Making the Web More Useful as a Source for Linguistic Corpora

William H. Fletcher
United States Naval Academy

To appear in (expected 2004):

Connor, U. and T. Upton (editors), *Corpus Linguistics in North America 2002: Selections from the Fourth North American Symposium of the American Association for Applied Corpus Linguistics.* Amsterdam: Rodopi.

Current version online at http://kwicfinder.com/AAACL2002whf.pdf

---

Abstract

Both as a corpus and as a source of texts for corpora the Web offers significant benefits in its virtually comprehensive coverage of major languages, content domains and written text types, yet its usefulness is limited by the generally unknown origin and reliability of online texts and by the sheer amount of "noise" on the Web. This paper describes and evaluates linguistic methods and computing tools to identify representative documents efficiently. To test these methods, a pilot corpus of 11,201 online documents in English was compiled. "Noise filtering" techniques based on n-grams helped eliminate both virtually identical and highly repetitive documents. Individual review of the remaining unique texts revealed that Web pages under 5 KB or over 200 KB tend to have a lower "signal to noise" ratio and therefore can be excluded *a priori* to reduce unproductive downloads. This paper also compares a selection of these web texts (4949 documents totaling 5.25 million tokens) with the written texts from the British National Corpus (BNC) to assess their similarity. Generally both corpora are quite similar, but important differences are outlined. With judicious selection Web pages provide representative language samples, often prove more useful than off-the-shelf corpora for special information needs, and complement and verify data from traditional corpora.

1. Web as Corpus

The World Wide Web has much promise as a source of machine-readable text for corpora. Over ten billion publicly-accessible online documents provide comprehensive coverage of the major languages and language varieties, and span virtually all content domains and written text types. Throughout the developed world the Web is readily accessible at low cost and has become a familiar information source for hundreds of millions of users. As a self-renewing linguistic resource it offers a freshness and topicality unmatched by fixed corpora; emerging usage and current issues are generally well represented in online texts.

When analyzing relatively rare features of a language, the Web is an inexhaustible resource. With appropriate tools it is simple to compile an ad-hoc corpus from online documents to answer a specific language question or meet a specialized information need. The following example illustrates convincingly that bigger can be better when it comes to corpora. In January 2003 a discussion thread on the CLLT Listserv (2003) focused on the phrase "not as ADJECTIVE as you think". In the Michigan Corpus of Academic Spoken English (MICASE, 1.7 million words) only two occurrences were found, and even the 100-million-word British National Corpus World Edition (BNC) yielded only 77 examples. In contrast, the AltaVista search engine reports over 66,328 Web pages with "as * as you think" and 41,189 with "as * as you * think", where the first wildcard * almost always matches an adjective or an adverb and the

second one typically matches *(woul)d, may* or *might.* In about 40 minutes, the Web concordancing search agent application *KWiCFinder*[1] downloaded and analyzed 500 Web pages, ample material for a thorough analysis.

Unfortunately one must be cautious when using online texts as linguistic data. Web pages are typically anonymous and Web server location is no certain guide to origin, so it is difficult to establish authorship and provenance and to assess the reliability, representativeness and authoritativeness of texts, both for their linguistic form and their content. Multilingual sites are common, as are English pages authored by non-native speakers of varying competence, raising questions about language quality and influence of the source language. Among the longer prose texts certain types predominate, especially legal, journalistic, commercial and academic prose, a much narrower cross-section of language usage than one might require. Overall, lower standards of form and content verification prevail than in printed sources. Web pages often contain a significant amount of "noise", i.e. language which is fragmentary, repetitive, formulaic, or ill-formed, and often entire documents which have no cohesive text.

A significant limitation on the Web is that systematic access to linguistic data online can only gained through full-text searches on commercial search engines. Designed for the general public, most search engines do not support targeted search criteria such as sophisticated pattern matching which would make them most useful to linguists. Among the search engines AltaVista offers the most powerful combination of features, but its database has often languished months without updating, and its unstable financial position raises doubts about its future. Even more unfortunately for researchers, AltaVista's reports of the number of documents matching a given query are inconsistent and can vary up to an order of magnitude during peak usage times; consequently they provide only a general numeric indication of the prevalence of a form, not statistically reliable proof. Perhaps the greatest weakness in contrast to most corpora is that the Web has no grammatical markup, so one can only match for strings, not specific structures.

Elsewhere I discuss in greater detail the benefits and challenges of exploiting the Web as a corpus for both pure and applied linguistic research and propose a solution to the limitations imposed by commercial search engines (Fletcher 2001, 2002). This paper concentrates on efforts to make the World Wide Web more useful as a source for corpus compilation by developing and evaluating linguistic methods and PC tools to identify linguistically representative documents more efficiently. My long-range goal is to establish the Web both as a "corpus of first resort" and as a supplement to traditionally compiled corpora.

2. Compiling a Web Corpus

2.1 Objectives and Preliminary Considerations

In seven years of developing and using KWiCFinder I have viewed excerpts from over a quarter million online documents and have examined thousands as complete Web pages. My cumulative impression has convinced me that the Web can yield linguistic data which are both useful and reliable. To confirm this conviction I compiled a pilot corpus with KWiCFinder of Web documents in English for analysis offline in October 2001. These sample documents totaling 5.5 million tokens allowed me to gauge how suitable and representative such texts could be for research or learning and to evaluate techniques to identify Web pages with a high proportion of connected text. My goal was to analyze language samples from the Web, not to investigate the language of the Web in general.

A major objective of this study was to develop procedures and software tools[2] to automate or expedite identification of the most useful texts. Some steps toward optimizing a search can be taken at the outset

[1] KWiCFinder, the author's Key Word in Context Web Concordancer, automates finding, analyzing, and saving online documents matching specific search criteria. It is described in detail in Fletcher (2001), and can be downloaded free from http://kwicfinder.com/.

[2] All programs were developed for Windows with PowerBasic, which is comparable to C in speed, power and compactness. My intention is to offer tools with a familiar graphical user interface for the most widespread desktop operating system so that colleagues and students need not become proficient UNIX users to do corpus research. I gratefully acknowledge my substantial debt to the PowerBasic user forums for peer support and sample code.

when formulating the query, by choosing selection criteria which either exclude a range of texts or favor inclusion of more relevant results. For example, by excluding documents with "copyright" or "all rights reserved" one can filter out many commercial and journalistic texts without excluding most academic, government and personal material.

Another simple indicator of potential usefulness is document size: a query to the server can reveal how large a Web page is before the search agent "decides" to download it. With guidelines for rejecting a page before fetching it[3] because it is relatively unlikely to contain useful text, search agent software can save both bandwidth and processing time. Web documents typically contain significant chunks of "noise": headers and footers that identify the document, declare who owns it and explicitly reserves rights to it; links both within the document and to other documents, media and sites (especially advertisers); and other formulaic elements; I will refer to these as "boilerplate". Unfortunately HTML provides no standard way to distinguish such boilerplate elements from the unique textual content of each page. Without insight into the structure of a Web page, a search agent has no criteria for extracting content while excluding formal elements.[4] Obviously, the shorter a Web page is, the lower its "signal to noise" ratio as well, and the less likely it will be to contain more than a few sentences of connected text; practical guidelines for a lower cutoff point are needed. At the other end of the spectrum, the very largest Web pages tend to consist of lists and fragments: server logs and statistics, indexes, glossaries, discussion group messages and headers, and "linketeria" pages. Such Web pages can be enormous–up to several megabytes–while documents of that length consisting primarily of connected prose are exceedingly rare. Since downloading long documents consumes significant bandwidth, guidelines for an upper size limit would be useful as well.

2.2 Collecting Web Pages as Corpus Data

Before compiling a sizable Web corpus I examined a sample of 100 Web pages retrieved and saved to local text files by KWiCFinder for the query *"the OR of OR a"*. As formulated this matches any document in English containing any of these three very high-frequency words almost certain to occur in an extended text. This search yielded primarily documents from commercial sites: *all rights reserved* was the most frequent 3-gram, occurring 43 times in 100 texts, and *copyright* #### fell among the top ten 2-grams.

In a second follow-up sampling I ran a series of queries for the ten highest-frequency words in the BNC. Among the 5859 documents these searches yielded were 2277 or 39% duplicates.[5] Early in 2001 AltaVista had instituted preferential treatment for paying advertisers, placing "sponsored links" prominently at the beginning of search results and updating its database only for links to its subscribers.[6] For exclusion from future searches I determined both which hosts (Web sites) were "overrepresented" in the results (presumably appearing higher within the search results due to sponsorship) and which had yielded the "noisiest" documents.

---

[3] An application can obtain information from a Web server about the size and date of a file before downloading it. While search engines report file size, changes to an online resource often make their data unreliable.

[4] Many websites do use custom templates with comments or element tags which allow one to find page elements like headers, footers, advertisements and contents automatically. While useful for analyzing numerous documents from a single site, parsing heuristics are rarely transferable from one site to another.

[5] For the first sample of 100 pages a single KWiCFinder search was run, so duplicates occurred only when two URLs pointed to the identical document. Since KWiCFinder uses the AltaVista search engine to find matching documents, it cannot go beyond the latter's 1000-document limit per query. Consequently it must "merge" data from multiple searches to find larger numbers of texts.

[6] AltaVista's serious deficiencies in updating its database and distinguishing sponsored links were resolved in 2002. With these factors out of play, AltaVista tends to provide a more random sampling of Web pages than Google. Each site's ranking algorithms are closely-guarded secrets subject to constant revision. Generally speaking, however, the former tends to rank a page higher in the search results based on formal criteria indicating relative salience of the search terms within the document, while the latter additionally weights results by "link popularity", i.e. the number of sites that link to a given Web page. Google's strategy favors relevance and reliability—which is why it quickly became the most popular search engine—but also skews results toward fewer, more prominent sites, often those run for business purposes.

Finally I conducted a third round of searches. My search terms were the twenty-one highest-frequency words in the BNC, supplemented by the underlined forms *the, of, to, and, a | an, in, is | are | be | was | were | been, that, for, it, on, with, as, he, she, by, I, at, not.* The requirement for each search was that it include at least one article and one form of the copula BE, on the assumption that any sizable chunk of prose will contain these words often lacking in fragmentary texts. To reduce the commercial bias of the sample, these searches were limited to documents last indexed by AltaVista before 1 January 2001; any clients who paid for preferential placement in search results would have been updated since then. In addition, the overbearing and noisy hosts identified in the second sampling were explicitly excluded. This third iteration yielded 11,201 documents and serves as the basis of the analysis below.

2.3 Reducing the "Noise" in the Data

Before analysis of the downloaded documents, four principal "noise-reduction" tasks were completed with a suite of Windows programs I developed.[7] These procedures help filter out repetitive and fragmentary documents so they do not bloat the corpus and skew the linguistic data.

2.3.1 Filtering Out Duplicate Identical Documents

First, duplicate identical documents (IDs) had to be identified and removed. It is common for a given document to have more than one URL[8] or to be "mirrored" on multiple sites (e.g. Rivest 1992 appears verbatim on over 22,000 sites), so duplicates cannot be avoided simply by comparing URLs. The documents had been saved by KWiCFinder in text format, i.e. all HTML tags had been stripped and HTML entities had been converted to characters. The challenge was to compare over 11,000 files totaling almost 70 MB *(after* removing HTML markup). The solution is relatively simple as it reuses portions of programs I had developed for other purposes. For an n-gram extractor I had already developed routines to normalize a text and to build a binary tree of representations of each n-gram for efficient comparison. To reduce memory requirements to a bare minimum my approach took advantage of the Message Digest 5 Secure Hash Algorithm (MD5 SHA), a 16-byte representation or

> 'fingerprint'… of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest… (Rivest 1992)

In other words, with the MD5 SHA a text of any length can be captured in a code string only 16 characters long which has an extremely high probability of uniqueness; in practice, only two identical texts will produce the same code. Each text in the binary tree requires only 24 bytes (16 bytes for the MD code and 8 bytes for pointers to the next nodes in the tree), so both storage requirements and the number of characters involved in each comparison are minimal.[9] My program *FindDuplicates* reads and normalizes each downloaded document, then reduces it to an MD5 hash and compares this code to hashes of previously analyzed documents following a binary tree algorithm. If it is unique, the hash code is added to the tree and the document is retained; otherwise the document is moved to a directory of discarded files.

---

[7] Some of this software is already available at http://kwicfinder.com, and other modules will be released when integration and documentation is complete.

[8] For example, the home page of the departmental website I administer is accessible via either *http://www.nadn.navy.mil/LangStudy/* or *http://www.usna.edu/LangStudy/,* followed or not by *homepage.html*. All four URLs point to the same document, but some appear redundantly in search engine results.

[9] Prior to settling on MD5 I evaluated numerous hashing algorithms (approaches to "digesting" a string into a short code) for uniqueness of results and speed. In tests with 20 million unique strings I found that hashes of four bytes or less resulted in numerous "collisions", i.e. different strings result in the same code. In comparison with SHA-1 and RIPEMD160 (both 20-byte hashes, i.e. 4 bytes longer), MD5 encoded faster over a greater range of string lengths while providing similar protection against collisions. (RIPEMD160 was only marginally slower, but SHA-1 took up to twice as long to encode; for a 67 kB file the range was 1-2 ms. This is not an absolute claim, as tests on different machines showed that the distribution and relative order of run times varies significantly depending on system configuration.) Those who work with much larger datasets where reducing memory load is critical might follow up Dillon's (no date) suggestion that CRC-64 (8 bytes, i.e. half the size of MD5) is sufficient, as it theoretically would lead to a collision only once in 2.3 trillion (10^{12}) times.

Encoding and comparison of all 11,201 files took only 33 seconds,[10] leaving 7294 unique documents. Since this search was limited to documents last indexed almost a year earlier, the most common duplicate texts were variants of the infamous "404 – File Not Found" error message, and the second most frequent were warnings that the site requires frames.

2.3.2 Identifying Virtually Identical Documents

Remaining among the unique files were a number of "virtually identical documents" (VIDs). These include multiple instances of the same text with only slight differences, such as news stories from the wire services appearing on several sites, mirrored Web pages with different footers, various pages from the same site in which boilerplate material predominates over unique content, and instances of the same URL with dynamically updated content (time of day, temperature etc.). FindDuplicates cannot help here, since even the slightest difference between normalized texts yields highly dissimilar MD codes. While I could not automate recognition of VIDs, I used n-grams to identify potential VIDs for visual comparison.

Here *n-gram* is used in the sense of "sequence of *n* words", and *word* is defined orthographically as "a string of alphanumeric characters preceded and followed by whitespace, punctuation or nothing".[11] Normalization converts alphabetic characters to lower case, strips punctuation except word-internal period, hyphen or apostrophe and the symbol ©, and maps numeric characters onto # so that *copyright 1997* and *copyright 2001* are both tallied as instances of *copyright ####*.

The method I explored for recognizing VIDs rests on two assumptions: after normalization, two or more VIDs will be of approximately the same size, and the identical content will be far more extensive than the surrounding boilerplate material. My program *ViewVIDs* cycles through all the document files in descending order of file size. For each file it looks for any smaller files whose size differs by no more than 5% or 1000 bytes, whichever is greater. If so, lists of the most frequent 3-grams occurring two or more times in each file are made. If 20 of the top 25 3-grams in two documents of comparable size agree in form and frequency, a tentative match is made, and both texts are presented to the user for comparison in side-by-side text windows. If no tentative matches are made, the program continues on down the list. With this approach, 26 VIDs were identified and dropped.[12] Several consisted of extensive boilerplate material with minimal unique content. In the most extreme example, three VIDs shared a footer almost 6000 bytes long![13]

2.3.3 Finding Highly Repetitive Documents

While IDs and VIDs incorporate significant amounts of text from **other** Web pages, "highly repetitive documents" (HRDs) repeat substantial chunks of content within the **same** document. To locate HRDs I tabulated the frequency of longer n-grams in the entire Web corpus for values of *n* equal to 20, 12 and 8 and kept lists of those found 5 or more times. While the most common shorter n-grams (say for $n \leq 4$) are typically found in a wide variety of texts and contexts, these longer n-grams are highly specific and invariably derive from a single source, such as a title, instruction, formulaic expression, quotation—or

---

[10] Approximate run times are based on a Pentium IV / 2.4 GHz / 512 MB desktop under Windows XP. Thanks to the binary tree comparison algorithm and the memory typical of today's systems performance would not degrade substantially for much larger document collections.

[11] Frequency-ordered lists of n-grams in each document were produced by my program *nGram,* for which the MD5 / binary tree algorithm was developed. Since this approach proved incapable of handling the far greater volume of material in the BNC, I subsequently programmed *kfNgram,* which adapts the far more efficient *Virtual Corpus* algorithm described by Kit and Wilks (1998) and offers a GUI.

[12] This approach is a first attempt to address the problem of VIDs which requires further testing and refinement. It relies on working assumptions about efficient but effective parameters to identify VIDs. The method does not work for very short texts, since few 3-grams if any are repeated; the distribution of 2-grams or even 1-grams may be more useful. On the other hand, for relatively long texts, patterns of 4-grams are more distinctive. The optimal relationship of file size to "window" size, i.e. the range of sizes of other documents to which a given file should be compared, also deserves study.

[13] Many online documents incorporate large chunks of superfluous text as "search-engine spam" in hopes of increasing traffic by matching more queries.

simple repetition of the same sequence of words. Any single text with several instances of a longer n-gram is a potential HRD, but the recurring text may be insignificant in a large document. To determine the nature and prevalence of the redundancies I developed and used *FindHRDs,* which searches each file for instances of frequent longer n-grams and displays any matching passages for assessment and possible elimination.

Overall 256 documents were deemed highly repetitive, and many others showed some degree of repetition; some remaining VIDs were identified as well. Shorter elements such as links often recur after each section of a document, or Web pages derived from books or articles may repeat titles as headings at regular intervals. Software documentation and programming tutorials may include the same long sequence of characters again and again. In transcripts of legal and legislative proceedings, repetition of formulaic elements is common, as is the verbatim reappearance of entire passages in laws and contracts. Generally such repetition was deemed minimal, so the documents were retained. Undoubtedly the most tedious HRDs are server logs, followed by forum threads where each response incorporates all preceding posts to the same discussion. Ironically, search-engine ranking algorithms favor such mindless echoes, since they make a search term very prominent within a document. I typically discarded such "fugues on a theme" without a second thought.

2.3.4 Unproven "Noise" Filters

Other techniques to automate filtering out "noisy" Web pages were investigated but proved less effective without further refinement. The Spelling and Grammar Checker engines of the Microsoft Office suite can be controlled programmatically. These modules could help automate recognition and normalization of ill-formed documents. The primary obstacle encountered was the large number of items not in the default lexicon, such as personal, commercial and place names, technological terms and other neologisms, and abbreviations. Consequently these tools either require constant user intervention when used interactively or else reject too many good documents in automatic mode. Still, they deserve further consideration, particularly for use in a well-defined content domain for which a custom dictionary could be compiled.

Presumably frequency patterns of 1- and 2-grams could indicate "primarily fragmentary documents" (PFDs) such as link lists, server logs and pages bloated with search-engine spam. Some types with high frequency in connected prose like articles, copula, and pronouns are rare in fragments, while others, such as common prepositions, are frequent. Content words, proper nouns and jargon are also relatively prominent in PFDs. In this investigation I did not succeed in exploiting these observations, but do intend to return to this technique in the future.

2.3.5 Separating Connected Prose from Fragmentary Texts

After sifting out duplicate and repetitive documents with computer assistance, I viewed each of the 7038 survivors briefly, and classified them as predominately useful text, "noisy" text, i.e. with a significant level of "overhead", and PFDs; the latter two categories were excluded from the final corpus. The principle of "predominance" was vaguely defined, and since I reviewed up to 12 documents per minute, no rigorous consistency is claimed. This cursory examination of the documents disqualified roughly 30% of the pages, leaving 4949 documents totaling 5,248,929 tokens and 34,995,762 bytes. Each document included was allowed a "reasonable amount" of overhead for its size—headers, footers, links, bibliography, lists, non-English words—but not exceeding 20% of content for very short documents, dwindling to about 5% maximum for longer ones.

During this visual dash through the Web pages I could not savor their content, but it did leave distinct impressions on me. Since I typically conduct narrowly-defined searches with criteria conceived to limit results to a single content domain, I was struck by the variety of material matching these general queries. Among the shorter documents—those of a few hundred to a couple of thousand words—commercial and personal text prevailed. At the other end of the scale—up to 60,000 words—legal texts and government proceedings were well represented. The middle range was filled with academic texts—papers, theses, syllabi and course materials, some computer hardware and programming documentation, other expository prose, drama (including Shakespeare) and fiction, and personal interest pages, as well as a surprising

number of religious documents and commentaries in the Christian, Islamic and Hindu traditions. Numerous "hobbyist" pages broadened the range of topics as well. As expected, it was this middle range that yielded the most useful texts.

2.3.6 File Size as Indication of Usefulness

As anticipated, the shortest and longest documents bore the brunt of this visual selection. Half of all documents were under 3330 bytes long, and of these about 40% were rejected. Only 10 documents were longer than 100 KB, and more than half of these were deemed primarily non-textual; in fact, no documents over 200 KB were retained. In the range of 5-100 KB, I judged over three-quarters of the documents to be primarily connected prose. The optimum size seems to fall around 50 KB, where only 17.8% of documents were rejected. Nevertheless, owing to the far greater number of smaller files, the median size of texts retained was only 3770 bytes!

Which HTML files are most worth downloading? Due to variations in HTML markup, the size of a file only indicates roughly how much text it contains. Some HTML editors (most notoriously Microsoft Word) grossly inflate file size, often to 5-10 times that of generic HTML with the same content, and embedded stylesheets and scripts add bulk, but not textual content. Stripping out such formatting elements typically reduces files to 40-65% of the HTML size; here again shorter files have greater overhead. This signal-to-noise ratio and the observations in the previous paragraph suggest the following rule of thumb: to maximize the "yield" of connected prose, download HTML files only between 10 and 150 KB in size.[14]

Had KWiCFinder followed these guidelines for this study, only one-third of the final number of files would have been downloaded, but that would have yielded a corpus two-thirds of the size of the current one with enormous savings in bandwidth and analysis time. The capability to exclude files *below* a given size is now on my KWiCFinder "to do" list (currently only a *maximum* file size can be specified).

Other researchers have sampled Web pages as a source of corpus data with other techniques to ensure that samples consisted primarily of running prose. Cavaglià and Kilgarriff (2001) use statistical methods to compare the rank frequencies of lexical items in individual Web pages to those in the BNC. This comparison requires a sample size of at least 2000 words per page, so briefer documents were rejected. This cut-off point would exclude about 90% of all Web pages in my sample. In a study for the American National Corpus (ANC) Ide et al. (2002) arrived at minimums of 2000 words and 30 paragraphs per document as a reasonable indicator of primarily connected text. They report that only 1-2% of Web pages investigated satisfied both criteria. To increase the likelihood of reaching this 2000-word threshold, one would have to raise the rule-of-thumb for the minimum size of HTML files to download to about 25 KB. In doing so, one would exclude many typical Web pages which consist primarily of prose. Good Web style requires breaking up long documents into shorter Web pages for quicker loading and more responsive hyperactivity.

3. Comparing this Web Corpus to the British National Corpus

My experience with KWiCFinder has convinced me that the Web is a reliable source of data when studying specific words or phrases. How representative of English is this Web Corpus? As a first step toward answering that question I compared lexical data from this corpus to the BNC. The 4949 Web documents which survived the various "filters" and selection processes were combined into a single file with 5,382,595 tokens (approximately 1/16 of the size of the BNC written corpus). To obtain comparable data from the BNC, I extracted all text within <body> tags in the BNC data files, stripped SGML tags including grammatical markup, and mapped SGML entities to the corresponding characters. Spaces

[14] Applying these size guidelines to all 7038 documents remaining after discarding IDs, VIDs and HRDs, 4,724 of them would not have been downloaded. On average the documents eliminated by this rule of thumb were 509 words long, and those retained had a mean size of 1985 words. On the other hand, 23% of the documents kept by this rule were dropped after visual review, so the suggested size range is only a modest indicator of usefulness.

around orthographic word-interior hyphen and apostrophe were removed. The resulting text data were amalgamated into nine large data files with 87,221,955 tokens total for further processing.[15]

Frequency lists of 1-, 2-, 3-, 4-, 5-, 25-, and 50-grams in the two corpora were produced with kfNgram. Relevant options chosen were: not case-sensitive, preserve word-interior hyphens and apostrophes, replace numerals with #, floor 50. Standard kfNgram character remapping was chosen, so boundaries between sentences, paragraphs and even entire texts were ignored on the reasonable assumption that the random "pseudo-n-grams" resulting from this expediency would fall below the relatively high threshold chosen.

The 5000 most frequent alphabetic n-grams for each value of *n* were then imported into a Microsoft Access database for further analysis. Three sets of queries yielded the following record sets[16]:

1. N-grams with a rank frequency of 1 to 250 in both corpora
2. N-grams with a rank frequency of 1 to 200 in one corpus and greater than 300 (i.e. relatively less frequent) in the other
3. N-grams with a rank frequency of 1 to 500 in one corpus not among the 5000 most frequent in the other.

A thorough analysis of the similarities and differences between the two corpora is beyond the scope of this paper, but will be the subject of a future study. Here I limit myself to preliminary observations about salient differences. Rank frequency lists of the 50 most common words in both corpora are quite similar, but some striking contrasts are found. Beyond these most frequent items the divergences become both greater and more numerous, and thus more indicative of the medium. The tables below detail all important differences for the top 50 word forms, and sample differences from those ranked 51-200 in frequency. Since these are frequency ranks, *lower* numbers reflect *higher* frequencies.

Word forms far more frequent in BNC by frequency rank			
Rank list	Word form	BNC	Web
1-50	*he*	23	39
	his	23	44
	she	33	155
	her	34	130
51-200	*Mr*	123	371
	man	146	414
	old	153	319
	thought	160	729
	never	155	331
	came	184	566
	rather	189	499

Word forms far more frequent in Web by frequency rank			
Rank list	Word form	BNC	Web
1-50	*you*	28	15
	will	41	27
	we	43	28

[15] No attempt was made to normalize spelling. Systematic differences between British and American orthography such as *-ize / -ise, -er / -re, -or / -our,* as well as national and personal tendencies to write compound forms with a hyphen, a space, or together—*log-in, log in, login*—can separate lexical variants, thus obscuring important patterns of similarity between the predominantly British BNC texts and the American-biased Web documents.

[16] Extensive excerpts from the datasets are available online at http://kwicfinder.com/WebCorpus/AAACL2002_ngramdata.pdf. The complete database is available upon request.

	information	206	45
	our	100	46
51-200	*site*	1054	67
	page	1011	70
	university	586	114
	data	490	120
	search	1367	135
	please	924	184
	file	1773	186

Inspection of these word form data and of the distribution of the most frequent phrases (n-grams) in the two corpora reveals the biases and gaps in each. The BNC clearly reflects British institutions, place names and spelling, while the Web sample is more oriented toward the United States. The BNC data show a distinct tendency toward third person, past tense, and narrative style, while the Web corpus prefers first (especially *we)* and second person, present and future tense, and interactive style. Since the BNC texts were all written before the mid-nineties, words referring to Internet concepts and information technology which permeate the Web texts (and contemporary life) are rare or missing. In the BNC texts the language of news and politics stands out, while in the Web corpus academic concepts are quite salient. Finally, the Web data are more varied: none of the most common 5000 words in the BNC is lacking in the Web corpus, yet the reverse is not true, despite the sixteen-fold greater sample size.

4. Conclusions and future plans

This paper has surveyed a number of techniques and algorithms for downloading, preprocessing and evaluating texts from the Web for inclusion in a corpus. Windows software to accomplish these tasks is (in some cases will be) freely available from my Web site so that readers can try it out—and help improve it. For comparability with the BNC I aimed to compile a domain-neutral sample Web corpus. Many colleagues will find these procedures especially beneficial for creating small- to medium-sized corpora from the Web for specific professional or pedagogical purposes, or to provide a corpus on a desktop machine for a language for which no corpora are currently available. With the programming done, it should take no more than two or three days' work to produce another corpus of similar size. I hope to have demonstrated that such a project would be both worthwhile and feasible for a motivated linguist or student.

The continuation of this project will lead me down several complementary paths. Currently I am working on a Web interface for an expanded version of this Web corpus as a prototype for the linguistic search engine outlined elsewhere (Fletcher 2002). Techniques and software developed will be disseminated so colleagues can share any Web corpora they do compile. Next I plan to complete a more sophisticated statistical analysis comparing with the BNC (and the ANC when it becomes available) to help dispel doubts about the representativeness of selected Web documents for English as whole. Finally I will investigate further refinements of the procedures and tools described here. Major goals will be to add grammatical markup to the texts and to extend my methods to morphologically richer languages like German and Spanish.

References

BNC Consortium (2000), *British National Corpus World Edition.* Oxford: Humanities Computing Unit. (2 CD-ROMs) http://www.hcu.ox.ac.uk/BNC

Cavaglià, G. and A. Kilgarriff (2001). 'Corpora from the Web'. Fourth Annual CLUCK Colloquium, Sheffield, UK, January 2001.
ftp://ftp.itri.bton.ac.uk/reports/ITRI-01-11.pdf

CLLT (2003), Discussion thread 'That/it is not as ADJECTIVE as you think' on the Corpus Linguistics and Language Teaching Listserv, January 2003.

http://listserv.linguistlist.org/cgi-bin/wa?A1=ind0301&L=cllt

Dillon, Matt (no date), 'CRC1—CRC64 test results on 18.2M dataset'.
http://apollo.backplane.com/matt/crc64.html

Fletcher, W. H. (2001), 'Concordancing the Web with KWiCFinder',
American Association for Applied Corpus Linguistics Third North American Symposium on Corpus Linguistics and Language Teaching, Boston, MA,
23-25 March 2001. http://kwicfinder.com/FletcherCLLT2001.pdf

Fletcher, W. H. (2002), 'Facilitating Compilation and Dissemination of Ad-Hoc Web Corpora', TaLC (Teaching and Language Corpora) 5, Bertinoro, Italy, 26-31 July 2002.
http://kwicfinder.com/Facilitating_Compilation_and_Dissemination_of_Ad-Hoc_Web_Corpora.pdf

Ide, N., R. Reppen and K. Suderman (2002), 'The American National Corpus: More Than the Web Can Provide'. *Proceedings of the Third Language Resources and Evaluation Conference (LREC),* Las Palmas, Canary Islands, Spain, 839-44. http://www.cs.vassar.edu/~ide/papers/anc-lrec02.pdf

Kit, C. and Y. Wilks (1998), 'The *Virtual Corpus* approach to deriving n-gram statistics from large scale corpora', in C. N. Huang (ed.), *Proceedings of the International Conference on Chinese Information Processing,* Beijing, 223-229. http://personal.cityu.edu.hk/~ctckit/papers/vc.pdf

Reppen, R. (2002), 'The American National Corpus Project: A Resource for Applied Linguists'. Fourth North American Symposium on Corpus Linguistics, Indianapolis, 1-3 November 2002.

Rivest, R. (1992), 'The MD5 Message-Digest Algorithm'. RFC1321 (Internet Request for Comments 1321), Cambridge, MA: Network Working Group, MIT Laboratory for Computer Science.
http://www.faqs.org/rfcs/rfc1321.html and numerous other mirror sites.