Using the web as a source of linguistic data: experiences, problems and perspectives

Marco Baroni

SSLMIT, University of Bologna

ICST/CNR Roma, April 2005
Outline

Introduction

Frequency estimates from search engines
  Web-based Mutual Information

The “linguists’ friendly” interfaces

Building your own web corpus
  Small corpora via search engine queries
  Thinking Big: The “real” Linguist’s Search Engine

Enter WaCky!
The Web as Corpus

- Computational/corpus linguists, lexicographers, ontologists, language technologists constantly hungry for data.
The Web as Corpus

- Computational/corpus linguists, lexicographers, ontologists, language technologists constantly hungry for data.
- The web is a huge database of documents, mostly text.
The Web as Corpus

- Computational/corpus linguists, lexicographers, ontologists, language technologists constantly hungry for data.
- The web is a huge database of documents, mostly text.
- Kilgarriff: The web is the most exciting thing that happened to human beings in the last 20 years or so, and it’s all about linguistic communication – we linguists are in a good position to lead the study of it!!!
The Web as Corpus (cont.)


- English  76,598,718,000
- German  7,035,850,000
- Italian  1,845,026,000
- Finnish  326,379,000
- Esperanto  57,154,000
- Latin  55,943,000
- Basque  55,340,000
- Albanian  10,332,000

(Obsolete, conservative estimates)
Some General Problems

- Web is not balanced corpus.
Some General Problems

- Web is not balanced corpus.
- More worryingly: if you use search engine, no control over data.
Some General Problems

- Web is not balanced corpus.
- More worryingly: if you use search engine, no control over data.
- Constantly changing.
Some General Problems

- Web is not balanced corpus.
- More worryingly: if you use search engine, no control over data.
- Constantly changing.
- Many languages, a lot of non-native English.
Some General Problems

- Web is not balanced corpus.
- More worryingly: if you use search engine, no control over data.
- Constantly changing.
- Many languages, a lot of non-native English.
- Python.
Some General Problems

- Web is not balanced corpus.
- More worryingly: if you use search engine, no control over data.
- Constantly changing.
- Many languages, a lot of non-native English.
- Python.
- Google frequency of “colorless green ideas sleep furiously”: 13,000.
Some General Problems

- Web is not balanced corpus.
- More worryingly: if you use search engine, no control over data.
- Constantly changing.
- Many languages, a lot of non-native English.
- Python.
- Google frequency of “colorless green ideas sleep furiously”: 13,000.
- Desperately seeking Blaberus Giganteus.
Some General Problems

- Web is not balanced corpus.
- More worryingly: if you use search engine, no control over data.
- Constantly changing.
- Many languages, a lot of non-native English.
- Python.
- Google frequency of “colorless green ideas sleep furiously”: 13,000.
- Desperately seeking Blaberus Giganteus.
- ...
But still... more data is better data!
(Mercer quoted by Church)

- Banko and Brill 2001 HLT paper.
But still... more data is better data!
(Mercer quoted by Church)

- Banko and Brill 2001 HLT paper.
- Confusion set disambiguation task.
But still... more data is better data!
(Mercer quoted by Church)

- Banko and Brill 2001 HLT paper.
- Confusion set disambiguation task.
- Choose correct word in context from set of words it is typically confused with: *affect/effect, principal/principle.*
But still... more data is better data!
(Mercer quoted by Church)

- Banko and Brill 2001 HLT paper.
- Confusion set disambiguation task.
- Choose correct word in context from set of words it is typically confused with: *affect/effect*, *principal/principle*.
- Even most naive learning algorithm trained on 10M word training set outperforms any learner trained on 1M word training set.
But still... more data is better data!
(Mercer quoted by Church)

- Banko and Brill 2001 HLT paper.
- Confusion set disambiguation task.
- Choose correct word in context from set of words it is typically confused with: *affect/effect*, *principal/principle*.
- Even most naive learning algorithm trained on 10M word training set outperforms any learner trained on 1M word training set.
- With 1 billion word training set, learners have not reached performance asymptote.
But still... more data is better data!
(Mercer quoted by Church)

- Banko and Brill 2001 HLT paper.
- Confusion set disambiguation task.
- Choose correct word in context from set of words it is typically confused with: affect/effect, principal/principle.
- Even most naive learning algorithm trained on 10M word training set outperforms any learner trained on 1M word training set.
- With 1 billion word training set, learners have not reached performance asymptote.
- (Learn language function by simple algorithm that has access to full extension of function.)
More web-data is better data!

More web-data is better data!

- Google- and AltaVista-based frequencies of A N, N N and V N bigrams:
More web-data is better data!

- Google- and AltaVista-based frequencies of A N, N N and V N bigrams:
  - correlate with BNC and NANTC frequencies;
More web-data is better data!

- Google- and AltaVista-based frequencies of A N, N N and V N bigrams:
  - correlate with BNC and NANTC frequencies;
  - correlate with WordNet-class-based smoothed frequencies;
More web-data is better data!

- Google- and AltaVista-based frequencies of A N, N N and V N bigrams:
  - correlate with BNC and NANTC frequencies;
  - correlate with WordNet-class-based smoothed frequencies;
  - correlate with human plausibility judgments more than corpus-based frequencies do (smoothed or not smoothed).
Approaches to Web as Corpus

- Collect (frequency) data directly from commercial search engines (e.g. Turney 2001, many many others).
Approaches to Web as Corpus

- Collect (frequency) data directly from commercial search engines (e.g. Turney 2001, many many others).
Approaches to Web as Corpus

- Collect (frequency) data directly from commercial search engines (e.g. Turney 2001, many many others).
- Small(-ish), focused crawls of the web to find and retrieve relevant pages (e.g. Ghani et al. 2001, Baroni and Bernardini 2004, Sharoff submitted).
Approaches to Web as Corpus

- Collect (frequency) data directly from commercial search engines (e.g. Turney 2001, many many others).
- Small(-ish), focused crawls of the web to find and retrieve relevant pages (e.g. Ghani et al. 2001, Baroni and Bernardini 2004, Sharoff *submitted*).
- WaCky!
Outline

Introduction

Frequency estimates from search engines
  Web-based Mutual Information

The “linguists’ friendly” interfaces

Building your own web corpus
  Small corpora via search engine queries
  Thinking Big: The “real” Linguist’s Search Engine

Enter WaCky!
Collecting frequency data from search engines

- Probably the most popular method (Keller and Lapata 2003, Turney 2001, many others).
Collecting frequency data from search engines

- Probably the most popular method (Keller and Lapata 2003, Turney 2001, many others).
- Rough approximation to frequency, but:
Collecting frequency data from search engines

- Probably the most popular method (Keller and Lapata 2003, Turney 2001, many others).
- Rough approximation to frequency, but:
  - Empirically successful;
Collecting frequency data from search engines

- Probably the most popular method (Keller and Lapata 2003, Turney 2001, many others).
- Rough approximation to frequency, but:
  - Empirically successful;
  - Easy: the engine does most of the hard work.
Collecting frequency data from search engines

- Probably the most popular method (Keller and Lapata 2003, Turney 2001, many others).
- Rough approximation to frequency, but:
  - Empirically successful;
  - Easy: the engine does most of the hard work.
- Web-based mutual information: typical example of research using search engine-based frequency data.
Web-based Mutual Information (WMI)

Turney 2001
Web-based Mutual Information (WMI)

Turney 2001

- (Pointwise) mutual information:

\[ MI(w_1, w_2) = \log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)} = \log_2 N \frac{C(w_1, w_2)}{C(w_1)C(w_2)} \]
Web-based Mutual Information (WMI)

Turney 2001

- (Pointwise) mutual information:

\[ MI(w_1, w_2) = \log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)} = \log_2 N \frac{C(w_1, w_2)}{C(w_1)C(w_2)} \]

- WMI: compute mutual information of word pairs using frequency/cooccurrence frequency data extracted from the web via AltaVista search engine.

\[ WMI(w_1, w_2) = \log_2 N \frac{\text{hits}(w_1 \text{ NEAR } w_2)}{\text{hits}(w_1)\text{hits}(w_2)} \]
Web-based Mutual Information

- Semantic similarity as **direct cooccurrence** (vs. occurrence in similar contexts).
Web-based Mutual Information

- Semantic similarity as direct cooccurrence (vs. occurrence in similar contexts).
- Simplicity of method counterbalanced by size of database (the WWW).
Web-based Mutual Information

- Semantic similarity as direct cooccurrence (vs. occurrence in similar contexts).
- Simplicity of method counterbalanced by size of database (the WWW).
Web-based Mutual Information

- Semantic similarity as direct cooccurrence (vs. occurrence in similar contexts).
- Simplicity of method counterbalanced by size of database (the WWW).
- Most researchers report that WMI outperforms more sophisticated methods based on smaller corpora.
Web-based Mutual Information

- Semantic similarity as **direct cooccurrence** (vs. occurrence in similar contexts).
- Simplicity of method counterbalanced by size of database (the WWW).
- Most researchers report that WMI outperforms more sophisticated methods based on smaller corpora.
WMI takes the TOEFL

Turney 2001

▶ TOEFL synonym match task.
WMI takes the TOEFL
Turney 2001

- TOEFL synonym match task.
- Target: *levied*; Candidates: *imposed, believed, requested, correlated*. 
WMI takes the TOEFL
Turney 2001

- TOEFL synonym match task.
- Target: *levied*; Candidates: *imposed, believed, requested, correlated*.
WMI takes the TOEFL (cont.)

- Performance on TOEFL synonym match task:
WMI takes the TOEFL (cont.)

- Performance on TOEFL synonym match task:
  - Average foreign test taker: 64.5%
WMI takes the TOEFL (cont.)

- Performance on TOEFL synonym match task:
  - Average foreign test taker: 64.5%
  - Latent Semantic Analysis: 65.4%
WMI takes the TOEFL (cont.)

- Performance on TOEFL synonym match task:
  - Average foreign test taker: 64.5%
  - Latent Semantic Analysis: 65.4%
  - WMI: 72.5%
WMI and synonym detection in terminology

- Baroni and Bisi 2004 applied WMI-method to synonym mining task in technical domain.
WMI and synonym detection in terminology

- Baroni and Bisi 2004 applied WMI-method to synonym mining task in technical domain.
- A harder task:
WMI and synonym detection in terminology

- Baroni and Bisi 2004 applied WMI-method to synonym mining task in technical domain.
- A harder task:
  - Technical terms less frequent than general language terms (potential data sparseness issues);
WMI and synonym detection in terminology

- Baroni and Bisi 2004 applied WMI-method to synonym mining task in technical domain.
- A harder task:
  - Technical terms less frequent than general language terms (potential data sparseness issues);
  - All terms in domain tend to be semantically related, to some extent.
Materials

- Nautical terminology.
Materials

- Nautical terminology.
- Terms and relational information from structured termbase of Bisi (2003).
Task

- Given a list of pairs in any order, rank them so that synonym pairs will be on top of list.
Task: example

- decks/cockpit
- frames/ribs
- bottom/hull
- ...
- frames/hull
Task: example

- frames/ribs
- bottom/hull
- decks/cockpit
- ...
- frames/hull
Task: settings

- Synonym term pairs vs. random term pairs (Exp 1).
Task: settings

- Synonym term pairs vs. random term pairs (Exp 1).
- Synonym term pairs vs. other “nymic” pairs (Exp 2).
Cosine Similarity

- Term of comparison.
Cosine Similarity

- Term of comparison.
- Intuition: Words with similar patterns of cooccurrence are likely to be similar.
Cosine Similarity

- Term of comparison.
- Intuition: Words with similar patterns of cooccurrence are likely to be similar.
- Correlation of vectors of cooccurrence frequencies of targets with (almost) all words in corpus:

$$\cos(\vec{x}, \vec{y}) = \vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i$$
Cosine Similarity (cont.)

- Corpora:
Cosine Similarity (cont.)

- Corpora:
  - 1.2M word specialized corpus manually assembled by terminologist;
  - 4.27M word corpus constructed via random nautical term queries to Google.
Cosine Similarity (cont.)

- Corpora:
  - 1.2M word specialized corpus manually assembled by terminologist;
  - 4.27M word corpus constructed via random nautical term queries to Google.
Cosine Similarity (cont.)

▶ Corpora:
  ▶ 1.2M word specialized corpus manually assembled by terminologist;
  ▶ 4.27M word corpus constructed via random nautical term queries to Google.

▶ Context windows:
Cosine Similarity (cont.)

- Corpora:
  - 1.2M word specialized corpus manually assembled by terminologist;
  - 4.27M word corpus constructed via random nautical term queries to Google.

- Context windows:
  - 2 words to either side of target;
Cosine Similarity (cont.)

- Corpora:
  - 1.2M word specialized corpus manually assembled by terminologist;
  - 4.27M word corpus constructed via random nautical term queries to Google.

- Context windows:
  - 2 words to either side of target;
  - 5 words to either side of target.
Experiment 1: Data

- 24 synonym pairs (e.g., bottom/hull, frames/ribs, displacement/weight).
Experiment 1: Data

- 24 synonym pairs (e.g., bottom/hull, frames/ribs, displacement/weight).
- 124 non-synonym pairs:
Experiment 1: Data

- 24 synonym pairs (e.g., *bottom/hull*, *frames/ribs*, *displacement/weight*).
- 124 non-synonym pairs:
  - 100 random pairs of nautical terms;
Experiment 1: Data

- 24 synonym pairs (e.g., *bottom/hull*, *frames/ribs*, *displacement/weight*).
- 124 non-synonym pairs:
  - 100 random pairs of nautical terms;
  - 24 recombinations of terms in synonym set.
Experiment 1: Data

- 24 synonym pairs (e.g., bottom/hull, frames/ribs, displacement/weight).
- 124 non-synonym pairs:
  - 100 random pairs of nautical terms;
  - 24 recombinations of terms in synonym set.
- 29% of random pairs rated “strongly semantically related” (e.g., awning/stern board, install/hatch, keel/coated).
### Experiment 1: Results
Percentage precision at various percentage recall levels

<table>
<thead>
<tr>
<th>recall</th>
<th>WMI</th>
<th>Cosines</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>man corp</td>
<td>man corp</td>
<td>web corp</td>
<td>web corp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2-word win</td>
<td>5-word win</td>
<td>2-word win</td>
<td>5-word win</td>
<td></td>
</tr>
<tr>
<td>12.5</td>
<td>100.0</td>
<td>100.0</td>
<td>60.0</td>
<td>60.0</td>
<td>42.9</td>
<td></td>
</tr>
<tr>
<td>25.0</td>
<td>100.0</td>
<td>75.0</td>
<td>60.0</td>
<td>46.2</td>
<td>46.2</td>
<td></td>
</tr>
<tr>
<td>37.5</td>
<td>90.0</td>
<td>42.9</td>
<td>39.1</td>
<td>40.9</td>
<td>45.0</td>
<td></td>
</tr>
<tr>
<td>50.0</td>
<td>92.3</td>
<td>17.9</td>
<td>19.4</td>
<td>26.7</td>
<td>25.5</td>
<td></td>
</tr>
<tr>
<td>62.5</td>
<td>88.2</td>
<td>10.8</td>
<td>15.0</td>
<td>19.0</td>
<td>17.6</td>
<td></td>
</tr>
<tr>
<td>75.0</td>
<td>36.7</td>
<td>12.7</td>
<td>12.7</td>
<td>12.7</td>
<td>13.4</td>
<td></td>
</tr>
<tr>
<td>75.0</td>
<td>30.4</td>
<td>14.5</td>
<td>14.5</td>
<td>14.5</td>
<td>14.5</td>
<td></td>
</tr>
<tr>
<td>100.0</td>
<td>16.2</td>
<td>16.2</td>
<td>16.2</td>
<td>16.2</td>
<td>16.2</td>
<td></td>
</tr>
</tbody>
</table>
Experiment 2: Data

- Same 24 synonym pairs as above.
Experiment 2: Data

- Same 24 synonym pairs as above.
- 31 nymic pairs from Bisi termbase added to test set:
Experiment 2: Data

- Same 24 synonym pairs as above.
- 31 nymic pairs from Bisi termbase added to test set:
  - 19 cohyponym pairs (e.g., *Bruce anchor/mushroom anchor*);
Experiment 2: Data

- Same 24 synonym pairs as above.
- 31 nymic pairs from Bisi termbase added to test set:
  - 19 cohyponym pairs (e.g., *Bruce anchor/mushroom anchor*);
  - 10 hypo/hypernym pairs (e.g., *stern platform/sun deck*);
Experiment 2: Data

- Same 24 synonym pairs as above.
- 31 nymic pairs from Bisi termbase added to test set:
  - 19 cohyponym pairs (e.g., Bruce anchor/mushroom anchor);
  - 10 hypo/hypernym pairs (e.g., stern platform/sun deck);
  - 2 antonyms (e.g., ahead/astern).
Experiment 2: Data

- Same 24 synonym pairs as above.
- 31 nymic pairs from Bisi termbase added to test set:
  - 19 cohyponym pairs (e.g., *Bruce anchor/mushroom anchor*);
  - 10 hypo/hypernym pairs (e.g., *stern platform/sun deck*);
  - 2 antonyms (e.g., *ahead/astern*).
- 31 randomly selected non-synonym pairs removed from test set (same synonym-to-non-synonym pair ratio as above).
## Experiment 2: Results
Percentage precision at various percentage recall levels

<table>
<thead>
<tr>
<th>recall</th>
<th>WMI</th>
<th>Cosines</th>
<th>Cosines</th>
<th>Cosines</th>
<th>Cosines</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>man corp 2-word win</td>
<td>man corp 5-word win</td>
<td>web corp 2-word win</td>
<td>web corp 5-word win</td>
</tr>
<tr>
<td>12.5</td>
<td>60.0</td>
<td>42.9</td>
<td>37.5</td>
<td>27.3</td>
<td>20.0</td>
</tr>
<tr>
<td>25.0</td>
<td>33.3</td>
<td>46.2</td>
<td>46.2</td>
<td>28.6</td>
<td>27.3</td>
</tr>
<tr>
<td>37.5</td>
<td>36.0</td>
<td>39.1</td>
<td>39.1</td>
<td>29.0</td>
<td>31.0</td>
</tr>
<tr>
<td>50.0</td>
<td>40.0</td>
<td>19.7</td>
<td>21.1</td>
<td>23.1</td>
<td>22.6</td>
</tr>
<tr>
<td>62.5</td>
<td>37.5</td>
<td>10.8</td>
<td>17.4</td>
<td>19.2</td>
<td>18.1</td>
</tr>
<tr>
<td>75.0</td>
<td>26.5</td>
<td>12.7</td>
<td>12.7</td>
<td>12.7</td>
<td>14.1</td>
</tr>
<tr>
<td>87.5</td>
<td>25.6</td>
<td>14.5</td>
<td>14.5</td>
<td>14.5</td>
<td>14.5</td>
</tr>
<tr>
<td>100.0</td>
<td>16.2</td>
<td>16.2</td>
<td>16.2</td>
<td>16.2</td>
<td>16.2</td>
</tr>
</tbody>
</table>
Houston, we have a problem
Houston, we have a problem

- On 31 March 2004, AltaVista’s parent company Yahoo! replaced the AltaVista’s engine with Yahoo!’s own engine.
Houston, we have a problem

- On 31 March 2004, AltaVista’s parent company Yahoo! replaced the AltaVista’s engine with Yahoo!’s own engine.
- End of the NEAR operator.
Houston, we have a problem

- On 31 March 2004, AltaVista’s parent company Yahoo! replaced the AltaVista’s engine with Yahoo!’s own engine.
- End of the NEAR operator.
- Change of underlying database.
Houston, we have a problem

- On 31 March 2004, AltaVista’s parent company Yahoo! replaced the AltaVista’s engine with Yahoo!’s own engine.
- End of the NEAR operator.
- Change of underlying database.
- WMI without NEAR:

\[
WMI(w_1, w_2) = \log_2 N \frac{\text{hits}(w_1, w_2)}{\text{hits}(w_1)\text{hits}(w_2)}
\]
Experiment 1: with and without NEAR
Percentage precision at various percentage recall levels

<table>
<thead>
<tr>
<th>recall</th>
<th>AltaVista w/ NEAR</th>
<th>AltaVista w/o NEAR</th>
<th>Google</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.5</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>25.0</td>
<td>100.0</td>
<td>100.0</td>
<td>85.7</td>
</tr>
<tr>
<td>37.5</td>
<td>90.0</td>
<td>100.0</td>
<td>81.8</td>
</tr>
<tr>
<td>50.0</td>
<td>92.3</td>
<td>75</td>
<td>85.7</td>
</tr>
<tr>
<td>62.5</td>
<td>88.2</td>
<td>62.5</td>
<td>60.0</td>
</tr>
<tr>
<td>75.0</td>
<td>36.7</td>
<td>45.0</td>
<td>64.3</td>
</tr>
<tr>
<td>87.5</td>
<td>30.4</td>
<td>34.4</td>
<td>45.6</td>
</tr>
<tr>
<td>100.0</td>
<td>16.2</td>
<td>19.3</td>
<td>17.3</td>
</tr>
</tbody>
</table>
Experiment 2: with and without NEAR
Percentage precision at various percentage recall levels

<table>
<thead>
<tr>
<th>recall</th>
<th>AltaVista w/ NEAR</th>
<th>AltaVista w/o NEAR</th>
<th>Google</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.5</td>
<td>60.0</td>
<td>42.8</td>
<td>50.0</td>
</tr>
<tr>
<td>25.0</td>
<td>33.3</td>
<td>50.0</td>
<td>37.5</td>
</tr>
<tr>
<td>37.5</td>
<td>36.0</td>
<td>52.9</td>
<td>45.0</td>
</tr>
<tr>
<td>50.0</td>
<td>40.0</td>
<td>38.7</td>
<td>40.0</td>
</tr>
<tr>
<td>62.5</td>
<td>37.5</td>
<td>32.6</td>
<td>31.9</td>
</tr>
<tr>
<td>75.0</td>
<td>26.5</td>
<td>28.6</td>
<td>34.0</td>
</tr>
<tr>
<td>87.5</td>
<td>25.6</td>
<td>25.6</td>
<td>30.0</td>
</tr>
<tr>
<td>100.0</td>
<td>16.2</td>
<td>18.5</td>
<td>17.0</td>
</tr>
</tbody>
</table>
Pros and cons of search engine frequencies

- The main advantage: it’s easy.
Pros and cons of search engine frequencies

- The main advantage: it’s easy.
- The main problem: we depend on commercial search engines.
Pros and cons of search engine frequencies

- The main advantage: it’s easy.
- The main problem: we depend on commercial search engines.
- Linguist’s satisfaction is obviously not their priority.
A telling anecdote

(Talking to a new acquaintance who works at Google)
A telling anecdote

(Talking to a new acquaintance who works at Google)

Me: So, do you guys have plans to introduce the NEAR operator?
A telling anecdote

(Talking to a new acquaintance who works at Google)

Me: So, do you guys have plans to introduce the NEAR operator?

The Google Acquaintance: You are a linguist right? Only linguists ask about that sort of stuff...
Consequences

- Limited query options (not even diacritics and accents), limited research options.
Consequences

- Limited query options (not even diacritics and accents), limited research options.
- You must know the words you are looking for.
Consequences

- Limited query options (not even diacritics and accents), limited research options.
- You must know the words you are looking for.
- No annotation, few, unreliable metadata.
Consequences

- Limited query options (not even diacritics and accents), limited research options.
- You must know the words you are looking for.
- No annotation, few, unreliable metadata.
- Automated querying constraints, over-querying strongly discouraged.
Consequences

- Limited query options (not even diacritics and accents), limited research options.
- You must know the words you are looking for.
- No annotation, few, unreliable metadata.
- Automated querying constraints, over-querying strongly discouraged.
- We know very little about the data we get.
Consequences

- Limited query options (not even diacritics and accents), limited research options.
- You must know the words you are looking for.
- No annotation, few, unreliable metadata.
- Automated querying constraints, over-querying strongly discouraged.
- We know very little about the data we get.
- No control over how search engines evolve.
Consequences

- Limited query options (not even diacritics and accents), limited research options.
- You must know the words you are looking for.
- No annotation, few, unreliable metadata.
- Automated querying constraints, over-querying strongly discouraged.
- We know very little about the data we get.
- No control over how search engines evolve.
- Brittleness!
Fletcher 2004 saying the same things

Search engines are not research libraries but commercial enterprises targeted at the needs of the general public. The availability and implementation of their services change constantly: features are added or dropped to mimic or outdo the competition; acquisitions and mergers threaten their independence; financial uncertainties and legal battles challenge their very survival. The search sites’ quest for revenue can diminish the objectivity of their search results, and various “page ranking” algorithms may lead to results that are not representative of the Web as a whole. Most frustrating is the minimal support for the requirements of serious researchers: current trends lead away from sites like AltaVista supporting sophisticated complex queries (which few users employ) to ones like Google offering only simple search criteria. In short, the search engines’ services are useful to investigators by coincidence, not design, and researchers are tolerated on mainstream search sites only as long as their use does not affect site performance adversely.
### Worrying data from the Google APIs

Pattern discovered by Luca Onnis

<table>
<thead>
<tr>
<th>Query</th>
<th>APIs</th>
<th>Website</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>pleasantly</td>
<td>369000</td>
<td>870000</td>
<td>0.42</td>
</tr>
<tr>
<td>awkwardly</td>
<td>124000</td>
<td>292000</td>
<td>0.42</td>
</tr>
<tr>
<td>silent</td>
<td>4610000</td>
<td>11000000</td>
<td>0.42</td>
</tr>
<tr>
<td>pleasantly silent</td>
<td>107</td>
<td>135</td>
<td>0.79</td>
</tr>
<tr>
<td>awkwardly silent</td>
<td>396</td>
<td>566</td>
<td>0.70</td>
</tr>
</tbody>
</table>
A few more things to worry about

- Google inflating its counts (Veronis’s blog, 2005).
A few more things to worry about

- Google inflating its counts (Veronis’s blog, 2005).
- Is the * operator still supported?
Outline

Introduction

Frequency estimates from search engines
  Web-based Mutual Information

The “linguists’ friendly” interfaces

Building your own web corpus
  Small corpora via search engine queries
  Thinking Big: The “real” Linguist’s Search Engine

Enter WaCky!
The “linguist’s friendly” interfaces

- WebCorp, KwicFinder, Linguist’s Search Engine.
The “linguist’s friendly” interfaces

- WebCorp, KwicFinder, Linguist’s Search Engine.
- “Wrappers” around Google, AltaVista, etc.
The “linguist’s friendly” interfaces

- WebCorp, KwicFinder, Linguist’s Search Engine.
- “Wrappers” around Google, AltaVista, etc.
- Nice interfaces, but ultimately inherit all problems of search engines, and perhaps add some more with their filters...
The “linguist’s friendly” interfaces

- WebCorp, KwicFinder, Linguist’s Search Engine.
- “Wrappers” around Google, AltaVista, etc.
- Nice interfaces, but ultimately inherit all problems of search engines, and perhaps add some more with their filters...
- E.g., “spongi*” query in webCorp (Stefan Evert).
Outline

Introduction

Frequency estimates from search engines
  Web-based Mutual Information

The “linguists’ friendly” interfaces

Building your own web corpus
  Small corpora via search engine queries
  Thinking Big: The “real” Linguist’s Search Engine

Enter WaCky!
Building special corpora with search engine queries

- By downloading text, more control over data.
Building special corpora with search engine queries

- By downloading text, more control over data.
- But less work, more targeted data than spidering your own corpus.
Building special corpora with search engine queries

- By downloading text, more control over data.
- But less work, more targeted data than spidering your own corpus.
- Good for “special purposes” corpora:
Building special corpora with search engine queries

- By downloading text, more control over data.
- But less work, more targeted data than spidering your own corpus.
- Good for “special purposes” corpora:
  - “minority” languages (CorpusBuilder; Ghani, Jones, Mladenić, CIKM-2001).;
Building special corpora with search engine queries

- By downloading text, more control over data.
- But less work, more targeted data than spidering your own corpus.
- Good for “special purposes” corpora:
  - “minority” languages (CorpusBuilder; Ghani, Jones, Mladenić, CIKM-2001);
  - specialized sub-languages (BootCaT).
The BootCaT tools

- **Bootstrap**trapping Corpora and Terms from the web.
The BootCaT tools

- **Boot**strapping **C**orpora and **T**erms from the web.
- Perl scripts freely available from:
  
  http://sslmit.unibo.it/∼baroni/bootcat.html
The BootCaT tools

- **Boot**strapping **Corpora** and **T**erms from the web.
- Perl scripts freely available from: http://sslmit.unibo.it/~baroni/bootcat.html
- Original motivation: fast construction of ad-hoc corpora and term lists for translation/interpreting tasks, terminography.
The BootCaT procedure

1. Select initial terms
2. Query Google for random term combinations
3. Retrieve pages and format as text (corpus)
4. Extract new terms via corpus comparison
5. Extract multi-word terms using corpus, uni-terms and...

- Distributional patterns
- POS templates
Terms and Term Combinations

- 5-20 terms typical of domain.
Terms and Term Combinations

- 5-20 terms typical of domain.
- Selection: human or automated (e.g. via text/corpus comparison).
Terms and Term Combinations

- 5-20 terms typical of domain.
- Selection: human or automated (e.g. via text/corpus comparison).
- Seed terms randomly combined into tuples to perform Google queries:
Terms and Term Combinations

- 5-20 terms typical of domain.
- Selection: human or automated (e.g. via text/corpus comparison).
- Seed terms randomly combined into tuples to perform Google queries:
  - Longer tuples: better precision;
Terms and Term Combinations

- 5-20 terms typical of domain.
- Selection: human or automated (e.g. via text/corpus comparison).
- Seed terms randomly combined into tuples to perform Google queries:
  - Longer tuples: better precision;
  - Shorter tuples: better recall.
Corpus/Term Bootstrapping

- The bootstrap:

  1. Retrieve corpus from web via Google tuple queries;
  2. Extract typical terms through statistical comparison with reference corpus (using Mutual Information, Log-Likelihood Ratio, etc.);
  3. Use found terms as new seeds and build new random tuples;
  4. Go back to 1.

- Retrieved pages formatted as text (character set issues, non-text format issues; in Japanese: tokenization issues).

- Reference corpus: better if balanced, but any corpus on different topic will usually do (but in Japanese register of corpora turns out to be crucial!)
Corpus/Term Bootstrapping

- The bootstrap:
  1. Retrieve corpus from web via Google tuple queries;
Corpus/Term Bootstrapping

- The bootstrap:
  1. Retrieve corpus from web via Google tuple queries;
  2. Extract typical terms through statistical comparison with reference corpus (using Mutual Information, Log-Likelihood Ratio, etc.);
Corpus/Term Bootstrapping

- The bootstrap:
  1. Retrieve corpus from web via Google tuple queries;
  2. Extract typical terms through statistical comparison with reference corpus (using Mutual Information, Log-Likelihood Ratio, etc.);
  3. Use found terms as new seeds and build new random tuples;
Corpus/Term Bootstrapping

The bootstrap:

1. Retrieve corpus from web via Google tuple queries;
2. Extract typical terms through statistical comparison with reference corpus (using Mutual Information, Log-Likelihood Ratio, etc.);
3. Use found terms as new seeds and build new random tuples;
4. Go back to 1.
Corpus/Term Bootstrapping

- The bootstrap:
  1. Retrieve corpus from web via Google tuple queries;
  2. Extract typical terms through statistical comparison with reference corpus (using Mutual Information, Log-Likelihood Ratio, etc.);
  3. Use found terms as new seeds and build new random tuples;
  4. Go back to 1.

- Retrieved pages formatted as text (character set issues, non-text format issues; in Japanese: tokenization issues).
Corpus/Term Bootstrapping

The bootstrap:
1. Retrieve corpus from web via Google tuple queries;
2. Extract typical terms through statistical comparison with reference corpus (using Mutual Information, Log-Likelihood Ratio, etc.);
3. Use found terms as new seeds and build new random tuples;
4. Go back to 1.

Retrieved pages formatted as text (character set issues, non-text format issues; in Japanese: tokenization issues).

Reference corpus: better if balanced, but any corpus on different topic will usually do (but in Japanese register of corpora turns out to be crucial!)
Example 1: Pseudo-seizures in English
Baroni and Bernardini 2004

Seed terms: dissociative, epilepsy, interventions, posttraumatic, pseudoseizures, ptsd.
Example 1: Pseudo-seizures in English
Baroni and Bernardini 2004

- Seed terms: *dissociative, epilepsy, interventions, posttraumatic, pseudoseizures, ptsd*.
- Reference: Brown (1.1M words).
Example 1: Pseudo-seizures in English
Baroni and Bernardini 2004

- Seed terms: dissociative, epilepsy, interventions, posttraumatic, pseudoseizures, ptsd.
- Reference: Brown (1.1M words).
- Corpus comparison: via Log Odds Ratio.
Example 1: Pseudo-seizures in English
Baroni and Bernardini 2004

- Seed terms: dissociative, epilepsy, interventions, posttraumatic, pseudoseizures, ptsd.
- Reference: Brown (1.1M words).
- Corpus comparison: via Log Odds Ratio.
- Two iterations.
Example 1: Pseudo-seizures in English
Baroni and Bernardini 2004

- Seed terms: dissociative, epilepsy, interventions, posttraumatic, pseudoseizures, ptsd.
- Reference: Brown (1.1M words).
- Corpus comparison: via Log Odds Ratio.
- Two iterations.
- 1.4M word corpus constructed, 1800 unigram terms extracted.
Example 1: Pseudo-seizures in English
Baroni and Bernardini 2004

- Seed terms: *dissociative, epilepsy, interventions, posttraumatic, pseudoseizures, ptsd*.
- Reference: Brown (1.1M words).
- Corpus comparison: via Log Odds Ratio.
- Two iterations.
- 1.4M word corpus constructed, 1800 unigram terms extracted.
- 20/30 randomly selected documents from corpus rated as relevant and informative.
Example 2: Hotel terminology in Japanese
Baroni and Ueyama 2004

- 20 manually selected initial terms.
Example 2: Hotel terminology in Japanese
Baroni and Ueyama 2004

- 20 manually selected initial terms.
- 3.5M word reference corpus built with BootCaT using random elementary Japanese words as seeds.
Example 2: Hotel terminology in Japanese
Baroni and Ueyama 2004

- 20 manually selected initial terms.
- 3.5M word reference corpus built with BootCaT using random elementary Japanese words as seeds.
- Corpus comparison: via MI and Log Likelihood Ratio.
Example 2: Hotel terminology in Japanese
Baroni and Ueyama 2004

- 20 manually selected initial terms.
- 3.5M word reference corpus built with BootCaT using random elementary Japanese words as seeds.
- Corpus comparison: via MI and Log Likelihood Ratio.
- Three iterations.
Example 2: Hotel terminology in Japanese
Baroni and Ueyama 2004

- 20 manually selected initial terms.
- 3.5M word reference corpus built with BootCaT using random elementary Japanese words as seeds.
- Corpus comparison: via MI and Log Likelihood Ratio.
- Three iterations.
- 1.3M word corpus constructed, 424 terms extracted.
Example 2: Hotel terminology in Japanese
Baroni and Ueyama 2004

- 20 manually selected initial terms.
- 3.5M word reference corpus built with BootCaT using random elementary Japanese words as seeds.
- Corpus comparison: via MI and Log Likelihood Ratio.
- Three iterations.
- 1.3M word corpus constructed, 424 terms extracted.
- 76/90 randomly selected documents assigned highest relevance/informativeness rating.
Example 2: Hotel terminology in Japanese
Baroni and Ueyama 2004

- 20 manually selected initial terms.
- 3.5M word reference corpus built with BootCaT using random elementary Japanese words as seeds.
- Corpus comparison: via MI and Log Likelihood Ratio.
- Three iterations.
- 1.3M word corpus constructed, 424 terms extracted.
- 76/90 randomly selected documents assigned highest relevance/informativeness rating.
- 58.4% terms rated very relevant, 81.7% rated at least somewhat relevant.
Applications

- Languages: English, Italian, Japanese, Spanish, German, French, Russian, Chinese, Danish.
Applications

- Languages: English, Italian, Japanese, Spanish, German, French, Russian, Chinese, Danish.
- Domains: medical, legal, meteorology, food, nautical terminology, (e-)commerce...
Applications

- Languages: English, Italian, Japanese, Spanish, German, French, Russian, Chinese, Danish.
- Domains: medical, legal, meteorology, food, nautical terminology, (e-)commerce...
- Uses: technical translation, interpreting tasks, resources for LSP teaching, populating ontologies, expanding a lexicon in systematic ways, general corpus construction (Sharoff submitted).
Ongoing and planned work

- Special queries.
- Better character set handling.
- Better pdf/doc conversion.
- Better integration with UCS and other tools.
- Multi-term extraction.
- Yahoo API?
Pros

- We still rely on commercial search engine, but less so.
Pros

▶ We still rely on commercial search engine, but less so.
▶ We only use most basic query function, less likely to change.
Pros

- We still rely on commercial search engine, but less so.
- We only use most basic query function, less likely to change.
- Language filtering and good relevance-ranking are crucial characteristics of successful search engines.
Pros

▶ We still rely on commercial search engine, but less so.
▶ We only use most basic query function, less likely to change.
▶ Language filtering and good relevance-ranking are crucial characteristics of successful search engines.
▶ We are less likely to bother engine by over-querying, since with one query we can obtain MBs of data.
Pros

- We still rely on commercial search engine, but less so.
- We only use most basic query function, less likely to change.
- Language filtering and good relevance-ranking are crucial characteristics of successful search engines.
- We are less likely to bother engine by over-querying, since with one query we can obtain MBs of data.
- We have full control over data (e.g. frequency counts, parsing, manual URL filtering) because we download them.
Cons

- We *still* rely on commercial search engine:
Cons

- We *still* rely on commercial search engine:
  - What happens if Google discontinues API service?
Cons

- *We still* rely on commercial search engine:
  - What happens if Google discontinues API service?
  - What happens if Google does something too smart or too commercial with the page ranks?
**Cons**

- We *still* rely on commercial search engine:
  - What happens if Google discontinues API service?
  - What happens if Google does something too smart or too commercial with the page ranks?
- Good for content-driven corpus building, problems with syntax/style/genre-based filtering.
Cons

▶ We *still* rely on commercial search engine:
  ▶ What happens if Google discontinues API service?
  ▶ What happens if Google does something too smart or too commercial with the page ranks?
▶ Good for content-driven corpus building, problems with syntax/style/genre-based filtering.
▶ Good for building small, targeted-corpora (but see Sharoff’s – and Ciaramita’s? – work).
Cons

- We still rely on commercial search engines:
  - What happens if Google discontinues API service?
  - What happens if Google does something too smart or too commercial with the page ranks?
- Good for content-driven corpus building, problems with syntax/style/genre-based filtering.
- Good for building small, targeted-corpora (but see Sharoff’s – and Ciaramita’s? – work).
- Not for exploiting vastness of web-as-corpus directly.
Biting the bullet...

- Crawling, cleaning, annotating, managing and maintaining your own indexed version of the web.
Biting the bullet...

- Crawling, cleaning, annotating, managing and maintaining your own indexed version of the web.
- Obviously, the “ideal” solution.
Biting the bullet...

- Crawling, cleaning, annotating, managing and maintaining your own indexed version of the web.
- Obviously, the “ideal” solution.
- But obviously a lot of work!
Build your own search engine

- Crawling.
Build your own search engine

- Crawling.
- Post-processing (html/boilerplate stripping, language recognition, duplicate detection, “connected prose” recognition...).
Build your own search engine

- Crawling.
- Post-processing (html/boilerplate stripping, language recognition, duplicate detection, “connected prose” recognition...)
- Linguistic processing.
Build your own search engine

- Crawling.
- Post-processing (html/boilerplate stripping, language recognition, duplicate detection, “connected prose” recognition . . . )
- Linguistic processing.
- Categorization, meta-data.
Build your own search engine

- Crawling.
- Post-processing (html/boilerplate stripping, language recognition, duplicate detection, “connected prose” recognition...)
- Linguistic processing.
- Categorization, meta-data.
- Indexing.
Build your own search engine

- Crawling.
- Post-processing (html/boilerplate stripping, language recognition, duplicate detection, “connected prose” recognition...)
- Linguistic processing.
- Categorization, meta-data.
- Indexing.
- Interfaces.
The huge web-corpus of Clarke and collaborators

The huge web-corpus of Clarke and collaborators

- From initial seed set of 2392 (English?) educational URLs.
The huge web-corpus of Clarke and collaborators

- From initial seed set of 2392 (English?) educational URLs.
- No duplicates, not too many pages from same site.
The huge web-corpus of Clarke and collaborators

- From initial seed set of 2392 (English?) educational URLs.
- No duplicates, not too many pages from same site.
- No language filtering.
The huge web-corpus of Clarke and collaborators

- From initial seed set of 2392 (English?) educational URLs.
- No duplicates, not too many pages from same site.
- No language filtering.
- 53 billion words, 77 million documents.
The huge web-corpus of Clarke and collaborators

- From initial seed set of 2392 (English?) educational URLs.
- No duplicates, not too many pages from same site.
- No language filtering.
- 53 billion words, 77 million documents.
- (BNC has 100 million words; Google indexes 8 billion documents.)
The TOEFL synonym match test, again

- Target: *levied*; Candidates: *imposed, believed, requested, correlated.*
The TOEFL synonym match test, again

- Target: *levied*; Candidates: *imposed*, *believed*, *requested*, *correlated*. 
WMI takes the TOEFL again
Terra and Clarke 2003

- Performance on TOEFL synonym match task:
  - Average foreign test taker: 64.5%
  - Latent Semantic Analysis: 65.4%
  - WMI: 72.5%
  - Terra & Clarke’s WMI: 81.25%
WMI takes the TOEFL again
Terra and Clarke 2003

- Performance on TOEFL synonym match task:
  - Average foreign test taker: 64.5%
WMI takes the TOEFL again
Terra and Clarke 2003

- Performance on TOEFL synonym match task:
  - Average foreign test taker: 64.5%
  - Latent Semantic Analysis: 65.4%
WMI takes the TOEFL again
Terra and Clarke 2003

- Performance on TOEFL synonym match task:
  - Average foreign test taker: 64.5%
  - Latent Semantic Analysis: 65.4%
  - WMI: 72.5%
WMI takes the TOEFL again
Terra and Clarke 2003

- Performance on TOEFL synonym match task:
  - Average foreign test taker: 64.5%
  - Latent Semantic Analysis: 65.4%
  - WMI: 72.5%
  - Terra & Clarke’s WMI: 81.25%
Pros

- Independence from commercial search engines.
Pros

- Independence from commercial search engines.
- Precious, multi-purpose resource.
Pros

- Independence from commercial search engines.
- Precious, multi-purpose resource.
- In principle, you can do what you want with it.
Cons

- A lot of work.
Cons

- A lot of work.
- Resource-intensive.
Cons

- A lot of work.
- Resource-intensive.
- *In principle*, you can do what you want with it...
Cons

- A lot of work.
- Resource-intensive.
- *In principle*, you can do what you want with it...
- In practice, almost anything you want to do with a terabyte corpus will be extremely complicated to do.
Cons

- A lot of work.
- Resource-intensive.
- *In principle*, you can do what you want with it…
- In practice, almost anything you want to do with a terabyte corpus will be extremely complicated to do.
- Forget about the “do it yourself with a perl script” approach.
Outline

Introduction

Frequency estimates from search engines
  Web-based Mutual Information

The “linguists’ friendly” interfaces

Building your own web corpus
  Small corpora via search engine queries
  Thinking Big: The “real” Linguist’s Search Engine

Enter WaCky!
Enter WaCky!

- The **Web-as-Corpus kool ynitiative**.
Enter WaCky!

- The **Web-as-Corpus kool yniative**.
- [http://wacky.sslmit.unibo.it/](http://wacky.sslmit.unibo.it/)
Enter WaCky!

- The **Web-as-Corpus kool ynitiative**.
- [http://wacky.sslmit.unibo.it/](http://wacky.sslmit.unibo.it/)
- WaCky crowd: Marco, Massi, Silvia Bernardini, Stefan Evert, Bill Fletcher, Adam Kilgarriff...
Enter WaCky!

- The **Web-as-Corpus kool ynitiative**.
- [http://wacky.sslmit.unibo.it/](http://wacky.sslmit.unibo.it/)
- WaCky crowd: Marco, Massi, Silvia Bernardini, Stefan Evert, Bill Fletcher, Adam Kilgarriff…
- Yet Another Linguist’s Search Engine proposal (see also: Kilgarriff 2003, Fletcher 2004).
Enter WaCky!

- The **Web-as-Corpus** **kool yniitiative**.
- [http://wacky.sslmit.unibo.it/](http://wacky.sslmit.unibo.it/)
- WaCky crowd: Marco, Massi, Silvia Bernardini, Stefan Evert, Bill Fletcher, Adam Kilgarriff...
- Yet Another Linguist’s Search Engine proposal (see also: Kilgarriff 2003, Fletcher 2004).
- The WaCky philosophy: try to get something concrete out there very soon, so that other will feel motivated to contribute.
Enter WaCky!

- The **Web-as-Corpus** kool ynitiative.
- [http://wacky.sslmit.unibo.it/](http://wacky.sslmit.unibo.it/)
- WaCky crowd: Marco, Massi, Silvia Bernardini, Stefan Evert, Bill Fletcher, Adam Kilgarriff...
- Yet Another Linguist’s Search Engine proposal (see also: Kilgarriff 2003, Fletcher 2004).
- The WaCky philosophy: try to get something concrete out there very soon, so that other will feel motivated to contribute.
- 3 1-billion word corpora (English, German, Italian) by spring 2006.
Enter WaCky!

- The **Web-as-Corpus kool ynitiative**.
- [http://wacky.sslmit.unibo.it/](http://wacky.sslmit.unibo.it/)
- WaCky crowd: Marco, Massi, Silvia Bernardini, Stefan Evert, Bill Fletcher, Adam Kilgarriff...
- Yet Another Linguist’s Search Engine proposal (see also: Kilgarriff 2003, Fletcher 2004).
- The WaCky philosophy: try to get something concrete out there very soon, so that other will feel motivated to contribute.
- 3 1-billion word corpora (English, German, Italian) by spring 2006.
- Web interface(s) and an open source toolkit.
Enter WaCky! (cont.)

- We must learn from IR and massive dataset studies (e.g., near duplicate detection, fast retrieval)...
Enter WaCky! (cont.)

- We must learn from IR and massive dataset studies (e.g., near duplicate detection, fast retrieval)...
- but there are important differences, for example:
Enter WaCky! (cont.)

▶ We must learn from IR and massive dataset studies (e.g., near duplicate detection, fast retrieval)...
▶ but there are important differences, for example:
  ▶ We probably want all data, or perhaps random data, or even linguistically interesting data, not necessarily most relevant data.
Enter WaCky! (cont.)

- We must learn from IR and massive dataset studies (e.g., near duplicate detection, fast retrieval)...
- but there are important differences, for example:
  - We probably want all data, or perhaps random data, or even linguistically interesting data, not necessarily most relevant data.
  - We care about (linguistic) form at least as much as about content.
Enter WaCky! (cont.)

- We must learn from IR and massive dataset studies (e.g., near duplicate detection, fast retrieval)...
- but there are important differences, for example:
  - We probably want *all* data, or perhaps *random* data, or even *linguistically interesting* data, not necessarily *most relevant* data.
  - We care about (linguistic) *form* at least as much as about *content*.
- A new challenge in computational linguistics: *data* are not *given*. 
Enter WaCky! (cont.)

- Emphasis on:
Enter WaCky! (cont.)

▶ Emphasis on:
  ▶ Transparency;
Enter WaCky! (cont.)

- Emphasis on:
  - Transparency;
  - Stability;
Enter WaCky! (cont.)

- Emphasis on:
  - Transparency;
  - Stability;
  - Pre-processing;
Enter WaCky! (cont.)

▶ Emphasis on:
  ▶ Transparency;
  ▶ Stability;
  ▶ Pre-processing;
  ▶ Categorization and annotation;
Enter WaCky! (cont.)

- Emphasis on:
  - Transparency;
  - Stability;
  - Pre-processing;
  - Categorization and annotation;
  - (Also) automated access;
Enter WaCky! (cont.)

- Emphasis on:
  - Transparency;
  - Stability;
  - Pre-processing;
  - Categorization and annotation;
  - (Also) automated access;
  - Sophisticated query options.
Enter WaCky! (cont.)

- Emphasis on:
  - Transparency;
  - Stability;
  - Pre-processing;
  - Categorization and annotation;
  - (Also) automated access;
  - Sophisticated query options.

- Not so important:
Enter WaCky! (cont.)

- Emphasis on:
  - Transparency;
  - Stability;
  - Pre-processing;
  - Categorization and annotation;
  - (Also) automated access;
  - Sophisticated query options.

- Not so important:
  - Access speed;
Enter WaCky! (cont.)

- Emphasis on:
  - Transparency;
  - Stability;
  - Pre-processing;
  - Categorization and annotation;
  - (Also) automated access;
  - Sophisticated query options.

- Not so important:
  - Access speed;
  - Updating;
Enter WaCky! (cont.)

- Emphasis on:
  - Transparency;
  - Stability;
  - Pre-processing;
  - Categorization and annotation;
  - (Also) automated access;
  - Sophisticated query options.

- Not so important:
  - Access speed;
  - Updating;
  - Size;
Enter WaCky! (cont.)

- Emphasis on:
  - Transparency;
  - Stability;
  - Pre-processing;
  - Categorization and annotation;
  - (Also) automated access;
  - Sophisticated query options.

- Not so important:
  - Access speed;
  - Updating;
  - Size;
  - Content-driven relevance.
The WaCkodules: Where We Are At

- Seeding the Crawls: BNC/Google seeding experiments and Massi’s measures of randomness.
The WaCkodules: Where We Are At

- Seeding the Crawls: BNC/Google seeding experiments and Massi’s measures of randomness.
- Crawling: with Heritrix, the Internet Archive crawler.
The WaCkodules: Where We Are At

- Seeding the Crawls: BNC/Google seeding experiments and Massi’s measures of randomness.
- Crawling: with Heritrix, the Internet Archive crawler.
- Post-processing: current focus on duplicate detection.
The WaCkodules: Where We Are At

- Seeding the Crawls: BNC/Google seeding experiments and Massi’s measures of randomness.
- Crawling: with Heritrix, the Internet Archive crawler.
- Post-processing: current focus on duplicate detection.
- Linguistic annotation, meta-data: nothing yet.
The WaCkodules: Where We Are At

- Seeding the Crawls: BNC/Google seeding experiments and Massi’s measures of randomness.
- Crawling: with Heritrix, the Internet Archive crawler.
- Post-processing: current focus on duplicate detection.
- Linguistic annotation, meta-data: nothing yet.
- Indexing: Lucene vs. the newly open (!) IMS Corpus WorkBench.
Introduction
Frequency estimates from search engines
The “linguists’ friendly” interfaces
Building your own web corpus
Enter WaCky!

The WaCkodules: Where We Are At

- Seeding the Crawls: BNC/Google seeding experiments and Massi’s measures of randomness.
- Crawling: with Heritrix, the Internet Archive crawler.
- Post-processing: current focus on duplicate detection.
- Linguistic annotation, meta-data: nothing yet.
- Indexing: Lucene vs. the newly open (!) IMS Corpus WorkBench.
- Interfaces: work by Stefan Evert.
A few references