PCW32

BRITISH NATIONAL CORPUS

PROPOSED OUTLINE SPECIFICATION

INPUT SEARCH AND RETRIEVAL PACKAGE (ISRP)

Geoffrey Leech and Michael Bryant

(Revised Version 22-02-93)

1 OVERVIEW

1. The objective is to produce as comprehensive a facility as possible, within the budgetary and time constraints of the project, for a typical user of the corpus interested primarily in linguistic information. This paper outlines the scope of the general objective. A separate document addresses the issue of what might be achievable.

2. A user may already have made some selection from the corpus in obtaining data from the OUCS database (to be held locally for processing), but the assumption will be that in principle the whole corpus could be interrogated, perhaps using wide or local network access.

3. The ISRP will consist of "front end" user interface components (the "input" part of the package) and "back end" tools for carrying out the required processing. The "front-end" should be Windows based and the "back-end" should comprise modular components designed to be command-line driven and able to be interfaced to the ISRP "front-end" and other applications. The "back-end" components will be developed in 'C' and an application builder used to develop the "front-end", with a view to maximising portability. Initially the ISRP will be developed on a Sun workstation using the Unix operating system.

2 KEY DESIGN PRINCIPLES

1. The components of the ISRP will be developed in a modular fashion, producing code which can be used in different modes of operation (e.g. as parts of stripped-down, command-line oriented programs, or part of a totally integrated GUI). This will enable useful "products" to emerge throughout the development phase implementing different parts of the functionality of the overall package. Given the ambitious nature of the "wish- list" on which the ISRP is based, this should mean that the package will grow like an onion, adding layers of functionality up to the end of the Project's funding period. User-interface modules will be developed in parallel with those relating to search, retrieval and processing.

2. The size of the corpus on which the ISRP might have to work is a major challenge. The intention is to incorporate comprehensive indexing prior to interrogation by the user, to minimise retrieval times. In addition tools will be provided for index building by the user. The development of these aspects of the ISRP are likely to receive most

attention in the early stages. Given adequately indexed data, all the search and retrieval needs outlined below should be implementable.

3. The highest development priority will be to assemble a package which will work with the BNC corpus tagged with the C5 grammatical tagset, and the core corpus tagged with the C6 grammatical tagset. The next level will be to develop modules to deal with the parse-tree annotation (in C6).

3 ISRP FUNCTIONAL REQUIREMENTS

3.1 Search Domain

Specification of what part(s) of the corpus are to be searched

1. It should be possible to name single or multiple divisions of the corpus, perhaps by means of a drop-down menu-style interface:
   - section
   - domain
   - year
   - sex

   etc.
2. There should be a separate option for selecting a subcorpus of varied composition and size: e.g. the Core Corpus of 2 million words.

3.2 Search Argument

1. CDIF mark-up characteristics should be included in the Search Argument as appropriate.
2. The search facility should include searches on
   - individual words
   - words with initial, medial, or final wild-cards
   - word combinations (including word sequences [up to n], and word sequences containing wild-cards spanning $\leq n$ words)
   - word(s) with tag(s), with wild-cards on either or both
   - combinations of words + tags in sequences, with or without wild-cards
   - subtrees with specified nodes (for searches from the syntactically annotated sections of the corpus)
3. It should be possible to include punctuation marks as separate "words" in the search argument.
4. Is should be possible to search in case-sensitive or non-case-sensitive mode.
5. In the case of portmanteau tags, it should be possible to search on one half of the portmanteau, or else on both halves.

6. Regarding annotations, punctuation marks, and diacritics in general: it should be possible to indicate any such features in search arguments.

   On the other hand, for word-based searches, it should also be possible to treat punctuations and diacritics as "invisible" to the search process.

3.3 Operations on Search Arguments

1. It should be possible to combine smaller search arguments into more complex ones by the Boolean operators *and, or,* and *not.*

2. To facilitate building up complex search arguments (especially by means of the *or* operator), it should be possible to set up a search argument file for possible editing and re-editing.

3.4 Operations on Output Data

For each Search Match X, it should be possible to

1. Specify the context to be displayed

   (a) as a window of n words to be displayed to the left and/or right of X
   (b) as a window of n lines of text to the left and/or right of X
   (c) as the sentence containing X

2. Specify the manner of display:

   (a) as a KWIC concordance
   (b) as a KWAL concordance, with the matched item displayed amid a multi-line unit
   (c) as a variable in terms of number of words or number of lines, with a separate variable for the preceding and the following contexts

3. Specify the ordering of matched items

   (a) order according to position in the corpus
   (b) order alphabetically according to left-hand context
   (c) order alphabetically according to right-hand context

4. Provide frequency count files, giving

   (a) Count of matches found
   (b) Count of word/sentences/lines searched
   (c) Normalized frequency (matches per 1,000 words, say)
   (d) Data of types (a)-(c) to be provided for each subsection of the corpus, if required
   (e) Frequency counts of the above types to be added at the end of the concordance file if desired

5. Provide more complex frequency information, such as:

(a) Most frequent n-word collocations of a given word or word-tag combination, down to a specified limit of m occurrences

(b) Frequency list of words or word-tag combinations, in rank order or alphabetical order

(c) Frequency list of collocations in rank order and alphabetical order

(d) Frequency list of tag-sequences, on the lines of (a) and (b)

6. Provide a sample from the Output File of desired size and composition

   Users often wish to control the number of examples they wish to examine in a concordance output. E.g.: 500. There should be a facility for specifying the size of the sample needed, and of random sampling from output files for that purpose.

   This random sampling should also be applicable to two or more files representing different sections of the corpus.

3.5 General Points

1. Each example in the concordance output should be identifiable by an address giving its location in the corpus. It should be possible to switch off or abbreviate the corpus addresses when desired.

2. There should be default settings for all choices made by the user, except for the specification of the Search Argument itself.