Corpus Tools Brainstorming Session
What tools do we need for the future?

Laurence ANTHONY
Professor, Center for English Language Education,
Waseda University, Japan
Honorary Research Fellow (2013-2015), Lancaster University
anthony@waseda.jp
http://www.antlab.sci.waseda.ac.jp/software/

Overview of this brainstorming session

- **Background**
  - Reasons for organizing this session
  - The current state of corpus linguistics tools
  - The case for programming your own corpus tools

- **Brainstorming**:
  - Part 1: What do you currently do with corpora
  - Part 2: What do you want to do with corpora (that you cannot already)?
  - Part 3: What prevents you collaborating with a tools developer to create new tools... or programming your own?
  - Part 4: How can we increase corpus linguists’ attention to corpus tools?

Background:
Reasons for organizing this session

- To highlight the importance of tools in our field
- To introduce and discuss the current solutions to corpus linguists’ needs
- To generate ideas for new (and useful) corpus tools
- But…
  - This is not a session to voice criticisms of current tools
    - (Unless the criticism is of AntConc!)
A definition of corpus linguistics
- It is an empirical (experimental) approach
  - an analysis of actual patterns of use in target texts
- It uses a corpus of natural texts as the basis for analysis
  - a representative sample of target language stored as an electronic database
- It relies on computer software for analysis
  - results are generated using automatic and interactive techniques
  - observations are counted and results are interpreted

Background: The current state of corpus linguistics tools

Four Generations of Corpus Tools
(see McEnery & Hardie, 2012)
1st generation (1960s-1970s)
- run on mainframes, ASCII-based, very limited functions
  - e.g., A Concordance Generator (Smith, 1966)
  - e.g., Discon (Clark, 1966)
  - e.g., Drexel Concordance Program (Price, 1966)
  - e.g., Concordance (Dearing, 1966)
  - e.g., CLOC (Read, 1978)

Discon (Clark, 1966)

Correspond with Roger Clark or Lewis Sasin, 123 W Helmes,
University of Colorado, Boulder, Colorado.


Four Generations of Corpus Tools
2nd generation (1980s-1990s)
- run on PCs, ASCII-based, limited functions, scalability problems
  - e.g., Oxford Concordance Program (OCP) (Hockey, 1988)
  - e.g., Longman Mini-Concord (Chandler, 1989)
  - e.g., Kaye concordancer (Kaye, 1990)
  - e.g., MicroConcord (Scott & Johns, 1993)
Background: The current state of corpus linguistics tools

Four Generations of Corpus Tools

3rd-generation (2000s-present)
- more functions, better statistics, improved scalability, multi-language support, more user-friendly, simple, flexible
  - e.g., WordSmith Tools (Scott, 1996-2014)
  - e.g., MonoConc Pro (Barlow, 2000)
  - e.g., AntConc (Anthony, 2004-2014)
- Problems:
  - limited functionality (still)
  - limited access to corpora containing copyrighted data
  - limited ability to scale to massive (100 million+) corpora

4th-generation (late 2000s-present)
- better scalability, access to copyrighted data, available anywhere or everywhere (via browsers)
  - e.g., corpus.byu.edu (Davies, 2011), COCore (Hartle, 2011), SketchEngine (Kilgariff, 2011), Wmatrix (Rayson, 2011)
- Problems:
  - limited functionality (still)
  - overkill for many purposes (e.g., analyzing small corpora)
  - logistics: registration, setup, licenses, payment
  - limited access to public-domain corpora
  - no access to "personalized" corpora
  - e.g., institution-owned collections of copyrighted material
  - explosion in one-off, web-based, single-corpus interfaces
Background: The current state of corpus linguistics tools

International survey of corpus linguists. Responses: 891. (Tribble, 2012)

- corpus.byu.edu
- Antconc
- WordSmith Tools
- Sketch Engine
- Sarah [with BNC]
- Monocconc Pro
- Xaira (with BNC XML or your own...)
- WMatrix
- Oxford Concoriding Program
- Longman Mini-concorider
- Other

0% 5% 10% 15% 20% 25% 30%

"Which computer programs do you use for analysing corpora?"

"When you use pre-configured corpus programs, you're a little bit at the mercy of the company or individual selling them...

One final big advantage of programming languages, therefore, is that you are in the driver's seat."

(Biber, et al., 1998, p. 256)

(Gries, 2009, p. 11-12)
The reality for most corpus researchers, however, is that computer programming is in a completely different world ... without extensive training in programming ... it is likely that these [DIY] tools would be more restrictive, slower, less accurate and only work with small corpora.

(Anthony, 2009, p. 95)

"Research should be led by the science not the tool."

Professor Jim Wild, Lancaster University
Vice-President, Royal Astronomical Society

Brainstorming Part 1:
What do you currently do with corpora?

- Write down everything you do with corpora now?
  - e.g. identify common word/phrase patterns (in context)
  - e.g. find unusually frequent words/phrases in the corpus (i.e. keywords)
- Think about WHY you do these things with corpora
  - e.g. to help EFL students use standard English
  - e.g. to identify characteristic features of a text/genre

Brainstorming Part 1:
Participant Responses (added after discussion)

- find word/phrase patterns (KWIC)
- match patterns in text (via scripting)
- find word/phrase positions (Plot)
- generate statistics (e.g. using R)
- find collocates
- measure dispersion of word/phrase patterns
- find N-grams/Lexical bundles
- compare words/synonyms
- find Clusters
- identify characteristics of texts
- generate word lists
- generate keyword lists

Brainstorming Part 2:
What do you want to do with corpora (that you cannot already)?
Brainstorming Part 2:
What do you want to do with corpora?

- Write down interesting things you want to do with a corpus.
  - Do not worry if the idea is crazy or impossible.
  - But, also consider WHY you want to do this.
    - e.g. Find topic sentences in paragraphs
      - WHY – As a source of examples for writing classes

<table>
<thead>
<tr>
<th>Participant Responses (added after discussion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute distances between subsequent occurrences of search patterns (words, lemmas, POS, ...)</td>
</tr>
<tr>
<td>Quantify the degree of variability around search patterns</td>
</tr>
<tr>
<td>Extract definitions</td>
</tr>
<tr>
<td>Work with private data but allow for powerful handling of annotation (e.g. comparing frequencies of sub-corpora)</td>
</tr>
<tr>
<td>Continue holding brainstorming sessions of this kind</td>
</tr>
</tbody>
</table>

Brainstorming Part 3:
What prevents you collaborating with a tools developer to create new tools... or programming your own?

<table>
<thead>
<tr>
<th>Participant Responses (added after discussion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not confident to contact developers directly</td>
</tr>
<tr>
<td>Not enough time to learn programming skills</td>
</tr>
<tr>
<td>Not sure what features are already available in current tools</td>
</tr>
<tr>
<td>The need for programming is not immediately apparent</td>
</tr>
<tr>
<td>Powerful tools already exist</td>
</tr>
<tr>
<td>People are happy to do what they <em>can</em> with corpora instead of doing what they <em>should</em> do with corpora</td>
</tr>
</tbody>
</table>

Brainstorming Part 4:
How can we increase corpus linguists’ attention to corpus tools?

<table>
<thead>
<tr>
<th>Participant Responses (added after discussion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continue holding brainstorming sessions of this kind</td>
</tr>
<tr>
<td>Introduce corpus tools tracks in conference programs</td>
</tr>
</tbody>
</table>