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
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)
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
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 ½ documents fetched with no country specified for broader range of sources
Representativeness

**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

**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

<table>
<thead>
<tr>
<th>concept – highly distributed parallel processing on 10-20 PCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>±8000 seed ST assigned to worker PCs, which independently</td>
</tr>
<tr>
<td>query &amp; fetch hits from LS</td>
</tr>
<tr>
<td>strip HTML, determine “keepers”</td>
</tr>
<tr>
<td>non-dupes, size</td>
</tr>
<tr>
<td>verify text-ness and English-ness</td>
</tr>
<tr>
<td>PoS-tag text</td>
</tr>
<tr>
<td>when all done, create $n$-grams of (un)tagged text</td>
</tr>
<tr>
<td>worker PCs communicate with servers only to...</td>
</tr>
<tr>
<td>avoid previously processed or rejected documents</td>
</tr>
<tr>
<td>upload texts and $n$-grams for final databases</td>
</tr>
</tbody>
</table>
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 databases copied manually to single PC
  - restrip, rehash to find true empties and dupes
  - rewinnnow (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)
      avoiding garbage pages
  - sample text starting / ending with first / last paragraph with at least 13 words
### Unique documents downloaded

<table>
<thead>
<tr>
<th>filetype</th>
<th>count</th>
<th>words</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>689,958</td>
<td>828,203,995</td>
</tr>
<tr>
<td>PDF</td>
<td>93,133</td>
<td>299,891,525</td>
</tr>
<tr>
<td>total</td>
<td>783,091</td>
<td>1,128,095,525</td>
</tr>
</tbody>
</table>
## Distribution “long tail”

<table>
<thead>
<tr>
<th></th>
<th>1-grams</th>
<th>2-grams</th>
<th>3-grams</th>
<th>4-grams</th>
<th>5-grams</th>
<th>6-grams</th>
</tr>
</thead>
<tbody>
<tr>
<td>total unique</td>
<td>3,123,996</td>
<td>57,140,986</td>
<td>210,320,192</td>
<td>359,073,268</td>
<td>440,426,238</td>
<td>471,511,994</td>
</tr>
<tr>
<td>1x only</td>
<td>57.0%</td>
<td>67.0%</td>
<td>79.5%</td>
<td>87.7%</td>
<td>92.5%</td>
<td>94.8%</td>
</tr>
<tr>
<td>2x only</td>
<td>14.0%</td>
<td>13.1%</td>
<td>10.2%</td>
<td>7.3%</td>
<td>5.1%</td>
<td>3.9%</td>
</tr>
<tr>
<td>3 or more</td>
<td>29.1%</td>
<td>19.9%</td>
<td>10.3%</td>
<td>5.0%</td>
<td>2.3%</td>
<td>1.3%</td>
</tr>
</tbody>
</table>
MySQL fulltext index / query

+ fully integrated with database
- slow, poor concurrency – FT search of BNC on PIE takes up to 2 mins \((10\% \text{ the size of } WaC)\)
- 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](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*)