The WebCorp Search Engine
A holistic approach to web text search

Antoinette Renouf, Andrew Kehoe and Jay Banerjee
Research and Development Unit for English Studies
University of Central England in Birmingham
{ajrenouf, andrew.kehoe, jbanerjee}@uce.ac.uk

1. Introduction

In this paper, we shall review the development of the 'WebCorp' search tool, demonstrating some of its functionality, going on to identify some of the linguistic and procedural problems that have been encountered and overcome in processing web text online and seeking to present the results at a standard of speed and usability approaching that expected by corpus linguists using conventional corpora. With reference to the less tractable problems we have encountered, in particular those occasioned by our reliance on the Google search engine, we shall explain how they will be overcome by replacing this commercial search engine with our own linguistically tailored web-search architecture.

2. The Web as a source of linguistic information

In the late 1990s, the emergence of the web meant a sea change in the speed, mode and scope of dissemination of information. Vast amounts of data, including text, became available for electronic consultation. There was at the same time a growing need among corpus linguists to find a data source which complemented in various ways the designed, processed and annotated corpora that had become the bread and butter of the field. Linguists sought immediate access to aspects of language which were missing from corpora, in particular the latest coinages, and rare, obsolescent or reviving language. Web text presented a serendipitous solution. While it had many well-rehearsed shortcomings, these were outweighed by the advantages it offered of access to free, plentiful, up-dated and up-to-date data.

A number of corpus linguists attempted to use the commercial Google search engine to find evidence of targeted aspects of language use, and some are still doing so. Google offers many services, but it is not primarily geared to the linguistic or academic user, and for their purposes its output is often not ideal. Meanwhile, other linguists and software engineers have undertaken various initiatives aimed at creating the means to access web text.

Like WebCorp, KWiCFinder (Fletcher, 2001) is a stand-alone web concordance tool that rides on a commercial search engine. It differs in being a Windows-only program which users must download and install on their own PCs. KWiCFinder downloads and stores HTML documents, displaying words in kwic contexts. It supports filtering by page location (e.g. *.uk*) and date, and wildcard matching. The system works relatively quickly but is (by the author's own admission) unstable. It suffers from search engine vagaries, as WebCorp does. Glossanet (Fairon, 2000) downloads data from newspaper sites, creates corpora and applies UNITEX parsing programs and LADL electronic dictionaries and local grammar libraries. Search results are emailed to the user on a drip-feed basis. Glossanet updates the corpus at regular intervals, as and when websites

are modified. It retrieves information (by means of large graph libraries) or looks for given morphological, lexical and syntactic structures. An 'instant' version of Fairon's Glossanet tool offers a reduced service online.

Building specialised corpora based on automated search engine queries has also gained favour amongst scholars. The RDUES unit has been using its own web crawler from 2000 to update its 600 million-word Independent and Guardian corpus. Ghani et al (2001) have created minority language corpora by mining data from the web. Baroni and Bernardini (2004) report on the BootCaT toolkit that iteratively builds a corpus from automated Google search queries using a set of seed terms. Resnik and Elkiss (2003) use a 'query by example' technique to build sentence collections from the web on the basis of lexical and syntactic structure. Results retrieved can be used to build up a user's personal collection. Again, the advantage of the syntactic parsing is limited by the dependence on external search engine and archive site.

3. The Current WebCorp Tool

The purpose of the WebCorp system is to extract supplementary or otherwise unavailable information from web text; to provide a quality of processed and analysed linguistic output similar to that derived from finite corpora; and to try progressively to meet users' expressed needs. In 1998, we developed a simple prototype web search feedback tool, which was made available on our website, to gather user impressions and requirements. In 2000, funding allowed full-scale system development to commence, and the basic tool was expanded to provide a range of functions within the limits imposed by our dependence on commercial search engines (predominantly Google) and the processing capacity of our servers. From the outset, it was clear that fundamental improvements would have to be achieved in both these areas in the long term, and so we established a relationship with the sole UK-based search engine company, searchengine.com, which allowed us to understand search engine technology, as well as gain back-door access to indexes in order to speed up response time. During 2002-3, we added further options to WebCorp, including the sorting of results; the identification of key phrases (Morley, forthcoming); simple POS tagging, diachronic search (Kehoe, forthcoming) and various other filters. In 2004, functionality continued to be expanded, with the design and future assembly of a linguistically-tailored search engine firmly in mind.

WebCorp architecture as it currently stands is represented in the diagram in Figure 1, which also explains the search and analysis routine; the WebCorp user interface is shown in Figure 2.

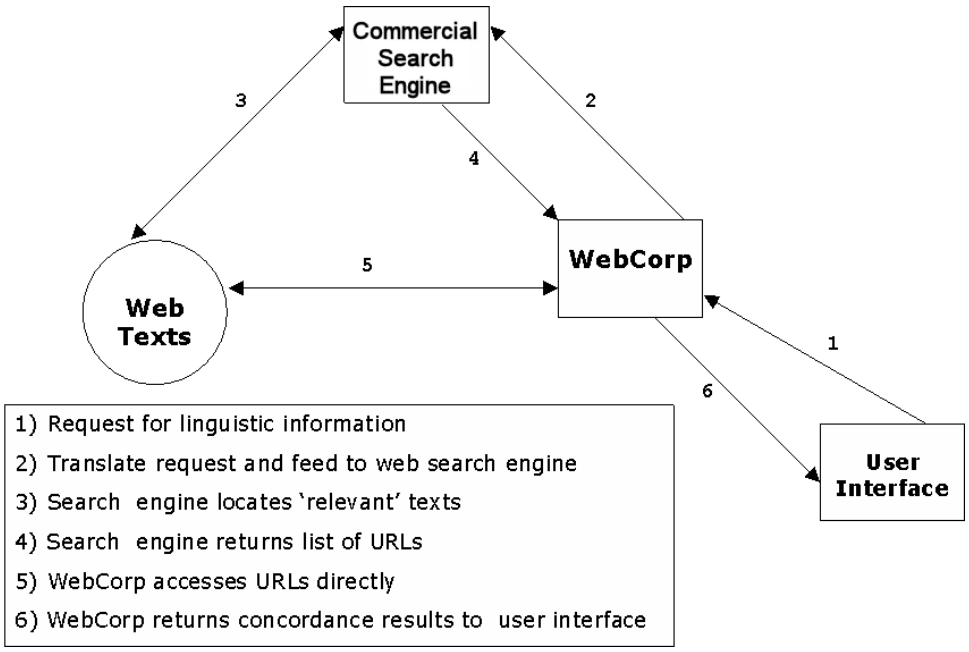

Figure 1: Diagram of current WebCorp architecture

As indicated by the WebCorp user interface depicted in Figure 2, WebCorp currently finds words, phrases and discontinuous patterns through word and wildcard search, allowing various options for filtering of information as well as for output format. It also supports a degree of post-editing, in terms of alphabetical and date sorting, and concordance line removal. Some examples of the types of information WebCorp is able to provide will now be briefly presented, with reference to Figures 3-6 below. These include neologisms and coinages; newly-vogueish terms; rare or possibly obsolete terms; rare or possibly obsolete constructions; phrasal variability and creativity; basic statistical information and basic key phrase analysis.

An instance of a **neologism** which emerged and swiftly became productive in web-based newspaper text in 2004, but one which will not be encountered in designed corpora for some time, is the term '*chav*'. Etymologically indeterminate, but thought to originate from Kentish dialect, it refers to a social underclass of youth which has adopted small-scale status symbols, such as Burberry baseball caps, as fashion accessories.

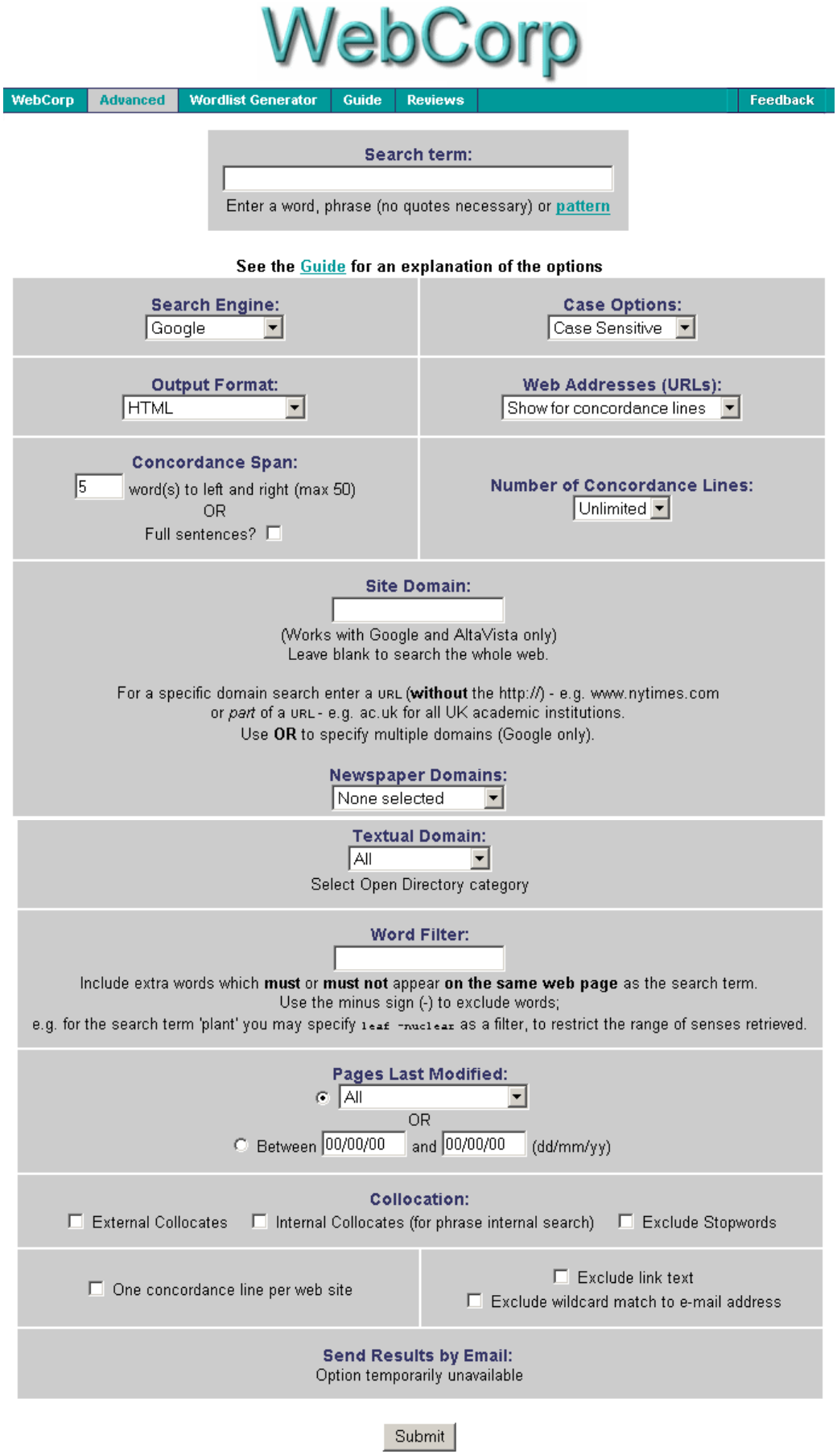

Figure 2: current WebCorp user interface

An extract of the linguistic information derivable from web text with WebCorp is presented in Figure 3, which shows not only that the usage patterns and meaning of the word are provided, but also the tell-tale signs of its assimilation into the language, at least in the short term, in the form of accompanying creative modification of the basic form to produce *chavvish, chavworld, chavdom*, *chav-tastic*, and the phrase *the chavs and the chav-nots* (a play on the phonologically and semantically similar 'haves and have-nots').

1. it as the badge of '**chav**' culture. With such undesirable celebrities
2. held up for our approval **chavvish** artefacts like the Sugababes, Kat
3. ugly and shallow affectations of **chavdom**, I began to claw at
4. listen, babe: you ain't no **chav**. And people who wear tracksuits
5. than the garish immediacy of **chavworld**. People who've read a book
6. but also a defender, championing **chavdom** against boring, moribund middle-class tastefulness
7. discussion about whether the word "**chav**" does come from the name
8. is wrong, argues Burchill - a **chav** is something to celebrate, not
9. Pop Idol, or some anonymous **chav** up before the beak, charged
10. on release day at certain **chav-tastic** catalogue stores. Not to mention
11. it extremely desirable among teenage **chavs**, who spend hours taking
12. in the battle between the **chavs** and the chav-nots, it is

Figure 3: results for search term [*chav**], filter: UK news

An instance of rare or possibly obsolete usage might be the object of curiosity, and an example is the colour term *donkey brown*, which was common in the fifties, but which, like many colour terms, may have disappeared and been replaced by several generations of alternative designations, such as *taupe*, for the referent in question. The output generated by WebCorp is shown in Figure 4. This is useful stuff for the linguist, in that it indicates firstly that the term is not totally obsolete, but only rare, and secondly, that it is used in restricted contexts, where each URL involved refers to a text apparently by an old-fashioned or country-based writer, evoking the old-fashioned, romantic or traditional nature of goods or natural phenomena (coats, trousers, leaves) through the use of old-fashioned colour terms for the materials of which they are made. The alternative interpretation to be investigated through more detailed search remains the possible ironic or parodic use of this anachronism.

1. wide choice of colours is on offer along with the traditional 'natural' colours which shade from off white, through fawns and grey to 'moorit'– a **donkey brown**, and Shetland black – a very dark brown.
2. the soft mix of colours from honey, light grey to **donkey brown** and a textured finish.
3. unisex grey trousers for the two men and two women, and shirts ranging from dark brown through **donkey brown** and dark blue until finally bursting out in a blaze of light grey.
4. One was the usual "**donkey" brown**; the other was a darker hue.
5. Dull green juvenile foliage which becomes **donkey brown** in winter.
6. My **donkey brown** coat which was such a joy when I bought it three years ago, now seems long, thick, hot and dowdy.
7. "The Crafty owd Divil", thought I as I watched him board the bus dressed in a faded jacket of county check, **donkey brown** trousers, and brown brogues

Figure 4: results for search term [*donkey brown*]

An instance of the phrasal variability and creativity which can be investigated with the use of WebCorp is the proverb *a stitch in time saves nine.* This conventional and established idiom can be searched for in its canonical form, but if the linguist suspects that, like all so-called 'frozen expressions', it can actually be modified in use, WebCorp offers the opportunity to test this through the submission of this string with various key words suppressed. Thus in Figure 5, we see the output of variants forced by the use of the word filter option to suppress the word *nine* in the output. What this reveals, among several other interesting facts about phrasal creativity in general, is that one convention of creative modification is that the substituted word may rhyme or be phonologically reminiscent of the original word, as in examples 9 and 10. Whether this is intended to assist interpretation or pay homage to the original phrase probably depends on the creative process and context involved.

1. A **stitch in time saves** embarrassment on the washing line.
2. Like they say, a **stitch in time saves** two in the bush.
3. The best maxim is be vigilant - a **stitch in time saves** a lot of money and inconvenience. Keeping a careful eye on your building will save fortunes
4. follow the adage "a **stitch in time saves** spoilt underwear".
5. A **stitch in time saves** lives. Tenants tipped to share safety training
6. Data Integrity: A **stitch in time saves** your data. Under OS 8.5 and higher Disk First aid automatically launches during startup
7. you know what they say; A **stitch in time saves** disintegration on entering hyperspace.
8. he winds up trying to tie his shambling creation together, just like the Doktor:
9. a **stitch in time saves**, **nein**?
10. Montrose team's **stitch in time saves** **canine**. Search-and-rescue crew rescues former mayor's dog stuck on ledge

Figure 5: results for search pattern [*stitch in time saves*] with *nine* filtered out

WebCorp also provides some basic statistical information, in particular about the 'collocational profile' (Renouf, e.g. 1993) of the word, though this is of necessity currently restricted to simple ranked frequency of occurrence in the set of pages visited. Figure 6 shows 'external collocates' for the phrasal fragment [*stitch in time saves*], since the word slot on which the query is focussed lies outside the pattern submitted (i.e. in position R1). If a search were being conducted on a variable word slot within the pattern, the corresponding 'internal collocate' (Renouf, 2003) analysis could equally be provided. In addition, a simple heuristic (Renouf, ibid.) provides a set of possible key phrases found within the results: in Figure 6, this indicates the more popular alternative phrases emerging in the place of the canonical *a stitch in time saves nine*.

As said, the development of WebCorp has been founded on user feedback. This has continued to flow, and because we have been in a constant state of iterative development and testing, the comments have very often been taken account of in response to an earlier request by the time the same comment reappears.

There are, alongside the extensive functions of WebCorp that have successfully been developed, a range of problems which hinder the further improvement of the system.

Some of these are intrinsic to web text, and include the unorthodox definition of 'text', heterogeneity of web-held data, lack of reliable punctuation**,** lack of reliable information on language, date, author; and the focus on current news and recently updated pages at the expense of access to earlier data.

Top external collocates of "stitch in time saves" (with stopwords)

Word	Total	L4	L3	L2	L1	R1	R2	R3	R4	Left Total	Right Total
time	19	1	1	8		6	1	2		10	9
lives	13					10	1	2		0	13
save	9	1	8							9	0
days	8					8				0	8
game	8		2	2		1	2		1	4	4
Imperial	7					7				0	7
importantly	7								7	0	7
women	5								5	0	5
JOHNSON	5		5							5	0
columnist	5					5				0	5
trouble	5								5	0	5
poor	5							5		0	5
training	5		5							5	0
COLUMN	5			5						5	0
exposure	5						5			0	5
world	5	5								5	0
phrase	4			2				2		2	2

Key Phrases: stitch in time saves lives stitch in time saves days stitch in time saves time

Figure 6: external collocates and key phrases for search pattern [*stitch in time saves*]

Other current WebCorp performance problems relate to the high degree of processing and storage required to meet user needs expressed for simultaneous use for more users, including class-sized groups; grammatical and better collocational analysis; and more sophisticated pattern matching.

However, the major constraint on the improvement of WebCorp performance is its reliance on a commercial search engine. The problems posed by this dependence are as follows: the speed of results is inhibited; there are unpredictable changes in Google service and even at the best of times, Google is geared to commercial rather than linguistic or even academic requirements, which can mean, for example, unreliable word count statistics, and lack of consistent support for wildcard search. Google also uses its own page ranking to deliver the results. The top ranked pages are not necessarily the most relevant ones in view of linguistics. In addition, the delay built in by Google-dependent text extraction means that the time subsequently required for the linguistic post-processing of text is currently prohibitive, whether for POS tagging, for date and alphabetical sorting, or other requisite procedures.

4. The WebCorp Linguistic Search Engine

Our response to the problems anticipated and cited above has been to develop WebCorp with an eye to creating the components that will be integral to an independent,

linguistically tailored search engine. We are currently calling this the 'WebCorp Linguistic Search Engine', since WebCorp functionality will be integrated into the new architecture alongside the search engine, and the whole fronted by an enhanced version of the WebCorp GUI. The new architecture is displayed graphically in Figure 7. The generic term 'linguistic search engine' is in fact a misnomer, since the search engine, while informed by linguistic knowledge, will not be 'linguistic' as such. We sometimes call our embryonic system 'the UCE Search Engine', since our university is the prime investor in the new search engine component, providing both vast storage and ample hardware, as part of its serious commitment to research support in its centres of excellence.

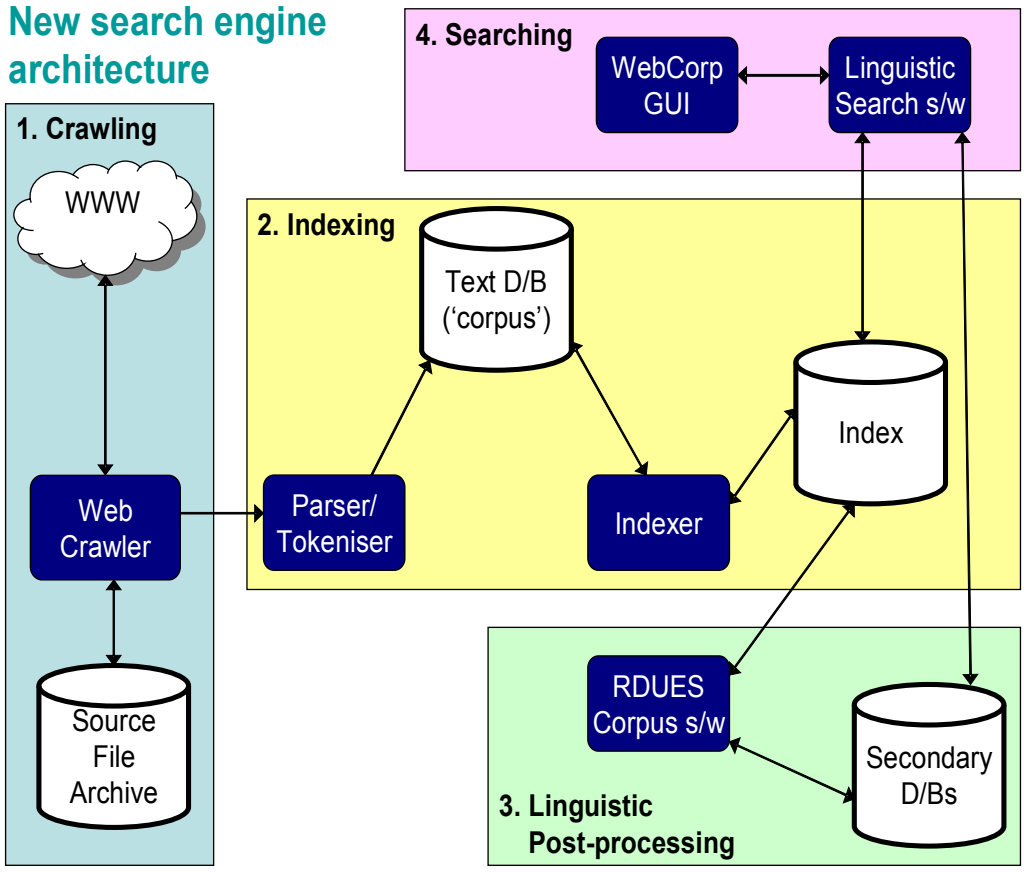

Figure 7: the new WebCorp Linguistic Search Engine architecture

The components of the new linguistic search engine system are as follows:

1. web crawler
2. parser / tokeniser
3. indexer
4. WebCorp tools
5. WebCorp user front end
6. more, also off-line, linguistic processing tools

and we have already developed them individually, as we shall now outline.

1. Web Crawler:

We have already developed a crawler module in Perl to select and download articles from UK newspaper websites. These are currently restricted to the *Guardian* and *Independent* but we shall add to them, with tabloid and other categories of journalism. Not all newspaper sites have full archives like the *Guardian*, so instead of downloading retrospectively, as we have done hitherto, we shall download the current day's articles daily in order to build up the corpus progressively. Our initial estimate is that the newspaper 'domain' accessible through the WebCorp Linguistic Search Engine will contain at least 750 million word tokens.

Our newspaper crawler has been employed for our own use for the past 5 years and incorporates the following:

- exclusion lists (i.e. particular kinds of pages on newspaper sites NOT to download)
- error logging and re-queuing of failed pages
- extraction of date, author, headline and sub-headline
- URL parsing to extract section of newspaper (Sport, Media, etc)
- storage of articles by date (to facilitate diachronic analysis)
- removal of advertising banners and links
- stripping of HTML mark-up

We shall continue to use these tailored crawlers for our newspaper 'domain' and for other domains where all pages are in a uniform format. We also have a specialised tool to extract neologisms from online articles in real-time. We shall expand this 'live' system to monitor and record neologisms, although once the web texts are downloaded into corpus format, we will begin to achieve this through the application of our full APRIL system [http://rdues.uce.ac.uk/april.shtml], as we have begun to do with *Guardian* articles more recently.

In addition to our structured sub-domains, we shall download a very large (multi-terabyte) subset of random texts from the web, to create a mini version of the web itself. Some users will prefer to look at this, much as they do with WebCorp at present, rather than at carefully chosen sub-domains. The aim will not in itself be to build either specific sub-corpora or 'collections' of texts from the web, as other people have done (e.g. BootCaT tools), but to find the right balance and combination of raw data, for instance in selecting random texts within a specific domain.

More generic tools will be required for the creation of this multi-terabyte mini-web, to cope with a variety of page layouts and formats. Several ready-made tools are available freely online but we are developing a new crawler for our specific task, building upon our experience with the newspaper downloads and making use of other open-source libraries whenever possible.

The new crawler will need to be 'seeded' in some way, i.e. told where to embark on its crawl of the web. We could make the search process completely random by choosing a starting page and allowing the crawler to follow all links blindly, downloading every page it encounters. This will not be appropriate, however, when building a structured corpus with carefully selected sub-domains.

We shall employ other 'seeding' techniques including the use of Open Directory index, where editors classify web pages according to textual 'domain'. Our crawler will make

use of the freely downloadable Open Directory 'RDF dumps' (http://rdf.dmoz.org/), containing lists of URLs classified by domain (or 'category'). We shall also consult human experts, university colleagues from our own and other disciplines, current WebCorp users and other contributors, to catalogue the major target websites in their field, so that these can be used to seed the crawler.

There are a number of issues which we still have to resolve in relation to web crawler design. One is the depth of crawl that should be undertaken. If the crawler starts at the BBC homepage, for example, and follows the links to other pages, at a 'crawl depth' of 1, a decision has to be made as to whether it follows the links on those other pages, reaching a crawl depth of 2, and so on to further pages and depths. Another issue relates to the internal and external links in web text. If the crawler starts at the BBC homepage, a decision has to be made about whether it only follows internal links, staying within the BBC site, or also follows external links to other sites. This is important when building specific sub-corpora where content control is required.

New features of the crawler which remain to be developed include better duplicate detection: methods of comparing newly-encountered pages with those already stored in the repository to identify both updated and mirror versions of pages. We are also determined to improve on the date detection mechanism we have already created. Knowledge as to when our crawler first encountered a page may provide a clue as to when it was created, and the discovery of a new version of a page already stored will reveal when it was updated. The existence of our own independent search engine will allow us to conduct date detection off-line, not in real-time as at present. We shall also be able to classify changes and updates from the linguistic perspective, by scrutinising the page for changes in actual content rather than simply in mark-up or layout. Another area on which we have done considerable work, but which we should still like to improve on, is language detection, which could be done by the crawler or at the indexing stage.

2. Indexing

The source files will be stored in our standard RDUES format and then processed using specially adapted versions of the parsing, tokenising and indexing software which we have developed over the past 15 years, and run on texts downloaded from newspaper websites for the past 5 years. This will construct the corpus as a series of binary files and indexes. Our past experience indicates that we will be able to store 10 billion word tokens per terabyte of disk storage, including the processed corpus, indexes, raw HTML files (the 'source file archive' in Figure 7) and the secondary databases resulting from the linguistic post-processing stage outlined below.

Corpus updates will be incremental. New articles will be added to the newspaper domain daily, while other domains and the large mini-web 'chunk' will be updated at monthly intervals. Corpus processing will take place off-line and the new version of the corpus will 'go live' when processing is complete.

3. Linguistic post-processing

We shall be able to run on web texts any of the gamut of tools we can run on our current newspaper corpus. Where necessary, we shall also develop new tools, to provide a comprehensive range of corpus-processing functions. A priority is to exploit the tools created in major projects over the last 15 years, including those which generate

collocates, 'nyms' (alternative search terms in the form of sense-related items), neologisms, summaries, document similarity measures, domain identification and so on. The sharing of these specialist language facilities will be a matter of individual negotiation: we shall be looking for relevant collaborative research proposals from potential users.

4. Searching

We shall develop new user interfaces, building upon our experience with WebCorp and other tools, such as the step-by-step and advanced APRIL neologisms demos [http://rdues.uce.ac.uk/aprdemo], taking into account user feedback, and so on.

All results will be stored in the secondary databases shown in the Figure 7 diagram of system architecture, and there will be new linguistic search software created to access the secondary databases.

5. Features and benefits of the new tailored web-search architecture

5.1 Increased speed

The system will now function as quickly as Google, but will be able to offer more functionality from a linguistic perspective. In terms of enhanced text quality, there will be a far greater rate of accuracy in respect of duplicate detection, sentence identification and full-text search. Text specification will be significantly improved with regard to domain detection, better date information for diachronic study and reliable language identification. Text search routines will be made more sophisticated with regard to specific domain search and specific URL sets.

5.2 Improved statistics

The web data will no longer be a vast, unquantifiable sea from which the system plucks an amount of data that cannot be evaluated in terms of its significance. The sub-web, or rather webs, which are regularly downloaded will be known entities, and thus reliable statistical counts and measures will be possible – in particular, the current WebCorp limitation to simple frequency counts will cease, and calculation of relative frequency and significance of phenomena such as collocation will commence.

5.3 Improved search

Many and varied will be the improvements to search. These will include wildcard-initial search; wildcard matching for a variable number of intra-pattern search words up to a maximum span; POS specification, and lexico-grammatical specification.

6. Indicative Output from the WebCorp Linguistic Search Engine

There follow some invented examples of the more complex and comprehensive linguistic and statistical analyses that we shall provide for the user once the WebCorp Linguistic Search Engine is up and running, and the post-processing operation will no longer be prohibitively time-consuming. The first two concern refined wildcard pattern search.

6.1 Wildcard-initial words as search terms

Google does not consistently support wildcard pattern search, but when it does, it does not allow wildcard-initial words as search terms. Our search engine will provide such information, as shown in the invented output for the term "**gate*" in Figure 8. In addition, it will continue to be possible to specify textual domain (here 'UK broadsheets'), context length (here 'sentences'), and require the presence of particular words within the text (here the term *scandal*) to improve precision.

1. If Janet Jackson's next album is great, we will look back on the Super Bowl incident as a stroke of PR genius; if it's a flop, we'll regard **Nipplegate** as the fatal blow
2. The investigation began from the force's Fettes headquarters in Edinburgh, and has been dubbed **Fetishgate** by the Daily Record news
3. That's what's truly rotten about **Svengate**: while Ms Dell'Olio cannot complain if her man no longer wants her, she has every right to be devastated by him wooing his conquests.
4. The New York Times thought **Rathergate** a bigger story than American hostage beheading.
5. It's been washed away in the bittersweet tide of **Blunkettgate**
6. On my travels, I've spotted a **pre-Scousegate**, **pre-lovergate** Boris Johnson balanced precariously atop his two-wheeler
7. Make up scandals for next year: **Williamgate**, **Top-upgate**, **Saddamgate**

Figure 8: mock-up of results for wildcard-initial search pattern "**gate*"

6.2 Variable number of words in wildcard position

For a search allowing for variation in number of words in the NP, we shall be able to provide a pattern search wildcard which allows for a specified maximum number of words. For instance, in a study of the 'It-cleft' construction, "*it was *(3) which*", the *(3) would be a specification of all words in the wildcard position up to a maximum of 3. This would allow a search to yield results such as those shown in Figure 9.

[*it*+*BE*+mod+N+*which*]
1 wd
1. it was that which stuck in everyone's memory
2. it was heroin which crocked Iggy
3. it was Londis which approached Musgrave with the takeover plan
4. it was Iraq which caught IDS in the tightest bind
5. it was telecoms which ruled the roost

2 wds
6. it was the Bank which delivered the coup de grace
7. it was the NHS which failed
8. it was these qualities which gave them victory over St Helens
9. it was her boots which really caught the eye
10. it was his research which embarrassed Michael Howard earlier this year

3 wds
11. it was a driver coach which struck the cow
12. It was the Tawney Society which staged the Holme-Thomas debate
13. it was the Renoir-esque films which were the more interesting
14. It was this apparent contradiction which confused some delegates
15. it was those unwise borrowings which cost the company its independence

[*it*+*BE*+mod+Npl+*which*]
16. it was Singapore trades which brought down Barings Bank
17. It is those countries which oscillate which have the worst rates
18. In all my books, it is the emotions which start the story
19. it is these contrasts which define England's uniqueness
20. it was their forces which had provoked the conflict

[*it*+*was*+possA+N+*which*]
21. It was her power which won the day
22. It was his modesty which needed protection

23. it was my bedroom which was small
24. It was our discipline which held us in good stead
25. It was their split which prompted McKellen to come out

Figure 9: mock-up of results for pattern [*it was *(3) which*], max 3 words in wildcard position

In addition, once we have established a sub-web processing system, we shall be able to provide lexico-grammatical search of the kind indicated by the search in Figure 9, where a combination of actual lexical realisations and grammatical categories may be specified.

As said, the new search engine will allow us to bolt on some of our past automated systems of linguistic analysis. One is the ACRONYM (Renouf, 1996) system of automatic identification of Wordnet-type sense-related synonyms or alternative search terms. Figure 10 shows the start of the ranked output it produces from our *Independent*/*Guardian* database for the term *cheated*.

conned	bewildered
cheat	tricked
cheating	robbed
short-changed	deceived
betrayed	nagged
duped	dissatisfied
abused	victimised
lied	upset
insulted	harassed
wronged	disillusioned
intimidated	misled

Figure 10: results from ACRONYM system for search term *cheated*

Perhaps more uplifting are the ranked 'nyms' for the search term *advantages* in Figure 11!

synonyms		**contrasts**
benefits	rewards	disadvantages
advantage	improvements	drawbacks
flexibility	strengths	disadvantage
benefit	incentive	risks
opportunities	synergies	difficulties
incentives	leverage	
potential	importance	
attractions	possibilities	

Figure 11: results from ACRONYM system for search term *advantages*

We also intend to append the APRIL (Renouf et al, forthcoming) project morphological analyser to the new system. This will allow users to search the web for the morphological analysis of target words, as well as to view plots of the target word or words across time. Figure 11 presents an extract of morphological output of the kind that will be available with the new system.

word	parse	tag	month
- untitivated	(unti) -ive -ate -ed	|JJ|	198912
- unintellectualised	un- (intellectualise) -ed	|JJ|	199004

- unbrochurised	un- (brochure) -ise -ed	|JJ|	199009
- unkeepered	un- (keeper) -ed	|JJ|	199009
- unironised	un- (irony) -s -ed	|JJ|	199101
- unwigged	un- (wig) -ed	|JJ|	199107
- unevangelised	un- (evangelise) -ed	|JJ|	199107
- unnovelised	un- (novel) -ise -ed	|JJ|	199107
- uncomplexed	un- (complex) -ed	|JJ|	199201
- untexted	un- (text) -ed	|JJ|	199211
- uninventoried	un- (inventory) -ed	|JJ|	199304
- unhistoried	un- (history) -ed	|JJ|	199304
- uncreolised	un- (creole) -ise -ed	|JJ|	199402
- un-Gothamed	un- '-' (Gotham) -ed	|JJ|	199712
- unroaded	un- (road) -ed	|JJ|	199810
- un-Sheened	un- '-' (sheen) -ed	|JJ|	200112
- un-chadored	un- '-' (chador) -ed	|JJ|	200212

Figure 12: results from APRIL system **-** new adjectives with prefix *un-*, suffix *-ed*

7. Concluding remarks

In view of the frustration and limitations posed by the current search engines, other researchers are also beginning to contemplate building their own search engine software and tools. The WaCky Project (2005) is still at the ideas stage, as is Kilgarriff (2003). Kilgarriff (2003) proposes a five-year project for a system similar to ours, where a set of URLs relevant to linguists would be downloaded, processed off-line and stored as a corpus for linguistic research. He plans less frequent updating than we do within our differentiated update schedule. There is also some mention of future Grid interaction in his design. We embrace the cooperative spirit that is implicit in the Grid ideal, but are not dependent on the distributing processing element of Grid activity, being more than adequately resourced with regard to computing storage and hardware.

We have completed the components required for the creation of a linguistically-tailored and accessorised search engine, and shall in the coming months assemble an infrastructure that will be progressively incorporated into the **WebCorp** front-end to enhance its performance, and that of its users, on the fronts outlined above. The improvements will be incrementally perceptible at http://www.webcorp.org.uk/

References

Baroni, M. and Bernardini, S. (2004) BootCaT: Bootstrapping corpora and terms from the web, in *Proceedings of LREC 2004*, Lisbon: ELDA, 1313-1316.

Fairon, C. (2000) GlossaNet: Parsing a web site as a corpus, *Linguisticae Investigationes*, October 2000, vol. 22, no. 2, pp. 327-340(14). (Amsterdam: John Benjamins).

Fletcher, W. (2001) Concordancing the Web with KWiCFinder, in *Proceedings of The American Association for Applied Corpus Linguistics Third North American*

Symposium on Corpus Linguistics and Language Teaching. Available online from http://www.kwicfinder.com.

Ghani, R., Jones, R. and Mladenic, D. (2001) Mining the web to create minority language corpora. *CIKM 2001*, 279–286.

Kehoe, A. (forthcoming) Diachronic linguistic analysis on the web with WebCorp, in A. Renouf and A. Kehoe (eds.) *The Changing Face of Corpus Linguistics* (Amsterdam & Atlanta: Rodopi).

Kehoe, A. and Renouf, A. (2002) WebCorp: Applying the Web to Linguistics and Linguistics to the Web. *World Wide Web 2002 Conference, Honolulu, Hawaii*, 7-11 May 2002. http://www2002.org/CDROM/poster/67/

Kilgarriff, A. (2003) Linguistic Search Engine. *Proceedings of The Shallow Processing of Large Corpora Workshop (SProLaC 2003) Corpus Linguistics 2003*, Lancaster University.

Morley, B. (forthcoming) WebCorp: A tool for online linguistic information retrieval and analysis, in A. Renouf and A. Kehoe (eds.) *The Changing Face of Corpus Linguistics* (Amsterdam & Atlanta: Rodopi).

Renouf, A., Pacey, M., Kehoe, A. and Davies, P. (forthcoming), *Monitoring Lexical Innovation in Journalistic Text Across Time*.

Renouf, A., Morley, B. and Kehoe, A. (2003) Linguistic Research with the XML/RDF aware WebCorp Tool. *WWW2003*, *Budapest*. http://www2003.org/cdrom/papers/poster/p005/p5-morley.html.

Renouf, A. (2002) WebCorp: providing a renewable data source for corpus linguists, in S. Granger and S. Petch-Tyson (eds.) *Extending the scope of corpus-based research: new applications, new challenges*. (Amsterdam & Atlanta: Rodopi) 39-58.

Renouf, A. (1996) The ACRONYM Project: Discovering the Textual Thesaurus, in I. Lancashire, C. Meyer and C. Percy (eds.) *Papers from English Language Research on Computerized Corpora (ICAME 16)* (Rodopi, Amsterdam) 171-187.

Renouf, A. (1993) Making Sense of Text: Automated Approaches to Meaning Extraction. *Proceedings of 17th International Online Information Meeting, 7-9 Dec 1993*. pp. 77-86.

Resnik, P. and Elkiss, A. (2003). The Linguist's Search Engine: Getting Started Guide. *Technical Report: LAMP-TR-108/CS-TR-4541/UMIACS-TR-2003-109*, University of Maryland, College Park, November 2003.

The WaCky Project (2005) http://wacky.sslmit.unibo.it/

WebCorp (1998 - ongoing) http://www.webcorp.org.uk/