



# **Implementing a BNC-*Compare*-able Web Corpus *Plans and Progress***

William H. Fletcher, U. S. Naval Academy  
Web as Corpus 3, 15-16 September 2007  
CENTAL, UCL, Louvain-la-Neuve, Belgium

# Goals of presentation

- outline my ongoing English Web as / for Corpus project
- describe progress and pitfalls
- familiarize you with specific resources (to be) used
- elicit feedback on alternative solutions and approaches
- stimulate discussion of various general issues

# WebAsCorpus.org - 1

Realware per September 2007

- Web Concordancer: real-time KWIC concordances in 31 languages using MS Live Search
- Wildcard-searchable / filterable 1-6-grams from two large English Web corpora
  - 2006: 950,087 types / 97,198,272 tokens (*texts lost*)
  - 2007 (now): 3,123,996 types / 518,129,710 tokens
- Webhit Counter for sets of words or phrases ("Googleology-enabler")
- Dutch (102,770 typ. / 1,605,346 tok.) and Afrikaans (62,785 typ. / 1,263,509 tok.) 1-gram frequency lists as down-payment on a Dutch Web corpus (*initially 180 MW*)

# WebAsCorpus.org - 2

## Planned capability for Web Corpus 2007

- Minimum 1 gigawords of English
- representative: geographic, semantic, filetype (HTML + PDF)
- PoS-tagging comparable to BNC
  - CLAWS4 tagger, mapped onto BNC tagsets
  - post-tagging cleanup using UCREL / BNC templates
  - search by lemma or word form
- seamless integration with PIE
- query with wildcards & regular expressions
- filterable – show  $n$ -grams not in other datasets
- growing / self-renewing via actual user queries
- archiving of each release for replicability

# Specific goals

- explore English beyond the BNC – recent and emerging usage, broader geographic representation, include more of the “long tail” victims of Zipf’s Law
- prototype a Windows-based acquisition and processing system extensible to other languages
  - use open-source software where possible
  - Produce sharable apps / code
- deploy on shared LAMP host
  - unrestricted access yet inexpensive
  - learn to work around provider’s policies

# Representativeness

– 1

## geographic concept

- “weighted” proportional distro of major English-speaking nations (*non-US 2x actual population proportion, with 10% oversampling for enough “keepers”*)
  - AU 10%
  - CA 13%
  - IE 2%
  - NZ 2%
  - UK 30%
  - US 43%
- reality: most country-specific hits – official sites or include geo. refs.
- consequently 1/2 documents fetched with no country specified for broader range of sources

# Representativeness

– 2

## semantic concepts

- for *breadth* selected terms from all semantic fields in UCREL's USAS, e.g.
  - A1.1.2 Damaging and destroying, general / abstract terms depicting / damage / destruction / demolition / pollution, etc
  - Prototypical examples: *armageddon, blemish, breakages, bulldoze, contaminating, crack...*
- for *interest* use ST from PIE and KF queries (*content words + phrases*)
- for future growth, archive pages matching WaC queries

# Representativeness

– 3

## **filetype**

- 90% HTML – general content
- 10% PDF
  - higher-quality text
  - specific genres of interest
    - scholarly papers
    - print media
    - government documents

# MS Live Search

- Powerful API supports
  - search by country
  - weighting result set by popularity, freshness and / or exactness of match, effectively many times 1000 hits
- License permits 10,000 queries per IP address (not total per license)
- Cache
  - HEAD returns doc size for pre-weeding
  - fetch typically much quicker than original
  - formats PDF usefully
  - sniffs charset and converts to UTF-8

# Acquisition & processing – Plan A

concept – highly distributed parallel processing on 10-20 PCs

- $\pm 8000$  seed ST assigned to worker PCs, which independently
  - query & fetch hits from LS
  - strip HTML, determine “keepers”
    - non-dupes, size
    - verify text-ness and English-ness
  - PoS-tag text
  - when all done, create  $n$ -grams of (un)tagged text
- worker PCs communicate with servers only to...
  - avoid previously processed or rejected documents
  - upload texts and  $n$ -grams for final databases

# Acquisition & processing – Plan B

- Seed STs assigned manually to 3 worker PCs, which independently
  - queried & fetched hits from LS
    - cache id used to avoid multiple downloads
    - local database tracked doc stats
  - stripped HTML, but...
- all files and local <sup>?AMn</sup>databases copied manually to single PC
  - restrip, rehash to find true empties and dupes
  - rewinnow (HTML horrors)
  - tagging put on hold
- worker PCs communicate with servers only to...
  - avoid previously processed or rejected documents
  - upload texts and  $n$ -grams for final databases

# Bumps along the road

- PDF (*iFilter?*)
- PHP strip\_tags( )
- search by country codes
  - consistent, but not 100% reliable
  - excludes pages without specific geographic references
- encodings
  - hybrid encodings
  - mapping to CLAWS4 specs

# un-EVAL-CLEAN

- avoiding garbage pages
  - ignore outsized pages ( $< 5k / > 300k$ )
  - eliminate / excerpt\* pages with
    - too few / many words ( $< 500 / * > 50,000$  words)
    - too short / long paragraphs ( $< 13 / > 500$  w/p)
- sample text starting / ending with first / last paragraph with at least 13 words

# Unique documents downloaded

<i>filetype</i>	<i>count</i>	<i>words</i>
HTML	689,958	828,203,995
PDF	93,133	299,891,530
total	783,091	1,128,095,525

# Distribution “long tail”

	1-grams	2-grams	3-grams	4-grams	5-grams	6-grams
total unique	3,123,996	57,140,986	210,320,192	359,073,268	440,426,238	471,511,994
1x only	57.0%	67.0%	79.5%	87.7%	92.5%	94.8%
2x only	14.0%	13.1%	10.2%	7.3%	5.1%	3.9%
3 or more	29.1%	19.9%	10.3%	5.0%	2.3%	1.3%



# Implementing fulltext search - 1

## MySQL fulltext index / query

- + fully integrated with database
- slow, poor concurrency – FT search of BNC on PIE takes up to 2 mins (*10% the size of WaC*)
- doesn't index stopwords, "short" words, words occurring in more than 50% of records
  - PIE workaround for queries including non-indexed words: wildcard search on randomized version of corpus
- later additions require re-indexing entire corpus
- offline indexing typically not feasible

# Implementing fulltext search - 2

## Lucene

- + lightning fast
- + excellent scalability
- + widely used, active development
- + useful features coming (*payloads*)
- + offline indexing feasible
- Java (*slower, hosting, my learning curve*)
- challenging integration MySQL / PHP
- long stopword list
- phrases with stopwords not indexed

# Implementing fulltext search - 3

**Sphinx** <http://www.sphinxsearch.com>

- + fast search, index creation and update (*"delta" index*)
- + scalable to 100 GB (*no reviewers cover databases size of WaC*)
- + distributed search (*multiple servers*) for larger datasets
- + addresses shortcomings of others
  - + AND / OR / NEAR operators, phrase search
  - + multiple fields can be indexed and queried
  - + **highly customizable** stopwords, length cutoff, case and diacritic folding, stemming...
  - + tight integration with MySQL and PHP
  - + various results prioritization strategies
- phrase search with stopwords still requires "acrobatics"
- like other FT no wildcard support
- possibly difficult to deploy in a shared hosting environment (*requires daemon or custom-compiled MySQL*)