummary( $res );
printHTMLSearchOptionsTable;
endHTMLForm;
```

Figure 1: Search Pseudocode

3 User Guide

SCoRE is used via a web browser. The address is `http://bach.dcs.kcl.ac.uk`. The user can choose to search, browse or review previously saved search results. These options are described in sections 3.1, 3.3 and 3.4.

3.1 Searching

The search interface is shown in figure 2. Searching for a simple string of words is very easy: the search string must be entered and the "Submit Query" button pressed. More complex searches are specified by using special characters and variables in the search string: the allowable syntax is specified in full in this section, together with all other options.

Once a search is complete, a table of summarised results is displayed. Section 3.2 explains how to view detailed results.

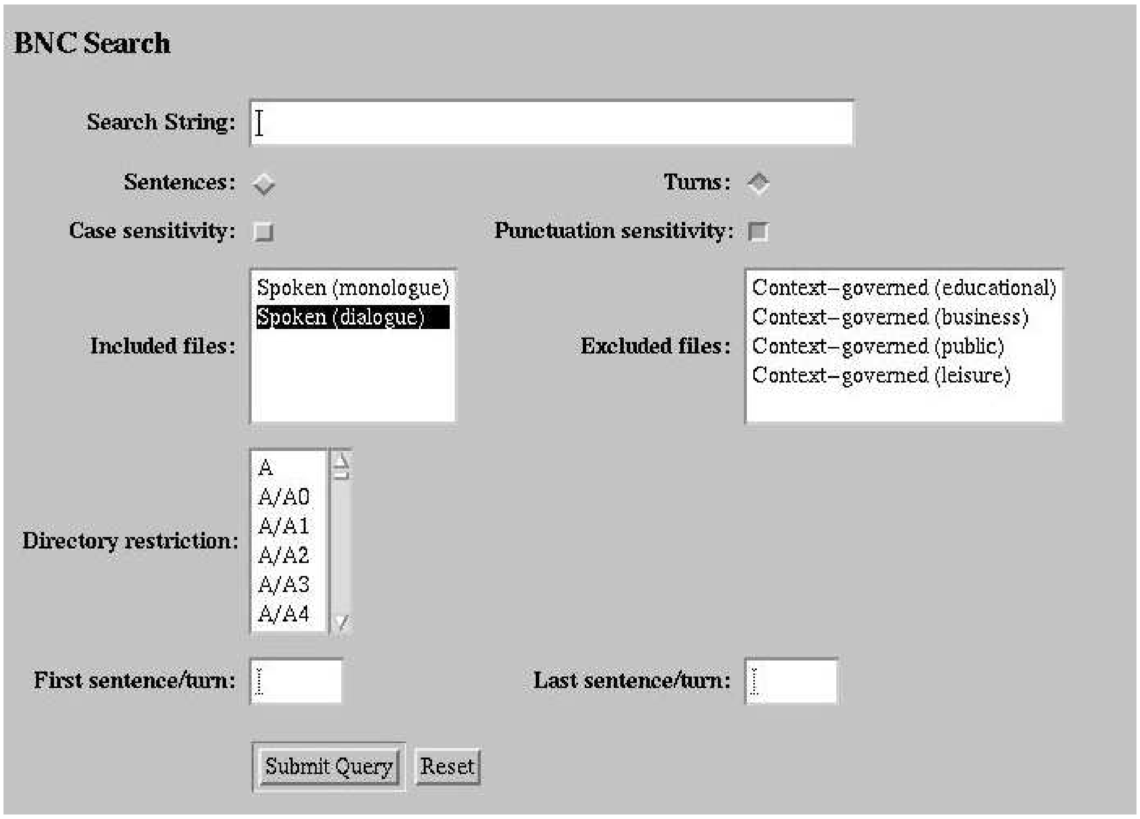

Figure 2: The search interface

3.1.1 Search String

The search string must be typed into this edit field. The string can be any list of valid terms, with each pair of terms separated either by whitespace or a comma & whitespace (see below).

Words

The simplest valid term is a word (any group of alphanumeric characters). A simple search string made up of words will match any sentence/turn which contains these words, adjacent to each other, in the same order.

Word Adjacency

Terms required to be adjacent should be separated by a space, terms that can be non-adjacent can be separated by a comma and space. For example, a dialogue containing:

"**... he usually likes ...**"

would be matched by the search string '`he, likes`', but not by '`he likes`'.

Wild Cards

Terms may contain wildcards. These must be specified as "DOS-style" wildcards rather than in Unix regular expression syntax:

'`*`' matches zero or more alphanumeric characters,

'`+`' matches one or more,

'`?`' matches exactly one.

" **... usually ...**" will be matched by '`usual*`' or '`usual+`', but not '`usual?`'.

Punctuation

Punctuation marks are valid terms. Like all terms, they must be separated by spaces from other parts of the string. This is partly due to the BNC tagging scheme, but also to distinguish between the use of '`?`' as a wildcard and as a punctuation mark - '`?`' on its own as a term will be interpreted as a punctuation mark. For example, a search for sentences containing "**why?**" requires the search string:

'`why ?`', rather than

'`why?`' (which will match four-letter words beginning with "**why**").

Parts-of-Speech

Putting a term in `<>` brackets allows it to match any word in a given PoS category (rather than a specific word). Available PoS codes are shown in table 1 (the BNC PoS tagging scheme is described in detail in (Burnard, 2000)). Wildcards can be used within the tag specification:

'`<AVQ>`' would match any wh-adverb (**when**, **how**, **why**)

'`<??Q>`' would match any wh-question word.

Optional Characters

Optional terms, or optional characters within a term, can be specified using square brackets:

'`hello [there]`' matches both "**... hello ...**" and "**... hello there ...**".

'`[good]bye`' matches both "**... goodbye ...**" and "**... bye ...**".

Sentence/Turn Beginnings/Ends

'`^`' matches the beginning of a sentence/turn,

'`$`' matches the end (including an optional punctuation mark).

Again, these characters must be separated by spaces from other search terms:

'`^ what, ?  $`'

would match any sentence/turn beginning with "**what**" and ending with a "**?**".

Variables

'`\n`' (where `n` is an integer) matches whatever matched the **n**th term in the string. For example:

'`^ wh* and wh*`' could match sentences "**who and what**", "**where and why**" etc., whereas:

'`^ wh* and \2`' could only match "**who and who**", "**where and where**" and so on.

All terms count here, so in the search string:

'`^ wh* \1 \2`'

'`\1`' would match the '`^`' (not particularly useful), while

'`\2`' would match whichever word the '`wh*`' matched (more useful).

Sentence/Turn Breaks

'`|`' (the pipe character) matches the break between sentences/turns. Together with '`\n`' variables, this allows repeats to be searched for:

'`* | ^ \1 ?  $`'

would match single-word questions which are repeats of a word in the previous sentence/turn (such as clarification ellipsis). Beware, though - searches like this (where the first term can be matched by anything) will be slow.

Repeat depth can be specified by varying the number of breaks required:

'`* | | | ^ \1 ?  $`'

would match single-word questions which are repeats of a word from 3 sentence/turns before.

3.1.2 Sentences/Turns

This radio button determines whether the scope of a simple search string is a single sentence or a whole speaker turn (there may be more than one sentence within a turn). It also determines whether '`|`' breaks (see above) match the end of individual sentences or speaker turns.

3.1.3 Case/Punctuation Sensitivity

Case Sensitivity

If this checkbox is checked, only case-sensitive matches are allowed - be aware that this may prevent your search string from matching in sentence-initial positions. It is unchecked by default.

Punctuation Sensitivity

If this checkbox is checked, punctuation is not allowed to separate words required to be adjacent (i.e. those not followed by '`,`' in the search string). It is checked by default.

3.1.4 Included/Excluded Files

Included Files

The search will ONLY consider those files belonging to the categories selected in this selection box. The categories are broad descriptions: "monologue" and "dialogue" (non-spoken texts are currently unavailable but will be added in later versions). The default is "dialogue" only.

Excluded Files

The search will NOT consider any files belonging to the sub-categories selected here. These sub-categories define contextual settings for spoken files (e.g. "business"). None are selected by default.

3.1.5 Directory Restriction

Selecting an entry in this box restricts the search to a particular subdirectory of the BNC - useful when testing out a new search string, as full searches take so long. By default, no entry is selected (no restriction).

3.1.6 First/Last Sentence/Turn

Any (integer) entries in these edit boxes restrict the search to a particular section of each file. This can be used to select a small sub-corpus while still maintaining a broad range of contexts. They are blank by default (no restriction).

3.2 Displaying Results

Once a search is complete a table is displayed showing, for each file containing successful matches: the number of matches, some header information about the file, and two display buttons ("Whole file" and "Matches only"). An example is shown in figure 3. This table of results can be saved for future reference (see below).

Pressing "Whole file" displays the entire dialogue file (this can be slow as dialogue files can be large). Pressing "Matches only" displays only the sections matching the search string, with some forward and backward context as specified (see below).

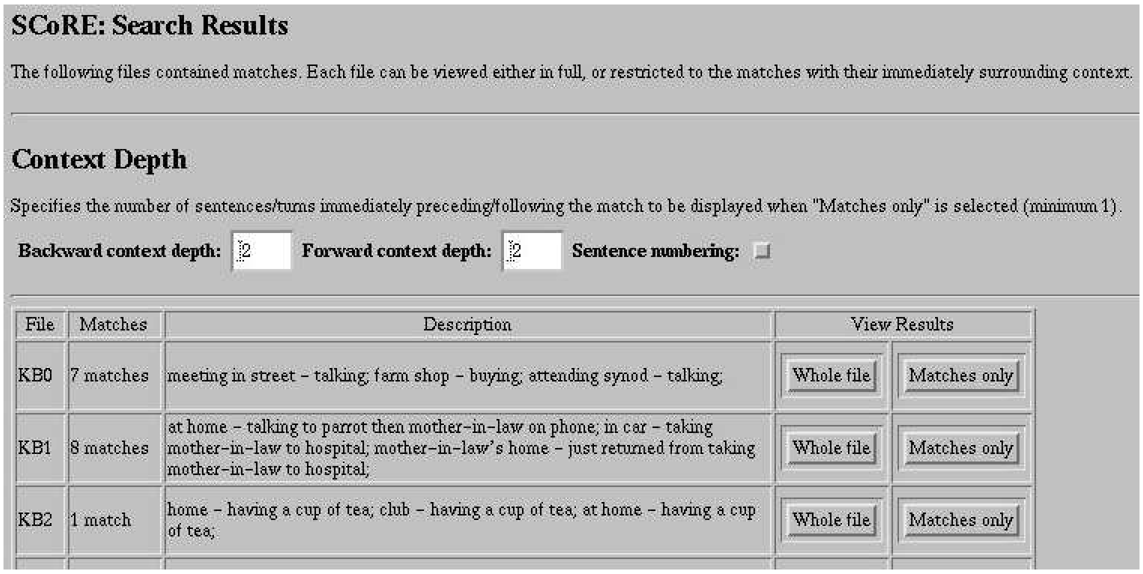

Figure 3: Typical search results

The file is displayed in a colour-coded HTML table format (a typical "Matches only" display is shown in figure 4). Sentences are displayed on separate rows, with speaker turns (possibly containing more than one sentence) marked with the (colour-coded) name of the speaker - unless the name is not available from the file, in which case a unique number and colour is still given to each speaker. The portion of text matching the search string is shown bold and underlined.

Dialogue markup (non-speech events such as tone of voice, laughing, external noises) are displayed in italics and within <> brackets. Truncated words are displayed struck-out. Overlapping portions of speech (produced by different speakers) are highlighted in colour (normal speech is in black).

3.2.1 Forward/Backward Context

The values specified in these edit boxes give the number of sentence/turns that will be displayed before and after the matching sentence/turn(s) (the choice of sentences or turns is controlled by the sentence/turn radio button in the Search Parameter section, as before). Increasing the number slows the display process, but gives more dialogue context. The default for both is 2.

3.2.2 Sentence Numbering

If this checkbox is checked, sentence numbers are displayed alongside the text. This can be helpful as it gives information about the distance between

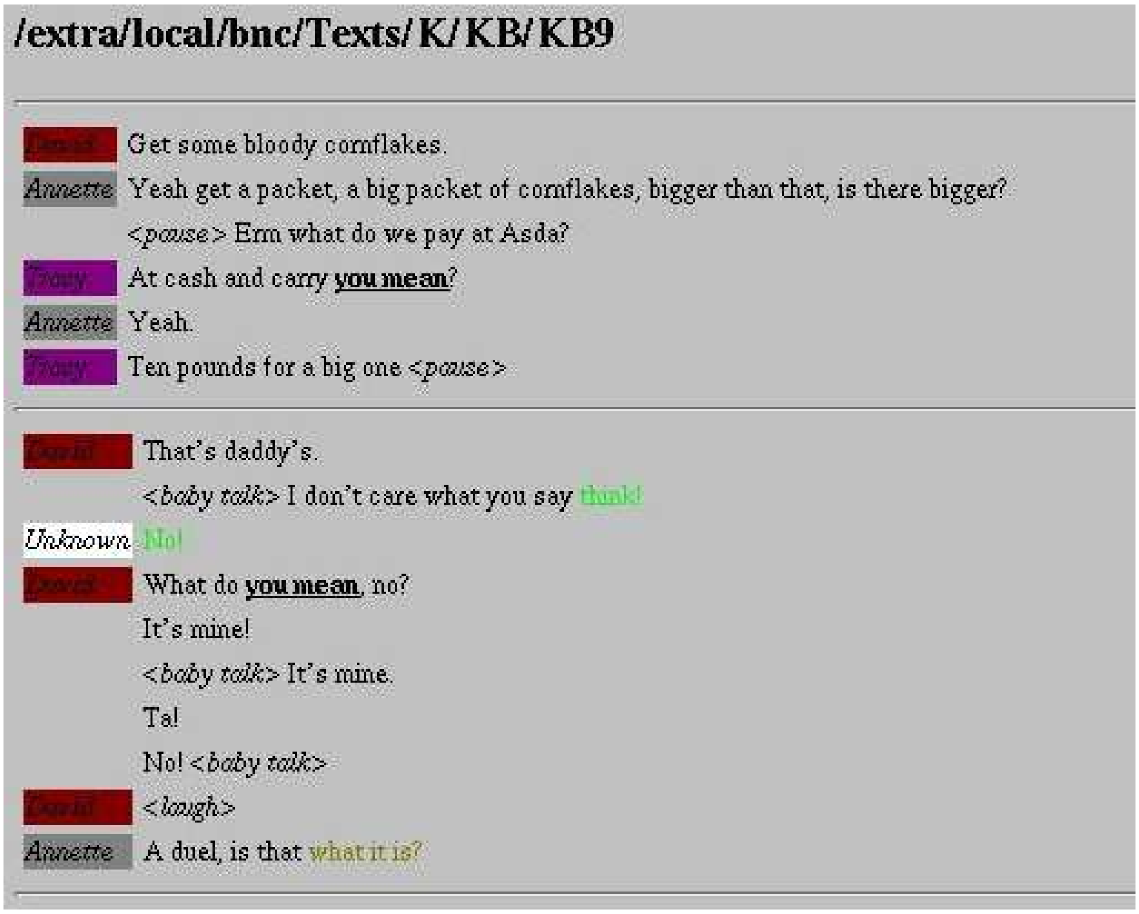

Figure 4: Typical "Matches only" display (search string '`you mean`', backward & forward context 2 turns)

matches displayed. It is unchecked by default.

3.2.3 Search Parameters

The parameters available here are a subset of those available in the original search, and have exactly the same function as described in section 3.1 above. Changing any values at this point may, of course, cause the number of matches for any particular file to change. The values are initially the same as those used for the search.

3.2.4 Saving Searches

A "Save this Search" button is available, together with edit boxes to specify the desired file name and a descriptive comment. If pressed, the search results page is saved, in a `searches` subdirectory on the server, as an HTML file for later viewing without having to repeat the search. The file name will have `.html` appended to it if not already present.

A link to the new file will be automatically added to the saved results page. The name of the link will be the same as the search string used (and the descriptive comment will be added). This results page (`saved.html`) can also be manually edited if desired.

3.3 Browsing

Browsing for files is performed in a similar way to searching: only the included/excluded file type and directory restriction need to be specified (see figure 5).

A table of suitable files is then shown as before (although only one button for each file is available here, "View"), from which files can be viewed (the whole file is displayed). An example is shown in figure 6. The sentence numbering option, first/last sentence/turn and sentence/turn options are available as before.

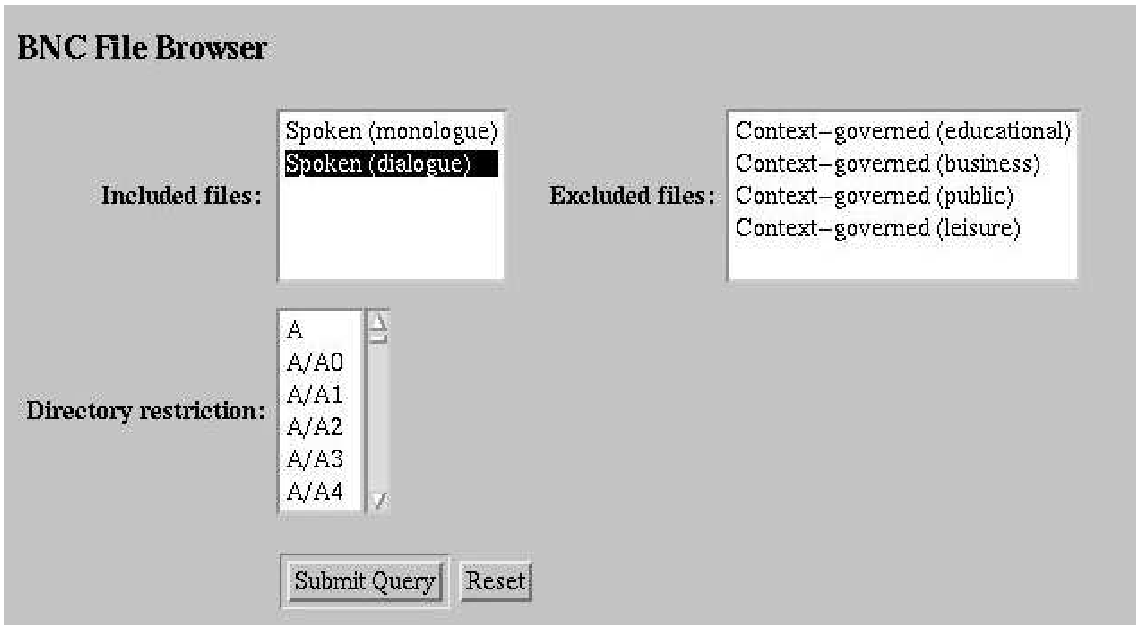

Figure 5: The browser interface

3.4 Viewing Previously Saved Searches

Previously saved search results can be recalled by clicking on links from a manually prepared page, or by choosing to browse the results directory. When recalled, the results are displayed exactly as described in section 3.2 above.

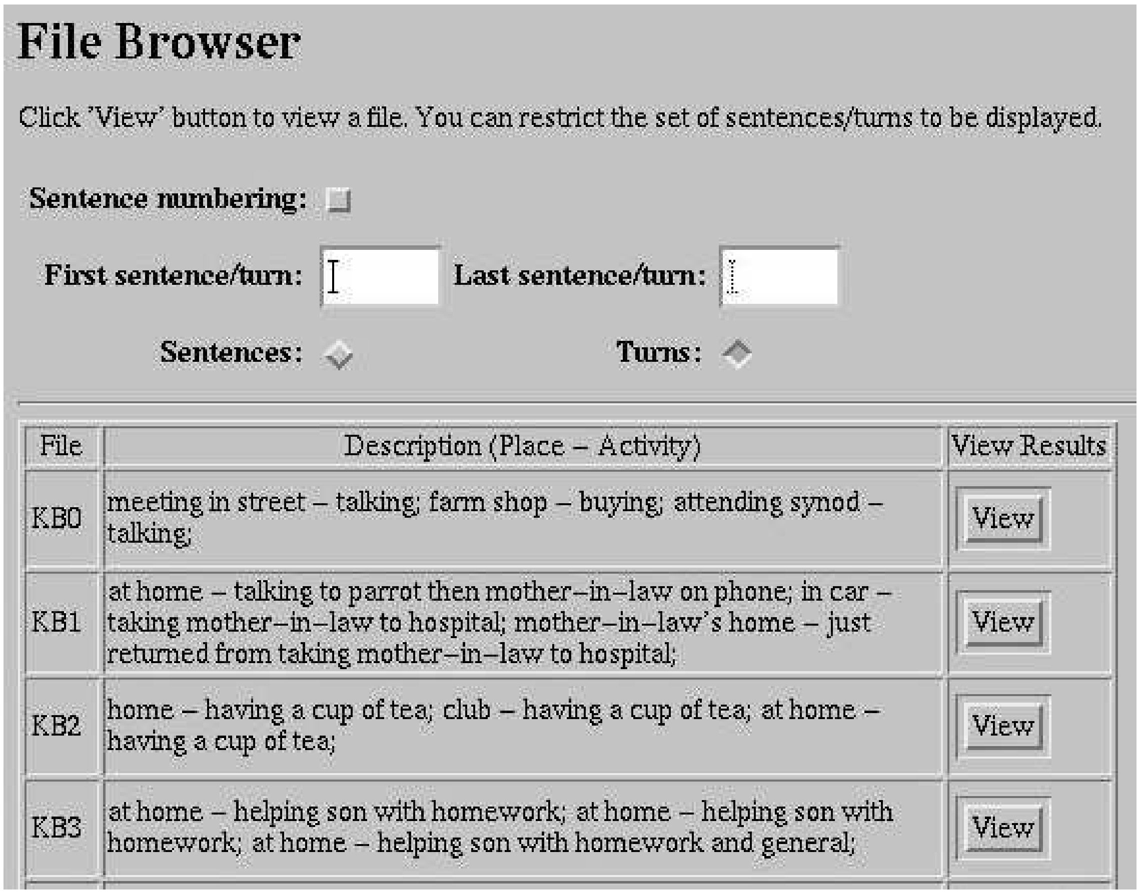

Figure 6: Typical browse results

4 Comparison with SARA

4.1 Functionality

Functions provided by SCoRE but not available in SARA:

- Repeat search
- PoS tag search
- Sentence/turn search scope restriction
- Display of dialogue with forward/backward context
- Display of dialogue with speaker names, overlapping portions etc. easily legible

Functions provided by SARA but not available in SCoRE:

- Word & collocation frequency calculation
- Concordance (cross-file comparative) display

- Sociolinguistic information
- Coverage of written portion of corpus (not yet fully implemented in SCoRE, but hoped to be added in later versions)

4.2 Other differences

Operating System The SARA server runs under 32-bit Windows and some versions of Unix; the SARA client runs under 32-bit Windows only. The SCoRE server runs under any operating system with an implementation of Perl and a web server (although has only been tested under Linux); the client runs in any table-capable web browser (tested with Netscape under Windows and Linux, and Internet Explorer under Windows).

Speed The SARA server uses a pre-compiled database for its initial word/string search, making this first stage extremely fast. Subsequent detailed viewing is slower. The nature of the SCoRE search functions make it difficult to compile an equivalent database, so the corpus files are searched directly, making the search process much slower.

Flexibility As no compiled database is required, SCoRE can be easily added to, and could be adapted for use with other corpora, particularly those encoded using SGML.

4.3 Summary

While the speed and collocational functions of SARA make it more useful for traditional corpus-based techniques (e.g. word frequency, collocation-based work - see (Aston and Burnard, 1998) for an introduction to its use in these fields), SCoRE provides many extra features required for dialogue research.

References

Guy Aston and Lou Burnard. *The BNC Handbook: Exploring the British National Corpus with SARA*. Edinburgh University Press, 1998.

Lou Burnard. *Reference Guide for the British National Corpus (World Edition)*. Oxford University Computing Services, 2000. URL `ftp://sable.ox.ac.uk/pub/ota/BNC/`.

Tony Dodd. *SARA: Technical Manual*. Oxford University Computing Services, 1997. URL `http://info.ox.ac.uk/bnc/sara/TechMan/`.

AJ0	adjective (general or positive), e.g. 'good', 'old'
AJC	comparative adjective, e.g. 'better', 'older'
AJS	superlative adjective, e.g. 'best', 'oldest'
AT0	article, e.g. 'the', 'a', 'an', 'no'
AV0	adverb (general, not sub-classified as AVP or AVQ), e.g. 'often', 'well', 'longer', 'furthest'
AVP	adverb particle, e.g. 'up', 'off', 'out'
AVQ	wh-adverb, e.g. 'when', 'how', 'why'
CJC	coordinating conjunction, e.g. 'and', 'or', 'but'
CJS	subordinating conjunction, e.g. 'although', 'when'
CJT	the subordinating conjunction 'that', when introducing a relative clause, as in "the day that follows Christmas"
CRD	cardinal numeral, e.g. 'one', '3', 'fifty-five', '6609'
DPS	possessive determiner form, e.g. 'your', 'their', 'his'
DT0	general determiner: a determiner which is not a DTQ, e.g. 'this' both in "This is my house" and "This house is mine"
DTQ	wh-determiner, e.g. 'which', 'what', 'whose', 'which'
EX0	existential 'there', the word 'there' appearing in the constructions "there is...", "there are ..."
ITJ	interjection or other isolate, e.g. 'oh', 'yes', 'mhm', 'wow'
NN0	common noun, neutral for number, e.g. 'aircraft', 'data', 'committee'
NN1	singular common noun, e.g. 'pencil', 'goose', 'time', 'revelation'
NN2	plural common noun, e.g. 'pencils', 'geese', 'times', 'revelations'
NP0	proper noun, e.g. 'London', 'Michael', 'Mars', 'IBM'
ORD	ordinal numeral, e.g. 'first', 'sixth', '77th', 'next', 'last'
PNI	indefinite pronoun, e.g. 'none', 'everything', 'one' (pronoun), 'nobody'
PNP	personal pronoun, e.g. 'I', 'you', 'them', 'ours'
PNQ	wh-pronoun, e.g. 'who', 'whoever', 'whom'
PNX	reflexive pronoun, e.g. 'myself', 'yourself', 'itself','ourselves'
POS	the possessive or genitive marker ''s' or ' ' '
PRF	the preposition 'of'
PRP	preposition other than 'of', e.g. 'about', 'at', 'in', 'on behalf of', 'with'
TO0	the infinitive marker 'to'
UNC	unclassified items which are not appropriately classified as items of the English lexicon
VBB	the present tense forms of the verb 'be', except for 'is' or ''s', i.e. 'am', ''m', 'are', ''re', 'be', 'ai' (as in 'ain't')
VBD	the past tense forms of the verb 'be', i.e. 'was', 'were'
VBG	the -ing form of the verb 'be', i.e. 'being'
VBI	the infinitive form of the verb 'be', i.e.'be'
VBN	the past participle form of the verb 'be', i.e. 'been'
VBZ	the -s form of the verb 'be', i.e.'is', ''s'
VDB	the finite base form of the verb 'do', i.e. 'do'
VDD	the past tense form of the verb 'do', i.e. 'did'
VDG	the -ing form of the verb 'do', i.e. 'doing'
VDI	the infinitive form of the verb 'do', i.e. 'do'
VDN	the past participle form of the verb 'do', i.e. 'done'
VDZ	the -s form of the verb 'do', i.e. 'does'
VHB	the finite base form of the verb 'have', i.e. 'have', ''ve'
VHD	the past tense form of the verb 'have', i.e. 'had', ''d'
VHG	the -ing form of the verb 'have', i.e. 'having'
VHI	the infinitive form of the verb 'have', i.e. 'have'
VHN	the past participle form of the verb 'have', i.e. 'had'
VHZ	the -s form of the verb 'have', i.e. 'has', ''s'
VM0	modal auxiliary verb, e.g. 'can', 'could', 'will', ''ll', ''d', 'wo' (as in 'won't')
VVB	the finite base form of lexical verbs, e.g. 'forget', 'send', 'live', 'return'
VVD	the past tense form of lexical verbs, e.g. 'forgot', 'sent', 'lived', 'returned'
VVG	the -ing form of lexical verbs, e.g. 'forgetting', 'sending', 'living', 'returning'
VVI	the infinitive form of lexical verbs, e.g. 'forget', 'send', 'live', 'return'
VVN	the past participle form of lexical verbs, e.g. 'forgotten', 'sent', 'lived', 'returned'
VVZ	the -s form of lexical verbs, e.g. 'forgets', 'sends', 'lives', 'returns'
XX0	the negative particle 'not' or 'n't'
ZZ0	alphabetical symbols, e.g. 'A', 'a', 'B', 'b', 'c', 'd'

Table 1: BNC Part-of-Speech Tag Scheme