

Towards an Independent Search Engine for Linguists: Issues and Solutions

*“La Rete come Corpus”
Forlì 14 January 2005*

*William H. Fletcher
United States Naval Academy
(2004-05 Radboud University of Nijmegen)
<http://pie.usna.edu>
<http://kwicfinder.com>*

Objectives of Presentation

- Describe my background and biases
- Survey applications of Web as / for Corpus (WaC)
- Discuss central role of Search Engines for WaC
- Summarize limitations of current SEs for WaC
- Outline essential and desirable features for target groups envisioned
- Sketch a path toward an SE for WaC
- Draw on your expertise and expectations so that a Corpus SE fully meets unanticipated needs

Background and Biases

- Erstwhile linguist
- Language teacher and webmaster
 - Multimedia in CALL – emphasis on user (interface)
 - KWICFinder to...
 - Identify useful texts
 - Find examples of actual use for teaching and writing
 - Clarify linguistic questions
 - Explore emerging semantic fields
- Many issues discussed in TaLC 5 paper ([click here](#))

Web as / for Corpus – Now – Concordancers 1

WebCorp

- Interface to various SEs
- Server-side generation of concordances
 - + No special software required
 - + *Can* be faster than dial-up
 - Not scalable – often slow due to server load
 - Limited support for foreign languages

Web as / for Corpus – Now – Concordancers 2

Linguist's Search Engine

- Searches not just for *word forms* but also for *structures* (trees based on Charniak's parser)
- Server-side generation of concordances
 - + No special software required
 - + *Very fast* (limited use, powerful machine)
 - + Impressive demonstration using authentic data from Web to investigate syntactic structures
 - + Powerful interface for editing trees (*daunting to casual user*)
 - + User can save datasets and even upload own data
 - Limited target audience – theoretical linguists
 - Password protection scares off casual user
 - No support for languages other than English
 - Grant is over, future uncertain (*“development not research”*)

Web as / for Corpus – Now – Concordancers 3

KWiCFinder

- Concordancing search agent
 - + Retrieves and analyzes webpages in background
 - + Various enhancements to goof-proof and focus queries and to filter documents
 - + Optimized for Western European languages (*knowledge about charsets, input special chars...*)
 - + Accessible to casual user while supporting sophisticated queries
 - + Produces stand-alone interactive concordances
 - + Options enable both macro- and micro-level study & evaluation
 - Relies on existing search engine
- Client-side generation of concordances
 - + Not dependent on server load
 - + Significant control over display (*user can cycle among several views*)
 - Requires Windows-only software download with (automatic) updates to address SE changes
 - Like alternatives, no instant gratification

Web as / for Corpus – Now – Concordancers 4

Lexware Culler

- Google “snippets”: runs query on Google and extracts the brief document excerpts from search results
- Server-side processing
 - + Fast – only Google search results page downloaded
 - + Smart
 - supports Part of Speech variables / filters (English, Swedish)
 - automatically generates “tamecards” / SmartMatch i.e. variant forms for highly inflected languages (*Polish, more to come*)
 - + Reports Google document frequency and other statistics
- Co-text (up to 20 words) may be too brief for user’s purposes
- Subject to all of Google’s biases and limitations
- Still experimental demo, with limited access and functionality

Web as / for Corpus – Now – Other

- End-user oriented WebLEAP

“Web Language Evaluation Assistant Program”

- user inputs a sentence or phrase
- WebLEAP queries a SE and displays the frequencies in Web documents of word sequences from the phrase
- Color-coding helps user estimate phrase’s likely acceptability

Search Engines in Web as / for Corpus 1

What do search engines (SEs) do?

■ Crawling

- retrieve documents from “seed” sites and store
- extract links from documents and crawl targets etc.

■ Indexing

- map character sequences in each document for efficient retrieval

■ Query

- find documents matching user query
- prioritize according to SE’s secret formula
- create output page to display results

Search Engines in Web as / for Corpus 2

Approaches to querying:

Classic AltaVista Geek-Seek supported...

- “Unlimited” query length
- Complex Boolean queries with nested bracketing
- Wildcards -- * stands for 0-5 characters, so *parl** matches *parlo, parlai...*
- Distinction upper / lower case, accented / plain character
- Search results ordered by query-term salience
- Proximity operators *NEAR, BEFORE* AFTER* + distance** (*undocumented features supported by AV)

Search Engines in Web as / for Corpus 3

Approaches to querying:
superstar Google...

SE

- Supports none of the features listed for AV Classic
- Ranks results by link popularity, which favors...
 - + appearance of a relevant link among the top search results
 - popular commercial sites
- Is used daily by all of us who decry its limitations
- Google's success and innovation has made searching the Web a more pleasant and effective experience for most users

Search Engines in Web as / for Corpus 4

Google phenomenon: why are *users* so happy, and why aren't *we*?

- + useful results on first try for average user: *they* search for base forms of nouns, *linguists* search for function words, structures, variant forms
- no support for wildcards, case sensitivity, accented chars (English bias?)
- No complex queries with bracketing (*rarely used feature, and most frequently used incorrectly on sites that supported it*)
- results skewed toward commercial sites with many incoming links

Alternative to commercial SEs: build your own specialized SE

SE for WaC General Features – *Essential*

- Full index and match of *all* characters
 - Either exact or fuzzy (disregard case and / or accents)
 - Query with “restrained wildcards” and regular expressions in any position
- Complex queries with nested bracketing and full set of Boolean operators
- Specific (*i.e. position, distance*) proximity operators
- Support for all popular document formats
- Archive original documents for verification of larger context

SE for WaC Features – *Desirable* 1

- Match punctuation, position in sentence and / or paragraph
- General “tamecards”, e.g. *on-line* to match *on-line, on line, online*
- Filter out low-quality documents: VIDs, HRDs, boilerplate and other non-coherent text
- Report *total* matches as well as *document* matches

SE for WaC Features – *Desirable 2*

- Language-specific knowledge:
 - match orthographic variants e.g. Ger. *schön / schoen, dass / daß*
 - query by lemma and / or match specific classes of forms (e.g. by tense, person, case)
- Linguistic markup for query by structure
 - POS, morphological class and syntactic groups
 - Sentence-level syntax

- Initially use off-the-shelf SE software like Nutch / Lucene for 1-2 languages to compile web corpus of 500M-1B words
 - crawls “seeded” by KF and PIE queries
 - webpages selectively fetched, tagged and archived
 - searchable by word form, lemma, POS...
 - “pass-through” – unsuccessful PIE queries handled by SEWaC to extend corpus database

SEWaC adaptable to *any* language

- Nutch / Lucene open-source SE software supports Unicode
- \$1000 Linux machine supports low-traffic site
- (group of) experts responsible for each language

Towards a SE for WaC

Reactions encouraged:
Let us know the needs and wishes
of *all* potential target audiences

<http://kwicfinder.com>

<http://pie.usna.edu>

fletcher@usna.edu