WORLD WIDE WEB ACCESS TO CORPORA

Doug Arnold
Department of Language and Linguistics
University of Essex

ABSTRACT
For a variety of reasons, getting started in corpiis *linguistics is difficult. The starting point for the W3Corpora* project *was that providing access to corpora via* the *World Wide Web would offer a way round many of these difficulties.* The *paper describes the motivation for the* project, *gives an overview* ofthe *W3Corpora web site* thnt *was built, and gives a technical but accessible description of the corpus searching techniques used. Some discussion of similar web sites is provided.*

KEYWORDS: *corpus linguistics, WWW corpora, corpus search techniques*

RESUMEN
Lu iniciación o la lingüística del *corpus lleva* consigo *una* serie *de dificultades. El objetivo* prioritario *del proyecto W3Corpora es ofrecer acceso* a *corpus lingüísticos* a *través* de *la World Wide Web con el fin* de *subsanar parte de esos problemas y dificultades iniciales. En este artículo se describe la motivación del proyecto, se ofrece una visión global* de *la página web construida* pura *el proyecto W3Corpora y se presenta una* descripción *técnica, aunque de fácil comprensión, sobre las técnicas* de *búsqueda y consulta empleadas.* El *trabajo concluye comparando* la *web site W3Corpora con otras similares ya existentes.*

PALABRAS CLAVE: *lingüística* del *corpus, corpus en WWW,* técnicos de *búsqueda* en *corpus*

I. INTRODUCTION

Linguistic corpora are important in many areas of linguistics and related disciplines. and are becoming more important as more and larger corpora become available. and better techniques are introduced for manipulating them. Despite this. their use is less widespread than it should be. A major reason for this is the difficulties that face the newcomer (student or established researcher) in getting started. The difficulties are of several kinds: corpora tend to be very large (perhaps too large to fit easily on a user's own personal computer). and the software required to manipulate them can be difficult to obtain and install. especially for the computational naive user. Users must also familiarize themselves with the software. and then decide what to do with it (one can believe that linguistic corpora should be useful and interesting withoiit knowing *how* to exploit them).

In theory. the World Wide Web (WWW) should have something to offer here. In principle. if corpora are available over the WWW. then access should require nothing except a Web connection and a browser: there should be no need to obtain and install corpora. or download and install software. The interface to the corpus inanipulating tools should already be familiar (since it will be based on the users web browser). Moreover. the Ilexibility of hypertext (html) should make it easy to provide access to supporting information. including tutorial information on 'what to do next'.

This paper reports on an effort to realize this: the W3Corpora project.[1] Apart from giving a general overview (Section 3). in Section 3. I will go into some detail about the technical aspects at a generally accessible level. with a view to 'de-mystifying' the whole business for the newcomer. and providing a point of departure for a more expert reader (who might. for example. want to implement something better). The paper may also be of value in giving some comparison with other Web-based corpus manipulation (see Section 4). Section 5 is a conclusion.

The general aim of making corpus resources available to naive users over the WWW suggests a number of desiderata:

- The system should be immediately usable by anyone with WWW access and a Web Browser. for example:
  - it should be usable with essentially any generally available browser:
  - it should be usable without the need to install or download any programs:
  - it should be usable without the need to register and get authorization.

- The interface should be as 'friendly' and easy to use as possible: it should be supported by extensive and appropriate on-line help. and backed up by detailed information about corpus linguistics in general. 'Friendliness' is manifest both in the way corpus searches are specified and the way results are show. As regards the former. standard searches (e.g. searching for whole words only) should be immediately available. but so should a more powerful tools (e.g. full regular expressions). As regards the presentation of results. apart from being as clear as possible. there should be scope for the user to modify the way results are presented.

- One can only really learn about corpus linguistics by doing it. so general information must be supplemented by practical information about how to 'do' corpus linguistics using a system such as this.
- In terms of functionality. the system should provide at least access to information about frequency of expressions and their context (e.g. in Key Word in Context (KWIC) concordance style). it should be possible to vary the amount of context provided and to access wider context. I should be possible to perform various operations on the results of searches (e.g. sorting. reducing. some other kinds of editing).
- It is typical of novice users that they make mistakes with queries: thus. there should he some method for users to correct and 'refine' their queries v e n easily.
- It should be possible for a user to install and search their own corpora -in this way a user can explore not only what is possihle in general. but what is possible in relation to the kinds of material they are interested in.
- A major problem with WWW access is the overhead incurred by transniission of data across the network. Once a user is convinced of the usefulness of corpus resources. they may prefer to install software locally. Hence. not only should information about other corpus searching software should be provided. hut the source code for the system itself should be freely available. allowing the system to be installed and run locally at other sites.

II. A USER'S OVERVIEW

We begin with a brief overview of the W3Corpora website. The site is divided into three main parts (see Table 1 for Weh Addresses):

The Search Engine which provides the user with an interface for carrying out corpus searches.

General Information Pages where the user can learn about corpus linguistics in general (e.g. 'What is a corpus?'. issues of corpus designand annotation. research areas. bibliography. etc). Here. the user will find the kind and level of information one might expect in a conventional introductory text book. such as Barnbrook (1996) or Kennedy (1998).

Tutorial Pages where the user can Iind out how to use the tools provided. and where some of tasks for which corpora are useful are descrihed in soine detail. with practica1 instructions (e.g. investigating the meaning of word. comparing two similar words. comparing hou one word is used in different contexts. investigating spelling. investigating the choice of preposition such as of/for in a context like an explanation ...something). Here and elsewhere. the emphasis is on classical corpus linguistics (rather than e.g. statistical techniques that can be built on top).

The key aim is to answer. as quickly and easily as possible the two questions: 'How do I use this system?' and 'What can I use it for?'. so users can go and get some practica] experience. and decide whether to continue (perhaps with other software. installed locally).

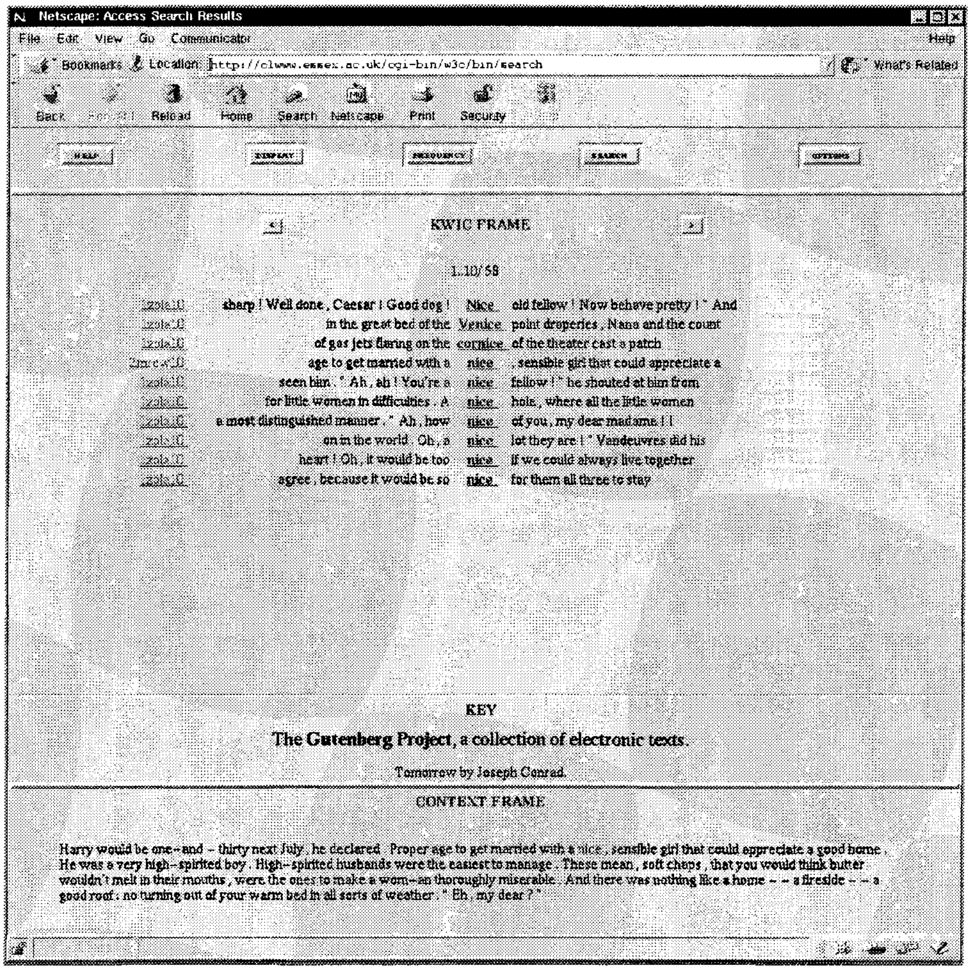
'Top level':	http://clwww.essex.ac.uk/w3c/
Search Engine:	http://clwww.essex.ac.uk/w3c/corpus_ling/content/search_engine.html
General Information:	http://clwww.essex.ac.uk/w3c/corpus_ling/content/introduction.html
Tutorial:	http://clwww.essex.ac.uk/w3c/help/intro/start_page.html

Table 1. Web Addresses for W3Corpora Resources

When a user enters the website. they are presented with a 'welcome' page that offers access to each of the above. Supposing they choose to access the search engine. they are presented with a screen which allows them to specify a query by: (a) Selecting a Corpus -that is. choosing a collection of texts to search; (b) Selecting a Search String -that is. specifying what they want to search for; and (c) Submitting the query.

In the first two cases the user is presented with a complete new screen containing a form to fill in. Also on this (and every) page is a "HELP" button. which generates an appropriate help message (i.e. appropriate to where the user is in the whole process - typically. it describes what sorts of action the user is expected to carry out. and what any specialized terms mean).

In selecting a corpus. users are presented with a list of the available texts. from which they can select as many as desired. When their selection is made. they are returned to the main search page. to specify a "search string". This involves specifying a type of search. and a pattern. Types of search include searches that are restricted to the beginning and/or end of words. as well as ordinary regular expressions.' When this choice is confirmed. the user is again returned to the top level page. from where they can instigate the actual search by clicking on the "SUBMIT" button.


Netscape: Access Search Results
File Edit View Go Communicator Help
Bookmarks Location: http://clwww.essex.ac.uk/cgi-bin/w3c/bin/search What's Related
Back Forward Reload Home Search Netscape Print Security
HELP DISPLAY FREQUENCY SEARCH OPTIONS

KWIC FRAME

```
                                          1..10/58

zola.3   sharp ! Well done , Caesar ! Good dog !   Nice     old fellow ! Now behave pretty ! " And
zola.3                   in the great bed of the   Venice   point draperies , Nana and the count
zola.3                  of gas jets flaring on the cornice  of the theater cast a patch
tmrw.3                   age to get married with a nice     , sensible girl that could appreciate a
zola.3                 seen him . " Ah , ah ! You're a nice fellow ! " he shouted at him from
zola.3            for little women in difficulties . A nice hole , where all the little women
zola.3        a most distinguished manner . " Ah , how nice of you , my dear madame ! !
zola.3                      on in the world . Oh , a nice   lot they are ! " Vandeuvres did his
zola.3                  heart ! Oh , it would be too nice   if we could always live together
zola.3              agree , because it would be so nice     for them all three to stay
```

KEY

The Gutenberg Project, a collection of electronic texts.

Tomorrow by Joseph Conrad.

CONTEXT FRAME

Harry would be one - and - thirty next July , he declared . Proper age to get married with a nice , sensible girl that could appreciate a good home . He was a very high - spirited boy . High - spirited husbands were the easiest to manage . These mean , soft chaps , that you would think butter wouldn't melt in their mouths , were the ones to make a wom - an thoroughly miserable . And there was nothing like a home - - a fireside - - a good roof : no turning out of your warm bed in all sorts of weather . " Eh , my dear ? "

Figure 1. KWIC Display of first 10 search Results (for regular expression search[Nn]ice). The user has selected options which allow results to be deleted and which indicate which sub-corpus each hit coines froin. At the bottom ofthe page the wider context of one of the hits is displayed (the user has clicked on one of the individual hits to obtain this)

Suppose. for concreteness. the user chooses the Gutenberg corpus. selects the documents in Table 2 to form the sub-corpus. and opts for a search for the regular expression `[Nn]ice` - that is. the string "nice". possibly beginning with an uppercase.

13,000 *Dreams Interpreted [Or ...What's In A Dream]*	by Gustavus Hindman Miller.	drmnt10
. *Hero of Our Time*	by M.Y. Lermontov.	ahero10
A Child's History of England	by Charles Dickens.	achoe10
A *Collection of Ballads*	by Andrew Lang.	cblad10

Table 2. Sub-Corpora (and their reference numbers. indicating the corresponding file names).

By default. the first search results shown relate to frequency. and give the simplest information about number of matches in each sub-corpus. It is also possible to view how the results are distributed across the different parts of the sub-corpora or to see the numbers of different words that were matched (looking at these results will immediately show the user that matches were returned for *Venice* and cornice. which may or may not have been intended).

For more insight into how the word nice is used. the user may click on the "DISPLAY" button at the top of the screen. which will generate a KWIC (Key Word In Context) display. as in Figure 1. This shows the user the first 10 results of the search. together with several words either side. Clicking on one of the key-words shows the wider context in the "Context Frame" at the bottom of the screen. The "DELETE" buttons associated with each result allow the user to reject some of the results (presumably keeping the remainder for further study). Other buttons allow the user to set various options (e.g. how many words of context are displayed. which results are shown first. etc.). access other forms ofdisplay. and see more of the results. A slightly clearer view of the sort of information that can appear in the KWIC part of the display can be seen in Figure 2.

The "SEARCH" button allows the user to perform another search. or -perhaps more interestingly- 'refine' the existing search. As already noted. the search we are describing matches *Venice* and *cornice*. Perhaps these are unintended -perhaps the user really wants words related to nicr (*nicely. niceness* etc.). Instead of simply deleting the unintended matches (which could be quite tedious). it is possible to 'refine' the existing search. that is. to perform a further search on the results of an existing search. Using `^[Nn]` (which only matches at the start of words) for the refined search has the desired effect. The results of this search can then be further refined. In this way it is possible for the user to build up an entire 'history' of searches. and to move back and forth between them.

As well as searching the corpora that are provided at the site. it is possible for users to upload (by anonymous ftp) their own texts and search them (step by step instructions on the use of ftp are given when help is requested on the page dealing with user-defined corpora). The choice of a user-defined corpus is available when the user selects the corpus to search. What happens is that when a user selects this option (in place of. e.g. Gutenberg). files that have been uploaded in this way are processed for searching. and then searched in the usual way - the full functionality of the search engine is available (the only difference is the information about the corpus that is displayed.) In this way. a user can explore whether the searching techniques are useful on their own material.

```
achoel0   Holy Land, and afterwards died at  Venice   of a broken heart. Faster and
aherc10   so very elegant, his complexion so  nice   and white, his uniform so brand
drmnt10      dream of having an abundance of  nice   , clean crockery, denotes that you will
drmnt10            young woman to dream of a  nice   , ready-made shirt-waist, denotes ttiat
```

Figure 2. Typical KWIC results for retunied for search for `[Nn]ice`

III. TECHNICAL DETAILS

This section is intended to give some of the technical details. at a generally accessible level.

A web browser interprets a command to visit a Web address such as `http://clwww.essex.ac.uk/w3c/`. as an instruction to contact the web serving program on the machine whose Web address is `clwww.essex.ac.uk` (this is called the *server*. the other machine. the one the user is sitting in front of. is called the *client*) and ask to be allowed to see the directory `/w3c/`. In general. this directory will contain files. one of which may be called something like `index.html` which the web server will send to the browser. and which the browser will render as nicely formatted test. What the file actually contains is a mixture of test and Hypertext Markup Language (HTML). along the lines of Figure **3.** The details of this are unimportant here. but for example `<B>` tells the browser to start using bold face. and the `</B>` tells it to stop (these bits of HTML are called *tags*).

```
<HTML>
<HEAD>
  <TITLE>W3-Corpora </TITLE>
</HEAD>
<BODY background="./backgr2.gif">
  <CENTER>
  <B> Welcome to the W3-Corpora Site!</B>
<CENTER>
  ....
</BODY>
</HTML>
```

Figure 3. A file containing some text marked up in HTML

As well as getting ordinary files. the browser can also ask the web server to execute prograrns. For rxarnple. if you ask your browsrr to visit `http://clwww.essex.ac.uk/cgi-bin/w3c/w3c`. then the web server will execute the 'w3c' prograrn in the directory `/cgi-bin/w3c`. What executing thisprograrn actually does is not very interesting: it simply prints sorne HTML. i.e. something like Figure 3 (with a couple of small differences) and the web browser renders this for the user just as if it had been in a file. The main differencrs are that what the prograrn prints contains some additional markup tags: the `<FORM>`. and `<INPUT>` tags. Roughly. the `<FORM>` tag is used to indicate that the associated piece of the HTML file. and the associated area of the browsers screen. is to be used for input from the user. The `<INPUT>` tag is used to say what will be sent back to the server. For example, some of the text that this program sends to a web browser can be seen in Figure 4. Here the `<FORM>` tag says something like "pass the user input to the program `cgi-bin/w3c/input/corpus`". The `<INPUT>` tag says "at this point in the page. put a button that the user can press. label it *Corpus*. and when the user presses it. execute the program above with the value of the variable `Corpus` set to `"Gutenberg."`(If the 'type' in the `<INPUT>` tag had been `text.` it would have produced a space for the user to write sorne text. which would be returned as the value of the `Corpus` variable. The reader who wants to know more about this sort of thing will find it in almost any book about designing materials for the WWW. for example Kim (1996)).

```
<FORM METHOD="POST" ACTION="http://clwww.essex.ac.uk/cgi-bin/w3c/input/corpus">
     <INPUT TYPE="submit" NAME="Corpus" VALUE="Gutenberg">
</FORM>
```

Figure 4. A file containing part of an HTML form

Thus. Ior the most part. designing a Web interface to corpora involves:

1. writing files containing HTML to contain information that does not change (these can be written with an ordinary word processor if you don't mind writing all the tags yourself);
2. writing a large number of programs which produce HTML -any text that might vary from user to user. or session to session has to be created in this way. of course (any prograrnrning language can be used for this. but Perl (e.g. Wall and Schwartz (1991)) is a natural choice for any large scale program that involves text processing);
3. obtaining corpora; and
4. writing programs to take the information the user supplies and actually search corpora.

The first of thesr is completely straightforward from a technical point of view. The only notable point about the second of these is the sheer volume of code that is required. The W3Corpora system involves over 17.000 lines of code in about 40 prograrns. Less than 5% of this is required for the actual corpus processing. Probably the hardest part of the whole business is getting the corpora to search: here the main problem is that most corpus collections are under copyright. and making them web accessible. in however limited a way. is at least potentially an infringement of copyright. and therefore illegal. Fortunately. there are some rather extensive public domain corpus collections (at least in English). notably the Gutenberg corpora.' The technically interesting task is the last one listed. and it will be the topic ofmost of the rest of this section.

The techniques used forcorpus searching are most easily understood by starting with how corpora are prepared. I will describe this in relation to a toy corpus of about 30 words (Figure 5).

```
I believe he left his house to his friends, his money to the
poor, and his clothes to the nation.
```

Figure 5. Sample Corpus

III.1. Corpus Preparation

Including punctuation. this toy corpus consists of twenty three 'tokens'. and sixteen 'types' (e.g. the word-type *his* is present as three separate word-tokens). The first task is to 'tokenize' it. i.e. split it into individual tokens. This process yields two files: `xcorp.tok.` and `xcorp.item` (`xcorp` being the name of the file containing original corpus).

`xcorg.tok` Tokenized corpus -the whole corpus. one word-token per line (cf. the first column ofFigure 7). and with paragraph breaks represented by `<P>` tags. Simply printing this file without the line breaks and with the `<P>` paragraph markers replaced by blank lines recreates the original corpus.

`xcorg.item` This file contains a character string. with one character per word-token. punctuation character. or blank line of the corpus (which means one character per line of the tokenized corpus file): the ith letter is `W` if the corresponding token in the `.tok` file is a word: `P` if it is punctuation. `S` if it is a paragraph separator `<P>`. Cf. Figure 6.
A glance at the Iirst two columns of Figure 7 will show the relation between this file and the `.tok` file.
Given the information in this file it is possible to treat words and punctuation differently (e.g. when deciding how many items ofcontext they want either side of a key word. users can decide whether to count punctuation). It is also possible to access the whole of the paragraph containing a particular token.

```
WWWWWWWWWPWWWWWPWWWWWWP
```

Figure 6. A '.item' file

The .tok file is further processed producing a further six files for each sub-corpus. These are summarized in Table 3.

xcorp.tok	xcorp.item	xcorp.seq
I	W	3
believe	W	
he	W	S
left	W	11
his	W	9
house	W	10
to	W	16
his	W	9
friends	W	7
	P	1
his	W	9
money	W	1%
to	W	16
the	[illegible]	15
poor		14
	P	1
and	W	4
his	W	9
clothes	W	6
to	W	16
the	W	15
nation	W	13
	P	

Figure 7. The tokenized corpus (xcorp.tok). its analysis as words. punctuation. etc. (xcorp.item). and the sequence file (xcorp.seq)

xcorp.lex This file contains a list of the word-types that occur in the .tok file. sorted in lexicographic order (Figure 8). one type per line. One might think of'this as the *lexicon* ofthe corpus. Looking at this file will answer the question "Does word X appear in this corpus?" (more generally "Does a word matching the pattern X appear in this sub-corpus?"). On the other hand. if one thinks of the corpus as consisting of n word-types. w_1, w_2, ... w_n. then this file allows one to answer the question "What string ofletters corresponds to word w_i?" -e.g. in this case w_5 is believe. w_3 is and. and w_1 is the comma. See the first two columns of Figure 8.

xcorp.seg This contains a representation of the original corpus. but with word-tokens replaced by nunibers (the type numbers given in xcorp.lex).Cf. Figure 7. To recreate the original corpus. one would look at each element w_i of the `.seq` file and print the string associated with wi in the `.lex` file.

`xcorp.lex.freq` A representation of the word-type frequencies; each element is an integer -the ith element gives the frequency with which word-type w_i occurs in the corpus.

`xcorp.lex.idx` An index into the lexicon (`.lex`). This file records, for each word-type $w_i, \ldots, w_n$, where in the lexicon the string corresponding to w_i appears. For exaniple. the relation between `xcorp.lex` and `xcorp.lex.idx` might be as in Figure 8: w_5 (*believe*) begins at record 10 in `xcorp.lex`

To Iind the printed representation of w_i, one retrieves the ith element of `.lex.idx.` and consults the specified record of the lexicon. This allows for rapid access to the string corresponding to to any word-type in the corpus.

`xcorp.lex.pos` For each word w_i, the ith record of this file gives the positions in the `.seq` where this word occurs (i.e. where in the corpus one can Iind the corresponding tokens).

`xcorp.lex.pos.idx` This gives an index into `.lex.pos.`It gives. for each word-type w_i, the position in `.lex.pos` at which the list of occurrences of w_i can be found.

The relationship between these files can be seen in Figure 8. To take a concrete exaniple. w_{15} is the word *the*. If we look at the 15th record of `.lex.idx`, we Iind the figure 70. this is the position in the `.lex` file where the string representation of this word begins. The string representation(`the`) can be found by opening `xcorp.lex` at position 70 (and reading forward three characters). It occurs in the corpus 2 times. as witness the figure 2 in the 15th record of the `lex.freq` Iile. To find out *where* in the corpus it occurs. we look at the 15th record of the occurrences index -`.pos.idx.` which gives a value of 72. Opening the occurrences files `.lex.pos` at position 72. and reading the next 2 records (the frequency of this word in the corpus) will tells us that *the* appears at positions 14 and 21 in the original corpus.

`xcorp.tok`	The iokeiiized corpus: one word or punctuatioii element per line.
`xcorp.item`	This records, Sor each token. whether it is a word or punctuation. or markup.
`xcorp.lex`	"Lexicon" -word-types (as strings) in lexicographic order: for eacli i the string associated with w_i.
`xcorp.seq`	The original corpus. but with i in place of the corresponding string.
`xcorp.lex.freq`	Frequency information for eacli word-type w_i.
`xcorp.lex.idx`	An index into .lex: where in .lex does the string associated with w_i occur?
`xcorp.lex.pos`	Lists of occurreiices: the positions in .seq where tokens of wi occur.
`xcorp.lex.pos.idx`	Aii index into .lex.pos: where iii .lex.pos does tlie list of occurrences for w_i appear?

Table 3. Main Corpora Files

It will be seen from this that frequency information can be very rapidly calculated. In the simplest case one simply looks through the `.lex` file and for each line *l* where the pattern matches. and recovers record *l* from the `.lex.freq` file. Summing these gives the total number of hits in the corpus.

This admittedly somewhat complicated picture will become a little clearer in the following section. However. the advantage of having all these files may already be clear: the only searching involved in finding instances of a word (e.g. *the*) is the search through the 'lexicon' file to find 'which word' it is (it is word 15). After that. all we have to do is follow pointers that take us to exactly the information we need about frequency and location.

III.2. Searching

We can now look at what happens when a user actually instigates a search.

Unfortunately. the protocols that govern web connections mean that things are not quite as simple as one might hope. In particular. these protocols do not provide a way of 'maintaining state'. What this means is that there is no mechanism for keeping track of information which the user supplies from one point in a session to another. For example. suppose the user fills in a form to select a corpus. unless something is done to keep it. the information about corpus choice will be lost when he or she accesses another form to specify a search string. Standard HTML does not provide interactive forms (i.e. forms which change their appearance interactively as the user makes choices). so realistically. one often has to present the user with a sequence of separate forms. Moreover. in an application such as corpus searching. there is a substantial amount of information to remember: apart from corpus choice. search string. and actual search results. there are user options relating to the display. and possibly a search history to keep track of. In general. one possibility here would be to find a way of passing responsibility for this to the users machine (the client). But this is not an option here. given the desideratum that the system should work independent of any facilities on the user's machine other than the web browsing programme itself. The only alternative is to save all relevant information in files on the server. Most ofthe complexity of what follows is a result of this.

w_i	xcorp.lex	xcorp.lex.idx	xcorp.lex.pos	xcorp.lex.pos.idx	xcorp.lex.freq
1	,	0	10 16	0	
			23	8	1
3	I	4	1	12	1
	and		17	16	1
5	believe	10		20	1
6	clothes	18	19	24	1
7	friends	26	,	28	1
8	he	34		32	1
9	his	37	5 8 11 18	?6	4
10	house	41		52	1
11	left	47		56	1
12	money	52	12	E3	1
13	nation	58	22	64	1
14	poor	65	15	68	1
15	the		14 21	72	
16	to	74	7 13 20	80	

Figure 8. Files giving Lexicon, Occurrences. Frequencies, and lndices

When a user accesses the initial search page. or requests a new search. a lile is generated with a randomly chosen name (such as 325073). this file uniquely identifies the session. The user must nou[7] choose some sub-corpora and a search expression. Suppose the user has chooses the same sub-corpora and search term as above (`[Nn]ice` and the sub-corpora in Table 2). Confirming these choices puts this information into the 'session' file. and adds information about any options the user has selected to a '`.options`' file (e.g. `325073.options`. if the user changes options. these changes are recorded in this file). Typical contents for such files can be seen in Figure 9 and Figure 10. (In actuality. the number of options is quite large. and the `.options` file is much larger than this: these figures contain explanaton comments. after the '#'. which are not part of the real files).

Confirming the choices also starts the search process itself. as follows. First. the user's input is read from the 'session' file. and the search expression is split into 'components' -in general. a search expression may consist of several components. each of which is intended to match a separate word. For example (a search expression like `takes advantage`. which is intended to find instances of the word *advantage* immediately following the word *takes* consists of two components).

String:[Nn]ice	# What is the search string?
Type:Regular	# What kind of searcli expression is this?
Corpus:Gutenberg	# Which main corpus is being searched?
Entire:	# Is it over the whole corpus? (No)
Subcorpus:drmnt10 0	# Which sub-corpora should be searched? Sub-corpus 0 is dnnnt10
Subcorpus:achoe10 1	# Sub-corpus 1 is achoe10.
Subcorpus:cblad10 2	# Sub-corpus 2 is cblad10.
Subcorpus:ahero10 3	# Sub-corpus 3 is ahero10.

Figure 9. Contents of a 'session' file (e.g. 325073)

```
Corpus:Gutenberg                # Which corpus?
Sample:0                        # Where in the search history is this? (0=at the start)
Display_Initial:F               # Which results should be displayed initially? (Frequency)
Display:10                # How many items in the KWIC display at one time? (10)
KWIC_L:6                        # How many words to the left of target in KWIC?
KWIC_R:6                        # How many words to the right of target in KWIC?
KWIC_Display_Reference:0        # Should the KWIC display show the sub-corpus's reference?
KWIC_Display_Delete:0           # Should there be a DELETE option in the KWIC display?
```

Figure 10. Part of a `.options file`. with comments

Next. the search string is normalized to a regular expression (e.g. if the user asked for a 'whole word' search. then the search expression `nice` is changed to the expression `^[Nn]ice$` which only matches complete words). In outline. processing then proceeds as follows (this description is simplified in various ways. e.g. it does not address sorting of results).

1. For each sub-corpus selected:
   (a) for each component of the search pattem:
      i. open the appropriate files containing information about the words in the sub-corpus. their frequencies and positions (the appropriate `.lex`. `.lex.freq`. `.lex.pos.idx`. and `.lex.pos files`);
      ii. read through the `.lex file` for the sub-corpus looking for matches (this is the file that lists the word-types that appear in the sub-corpus). If a line in this file matches this component of the search pattern. then:
         A. if this is the first component of the search pattern. access frequency and position information for the corresponding tokens. and continue with other components (if any):
         B. if this is not the first component. then make sure any potential matches occur at positions adjacent to places where the previous component occurred.
      iii. record information about the start and end positions of each match (if there was only one component. then start and end are the same). and record frequency information;
   (b) Frequency information is written out to a 'frequency' file for the session (`.freq`). Position information is written in a 'results' file for the session (`.results`).

2. For each sub-corpus searched. we recover the strings corresponding to the successful matches. To do this. we look in the `.seq` file for the sub-corpus (where the corpus is represented as 'running text' but with word-type identifiers rather than strings). at the

appropriate positions. and for each such position recover the identifiers of the word-types; looking in the `.lex.idx` Gle for the corpus in the appropriate place gives the location of the required string in the sub-corpus's `.lex` file. To avoid having to do this again. the string information is also written onto a file for this session (this file has the extension `.li`).

A summary of the files involved in each session can be found in Table 4. The `.0` in the names of these files indicates that they are the result of the initial search in this session. If the user 'refines' the search to produce a search history. then files with names `325073.1.freq.` `325073.2.freq.` etc. will be produced.

325073	# 'Session file' Basic Searcli parameters
325073.options	# Other User options
325073.0.freq	# Frequency infonnation: How often each 'hit' appears in each sub-corpus
325073.0.results	# Position information: start and end position of each 'hit'
325073.0.sub	# List of sub-corpora for each 'hit' (i.e. which sub-corpus contained the hit)
325073.0.li	# List of strings: for each 'hit'. the actual text that was matched

Table 4. Files containing search results (a 'hit' is a successful match)

To generate a frequency report. we look at the `.freq` file (Figure 11). see that sub-corpus 0 contained two hits. see (from the session lile) that corpus 0 is `drmnt10`. and look up what this sub-corpus is called (10.000 *Dreams Interpreted...*). and print this information (appropriately formatted). Frequency information can be displayed very rapidly.

The relationship between the other files can be more easily seen if their content is printed out in vertical columns. as in Figure 12. For example. the first entry in the `.results` lile is `80892 80892`. the first entry in the `.sub` file is `1`: from this. we know that there was a match beginning at position 80892 in sub-corpus 1 (i.e. `achoe10`). and ending at the same point. Since the Iirst entry in the `.li` file is `Venice`. we know that the string that occurs there is *Venice*.

0:2	# Sub-corpus 0 is drmnt10.2 hits
1:1	# Sub-corpus 1 is achoe10. 1 hit
2:0	# Sub-corpus 2 is cblad10. 0 hits
3:1	# Sub-corpus 3 is ahero10. 1 hit
All:4	# Total: 4 hits

Figure 11. A `.freq` file: how oflen the search string is matclied in each sub-corpus

.results (Start-End Positions in Corpus)	.sub (Sub-corpus number)	.li (the string that was matched i.e. which occurs at that position)
80892 80892	1	Venice
3150 3150	3	nice
51445 51445	0	nice
165263 165263	0	nice

Figure 12. The relation between some of the files holding search results

Given these results. if a KWIC display is requested for (say) the first result with (say) six words of context. then all that is required is to open the `.seq` file of the appropriate corpus (`achoe10.seq`) at record 80892. read six records either side. and for each record recover the associated string. This can then be printed. with appropriate markup that puts it in an appropriately sized window. and with an HTML tag on the string that was matched that will cause wider context to be displayed if the user clicks on this word. Actually accessing wider context is similar. except that one should read backwards and forwards for the nearest paragraph markers (where these are is quickly recovered from the appropriate `.item` file Ior the sub-corpus). Notice that no searching is required after the initial search: all that is necessary is to open files and follow pointers to appropriate places. Apart from the need to open and close Iiles. and hence for disk access. the process is very efficient.

As noted. if the user requests a refinement of this search. the results go in files with the same 'session' name. but numbered sequentially. e.g. `325073.1.freq`. `325073.2.freq`. Managing a history of searches is then straightforward. Though we have not mentioned it. it is also possible to handle some kinds of tagged corpora in exactly the same way. In particular. corpora which are tagged by associating a tag with each word can be searched by treating the tags as separate words (so searching for word *w* followed by tag T just involves a 'two component' search for *w T*).

IV. ALTERNATIVES. COMPARISON

There are a large number of tools and systems that offer something similar to what the W3Corpora site seeks to provide. A list of some for which information can be obtained over the WWW is provided in Table 5. They range from simple Unix con-imand-line style programs like *Ptx*. to programs with sophisticated Graphical User Interfaces (like *Xkwic*). Some are commercial products. some completely free. and some can be obtained at a nominal cost. or are free Sor research purposes (requiring only registration). Some can be run directly on unformatted text. while others require the text to be preprocessed to a high level (essentially it must be turned into a database -the corpus search program is then in effect a database query program). Many are very expressive. allowing a wide range of searches (e.g. allowing recovery of collocational information. and some relatively sophisticated statistical and/or grammatical information). and some are designed to be multilingual (e.g. Sor searching parallel corpora). Foi many there are important limitations. however. Typical limitations include: being available for only one platform (e.g. *Ptx* and *Xkwic* require Unix. *Conc* is available only for Macintoshes. *LEXA*. *Wordsmith Tools*. and *Sara* are only available Sor MSDOS/Windows); being only able to handle files up to a certain size (files small enough to fit into memon all at once); being designed only for a specific corpus: requiring registration and installation (which may be more or less easy depending on the usual range of factors).

Of course the W3Corpora system does not suffer from these particular limitations (nor. to be sure. does it have the full fiinctionality of the most powerful systems). Still. a more interesting comparison is with Web sites that offer access to English corpora (see Table 6 or web addresses):[4]

BNC The British National Corpus is a very large (over 100 million words) corpus ofmodern English. both spoken and written. The BNC site provides access to a subset of the British National Corpus on a trial basis. This permits simple searches on-line. but with limited number of hits. and limited information about the hits. Registration for a trial account (20 days) is required. Full access requires downloading a (Windows) client program (available for Windows95. and Windows3.x only). and payment of an annual registration fee. It is restricted to users within the EC.

Cobuild This site gives limited access to the Cobuild Corpora: the "Bank of English"(over 50million words). The page is intended to provide a ilavour of the kinds of search that can be carried out. It is possible to search for regular expressions (including a special character which matches inflectional endings). combinations of words. and part of speech tags. Only 40 lines of concordance are returned. and no information about frequency. or wider context is accessible. lt is also possible to search for collocates of words. based on either of two statistical scores (mutual information and T-score). ranked by statistical significance (100 collocates are returned by default).
The site does not provide much in the way of help pages. and there is no tutorial.

Bergen Corpus of London Teenager Language At this site it is possible to search a pilot version of the Bergen Corpus of London Teenager Language corpus using the TACTWeb software (TACTWeb is intended to make TACT software usable over the WWW. TACT is a text-analysis and retrieval system for MSDOS that permits inquiries on text databases in European languages).

Canadian Hansard This site permits access to the proceedings of the Canadian Parliament in English and French. These are parallel corpora (English and French). searches may be mono- or bi-lingual (in either case. the results retumed are bi-lingual -i.e. the user sees both the context where the search term appears. and translation):

- with the simple query interface. entering a word or expression in one language will retrieve examples of its use together with the translation of these examples. For instance. typing *passer un sapin* will allow the user to see how this expression is used and how it can be translated.
- With the bilingual query interface. the user can also enter a pair of words or expressions to retrieve examples where one element is translated as the other. For example. entering *commitment* in the English Expression field and *attachement* in the French Expression field will produce examples where one of these words is translated as the other.

Normally. the program searches for expressions verbatim: a query like *pull the plug* will find all occurrences of that string (and none of *pulled the plug* or *pull the plugs*). It is also possible to perform a dictionary search. e.g. the query: *pull+ the plug* will look for *pull the plug*. but also *pulling the plug, pulls the plug*. etc. and searches for words that do not appear contiguously (e.g. *make...arrangements*); there is also a restricted form of ellipsis (indicated as "...") which only spans a few words (25 characters). It is possible to view the wider context of search results. No frequency information is provided.

Swedish Government Site This site gives access to "Regeringsforklaringen": the yearly declaration of the Swedish government issued in Swedish. English. French. German. and Spanish. Simple searches are supported. At the time of writing. this is simply a demonstration program.

Linguistic Data Consortium site gives remote access to the Rrown Corpus (1 Million words of American English). after registration at the main Linguistic Data Consortium site. For individuals who are not (affiliated to) members of the LDC it is possible to register as a guest (and later upgrade to full membership). and access corpora with the password that is supplied; authoiization and password are sent to the user by email. Frequency information is available. and a wide variety of searches is supported. concordances can be generated. and collocational information retrieved.

Conc 1.7	http://www.sil.org/
LEXA	http:!lwww.hit.uib.nolicame/icanie.litml
ParaConc	http://www.ruf.rice.edu/~barlow/parac.html
Wordsmith Tools	http:llwww1.oup.co.uk/elt/catalogu/multimed/
MicroConcord	http::/www.nol.net
IMS Corpus Workbench (Xkwick)	http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/
Micro-OCP	http://www.oup.co.uk/
Multiconcord	http://sun1.bham.ac.uk/johnstf/lingua.htm

Table 5. Some Corpus Processing Tools

British National Corpus (Sara)	http://info.ox.ac.uk/bnc/sara/index.html
Cobuild	http:lltitania.cobuild.collins.co.uWform.html
Bergen Corpus of Teeiiager Language	http://www.hf.uib.no/i/Engelsk/COLT/index.html
Canadian Hansard	http://www-rali.iro.umontreal.ca/TransSearch/TS-simple-ucn.cgi
Swedish Government Site	http://strindberg.ling.uu.se/~corpora/rf/
Linguistic Data Consortium	http://www.ldc.upenn.edu/ldc/catalog/html/text.html (see also: http://www.ldc.upenn.edu/ldc/about/

Table 6. Web Sites Offering Corpus Access

It is obvious that some of these sites provide functionality that is not available at the W3Corpora site. In particular. some provide one or more of: (i) multi-lingual searching and searching over parallel corpora. (ii) access to collocational information. and (iii) 'dictionary style searching' (i.e. the ability to search for inflectional variants of a word -to use *take* as the search term and recover instances of *takes, taken* and *took*. or for a regular verb like *walk*. to recover *walks* and *walked*: the W3Corpora system requires such searches to be simulated using regular eupressions). Several provide access to far more extensive corpus resources than are available via the W3Corpora site.

On the other hand none of these sites duplicates what is available at the W3Corpora site. In particular. none of them provides the balance of easy (imtnediate) access to usable quantities of corpus material. with easy. custornizable functionality. and extensive user support and tutorial facilities.

V. CONCLUSION

In the tnain. the W3Corpora site and search engine satisfy the desiderata that were listed at the outset. It is usable without the need to install or download programs. to register or get authorization. The interface is relatively 'friendly'. and searches return a useful minimum of information (frequency and KWIC. with access to wider context on demand). lt is possible to perform some operations on the results of searches (e.g. 'refinement'). It is possible to install and search user defined corpora. The search engine is supported by extensive help and general documentation (as readers may judge for themselves by visiting the site). One may reasonable claim that this is a good place to start learning about corpus linguistics.

However. there are both minor and major limitations. It is not really possible to use *any* web browser: providing results in a readable fornat requires the use of HTML frames. and some (obsolete) browsers do not do this. The operations that can be applied to search results are limited to 'refinement' of searches: to do anything more. the user has to use standard tricks (e.g. 'drag and drop') to save results to their own machine. The range of searches that can be carried out is not very exciting: tliere are no tools for investigating collocations. for exaniple. There is no provision for languages other than English. Perhaps most seriously. the fact that only public domain corpora can be made freely available means that only limited and rather unsystematic corpora are available. and no access to tagged corpora is possible without registration.

As the WWW matures. as better tools become available for developing web applications. it should be become easier to develop applications that improve on W3Corpora in many ways. Unfortunately. the lack of tagged public domain corpora is a social. not technological. phenomenon. so in this respect the situation is unlikely to improve greatly in the near future. Really getting started in corpus linguistics is going to require some kind of registration to use a non-public domain corpus. and hence a significant commitment on the part of the user.

NOTES

(1) The project was tlie joiiit work of Ylva Berglund. Natalia Brines-Moja, Martiii Rondell and tlie author in the period 1996-8. The project was funded by JISC (tlie Joint Information Systems Comiiiittee of the UK Higher Education Funding Councils). under JTAP (the JISC Technology Application Programme), as part of project JTAP-2/247, which also involved the development of the "Internet Grammar of English" by a team at University College London (cf. Aarts et al. (1999)). Other discussion ofthe project caii be found in Arnold et al. (1999) and Arnold (1999) all available from `http://clwww.essex.ac.uk/~doug/`.

(2) Regular expressions provide a general way of specifying textual pattems. for example ihe regular expression `^[Nn][a-z]*$` will matcli either N or `n` followed by zero or more lower case letters (any letter between a and `z`). The `*` means 'zero or more of', the initial `^` means only match at tlie stan of a line, tlie final `$` nieans only niatch at the end of a line. If a corpus is represented oiie word per line, then this will match words that stait with an upper or lower case `n`. and otherwise coiisist only of lower case letters (i.e. no numbers, capitals. etc).

(3) For iiiore information about the Gutenberg project. see `http://promo.net/pg/`. Only a subset of the Gutenberg collection are available for searching at the W3Corpora site. However. this still gives access to 321 tests. totally around 19.000.000 words.

(4) The information given here is probably out of date. given the speed at which WWW related things move. Most of the systems mentioned are under developnient. which normally nieans that they either improve. or become free or are ported to new platforms. The web accessible sites listed here can all be accessed via the W3Corpora web-pages.

REFERENCES

Aarts. R. et al. *(1999)Internet Grammar of English.* http://www.ucl.ac.uk/english-usage/internet-grammar/.

Arnold. D. *(1999)* "Web access lo corpora: the W3Corpora project". In *Post-Conference Workshop on Computer and Internet Supported Education in Language and Speech Technology.* University of Bergen. Norway. June *1999.* European Association for Computational Linguistics. http://clwww.essex.ac.uk/~doug/papers/eacl99.ps.

Arnold. D. et al.*(1999) Corpora and grammars on the web: the W3Corpora-IGE Project, final report JTAP-2 247.* http://clwww.essex.ac.uk/w3c-ige/FinalReport/, February 1999.

Barnbrook. G. *(1996) Language and Computers. a practical introduction to the computer analysis of language.* Edinburgh: Edinburgh University Press.

Kennedy. G. (1998) *An Introduction to Corpus Linguistics* London: Addison Wesley Longman Ltd.

Kim. E.E. *(1996)CGI Developer's Guide.* Indianapolis. Indiana: Sams.net Publishing.

Wall. L. and R.L. Schwartz *(1991)Programming Perl.* Sebastopol: O'Reilly & Associates. Inc.