

Toward cleaner Web corpora: recognizing and repairing problems with hybrid online documents

Corpus Linguistics 2007
Birmingham 27-30 July

William H. Fletcher
United States Naval Academy
<http://pie.usna.edu>
<http://kwicfinder.com>
<http://webascopus.org>

Objectives of Presentation

- Describe Web as / for Corpus applications which motivate this paper
- Familiarize you with character set encoding issues which lead to inconsistent Web data
- Outline techniques for recognizing and standardizing Web data
- Discuss and suggest solutions for other sources of “noise” in Web data
- Elicit your feedback on alternative approaches to these issues

Web as / for Corpus Applications - 1

KWiCFinder

- Concordancing search agent
 - + Retrieves and analyzes webpages in background
 - + Produces stand-alone interactive concordances
- Client-side generation of concordances
 - + Not dependent on server load
 - + Significant control over display (*user can cycle among several views*)

Web as / for Corpus Applications - 2

kfWebsuite

Tools to compile corpora from the Web one step at a time

- kfSearch queries Live Search and saves matching webpages
- kfHTML2Text strips HTML markup and (attempts to) standardize character encoding
- kfwinnow filters out documents that do not meet certain criteria and creates composite files for ensuing steps
 - duplicates
 - too many / too few total words
 - too long / too short average paragraph length
- kfReviewFiles displays files winnowed out in previous step for possible retention or excerpction
- kfNgram* normalizes texts and generates *n*-grams (*wordgrams*, *chagramms*) and phrase-frames from them
- kfNgramDB creates and manages text and *n*-gram databases

*freely downloadable; others on request.

Web as / for Corpus Applications - 3

WebAsCorpus.org - 1

Web as Corpus (search Web directly)

- queries MS Live Search and creates KWIC display in “real time”
- reports documents statistics (wordcount, paragraph length...)
- supports 34 languages
- pages from user searches cached for Web Corpus
- KWICFinder-like interactive results planned
- Limitations
 - exact match only
 - max. 1000 matches (*can be extended*)

Web as / for Corpus Applications - 4

WebAsCorpus.org - 2

Web Corpus (corpus compiled from the Web)

- wildcard searchable (RegExp planned)
- click on matching word-forms to fetch concordances from Web
- filters to find innovative (or spurious) forms

Web as / for Corpus Applications - 5

WebAsCorpus.org - 3

Web Corpus 2006

- compiled with KWICFinder searches (AltaVista SE) on most frequent words in BNC, plus terms from UCREL USAS to ensure semantic and stylistic diversity
- after “purging”: 38 K texts, 1 M types, 104 M tokens (140 MW “dirty”)
- online database inconsistent: original texts and much derived data lost in HD crash

Web as / for Corpus Applications - 6

WebAsCorpus.org - 4

Web Corpus 2007

- under active development; details subject to change
- currently 540 MW online; 1.2 GW downloaded
- compiled with Live Search
- seeded with combinations of terms from UCREL USAS as well as user search terms from my Phrases in English site and KWiCFinder

Web as / for Corpus Applications - 7

WebAsCorpus.org - 5

Web Corpus 2007 *(cont)*

Weighted geographical representation

2x proportional to population for non-US pages:

- AU 10%
- CA 13%
- IE 2% (i.e. 20 MW)
- NZ 2%
- UK 30%
- US 43%
- 100 pages total per search-term pair + 15% in PDF format
- oversampling to ensure sufficient “keepers” in each category

Web as / for Corpus Applications - 8

WebAsCorpus.org - 6

Web Corpus 2007 *(cont)*

- will support “BNC-compare-able” search by combination of word-form and PoS tag to supplement PIE *(massively parallel tagging)*
- may be large enough to enable study of usage by country / region *(NA, ANZ...)*
- self-renewing via user searches and regular updates, but snapshots will be preserved for the sake of replicability

Common Goals for these Purposes

- primarily connected text, without fragments, boilerplate, duplicates or repetitious documents
- mapping of variants onto single form to keep databases manageable and to allow patterns to emerge
 - no case distinctions
 - numerals mapped onto #
 - punctuation eliminated
 - **exception:** original spelling retained

Challenges and solutions - 1

- Recognizing (portions of) webpages in which coherent text predominates
 - Min. / max. doc size and para length
 - $500 < \text{docwords} < 50,000$
 - $14 < \text{parawords} < 500$
- Sample for database starting with 1st para 14 words or long, ending with last ditto
- Reject or reserve for evaluation documents with low text yield
(HTML file 20-100 times larger than textfile!)

Challenges and solutions - 2

- Unspecified or inconsistent character set encoding
- Mistaken language identification
- Language (or garbage) islands
- Truly multilingual documents

Character set encoding issues - 1

Hundreds of character encoding schemes, i.e. mappings of numeric codes onto specific characters, exist for various combinations of hardware platforms, languages and applications.

- Venerable 7-bit ASCII supports 128 chars
 - 0-31 – control chars, most rarely used
 - 32-127 – printable chars, no special chars
- 8-bit (single byte) schemes add 128 char positions
 - Mac, IBM OEM, Windows all differ in mappings of codes onto specific characters, both in Western European and other “codepages”
 - http default ISO-8859-1 (Latin 1) leaves 128-159 unused
 - Windows-1252 is a superset, filling many vacant positions with special punctuation, currency and other useful symbols “„” `’ ... – € f ™ ‡ %o
 - Many webpages use ANSI / Win-1252 whether declared or not, a practice supported by most browsers

Character set encoding issues - 2

- 16-bit (double-byte) Unicode provides 65K positions, enough for all languages, **but** requires two bytes per char even for “plain lower” ASCII
- current versions of both Windows and Macintosh use Unicode for internal representation of text in memory and CPU, but store files in other proprietary formats
- UTF-8...
 - provides more compact encoding of Roman chars and punctuation
 - represents all Unicode chars unambiguously
 - used by standards-setters: SEs, Wikipedia, XML

Character set encoding issues - 3

So what?

Fancy puncs!

Why is everyone making him out to be such a saint? the poor young man trying to get his brother out of debt by smuggling drugs? How about thisâ€why didnâ€t he relive his brothers debt the old-fashioned way, and get a job. What makes me sick is the outpouring of sympathy for someone who was peddling heroin that was going to addict and kill young people around the country. One less drug dealer on the street. Sooner the better Im sick of hearing about drug smuggling scum like him on the news.

Special chars!

- Fÿgung = Fügung
- Schšn = Schön
- MontrÈal = Montréal
- M%odchen = Mädchen

Character set encoding issues - 4

- WWW standard allows for character set encoding declaration...
 - by server
 - by document
- but ISO-8859-1 is default if unspecified
- WC06 43% unspecified (WC07 only 25%)
- many servers re-encode to UTF-8
- must know / infer original encoding to map HTML entities and decode properly
- collaborative pages – blogs, forums – often contain hybrid encodings due to different users / sources (e.g. copy + paste)

Character set encoding issues - 5

- Must recognize whether...
 - charset declaration or default is correct
 - encoding is consistent
- Charset “sniffing” algorithms assume consistency
 - if not 100% compliant UTF-8, it’s not UTF-8
 - UTF-8 fingerprint allows recognition and conversion of “islands” at chargroup level
 - convert to declared charset else default
 - don’t trust declaration – verify at word level

Character set encoding issues - 6

- Demo + solution
- Since Mac_in_Win and Win_in_Mac both contain either camelCase or non-appropriate upper-ASCII word-internally..
 - scan text for these symptoms
 - reconvert if necessary
 - verify (likely) correctness
 - will / may not work for non-Western European encodings!

Language Identification Issues – 1

- SEs typically sample only first 1-10k bytes for language identification using char- n -grams and tell-tale words
- Graphically similar languages may not be distinguished by SEs
 - search for Dutch pages also returns Afrikaans, Frisian, Low German, even Scandinavian
 - no major SE distinguishes Afrikaans
 - domain insufficient
 - there are NL pages in the ZA domain, and AF pages in NL and BE
 - both languages found in NET and COM domains

Language Identification Issues – 2

- Language identification with char- n -grams is computationally expensive and unnecessary to distinguish NL / AF
- For NL / AF both tell-tale high-frequency lexical items and language-specific chagrams (e.g. AF *sk, y*; NL *[s]ch, z*) were used (*details upon request*)
- After tweaking (*Stellenbosch*) 99% correct
 - failures: AF-NL glossary and NL-FR bilingual page
 - most pages clear-cut decision

Extraneous Islands

- Other languages, e.g. EN in AF and NL
- Digital garbage
- Peek into the HTML House of Horrors
 - SE spam HTML
 - <http://webascorpus.org/en230159376145.html>
 - <http://webascorpus.org/en230159376145.txt>

You don't see the SE spam text because it's in a 1px x 1px <DIV>. It consists of likely misspellings of fairly common words.

- HTML entity "abuse" http://webascorpus.org/Armenian_logo.html

no charset is declared; Armenian characters are encoded as HTML entities with very different functions

Toward cleaner Web corpora

Suggestions and tales of your own experiences encouraged!

<http://kwicfinder.com>

<http://pie.usna.edu>

<http://webascorpus.org>

fletcher@usna.edu