Web N-Grams as a Resource for Corpus Linguistics

Stefan Evert

English Computational Corpus Linguistics
Technische Universität Darmstadt, Germany
evert@linglit.tu-darmstadt.de

Heidelberg, 12.01.2012

Introduction

Google Web 1T 5-Grams

Scaling up because more data are better data for statistical NLP (Church and Mercer 1993)

- 1964: 1 million words (Brown Corpus)
- 1995: 100 million words (British National Corpus)
- 2003: 1,000+ million words (English Gigaword, WaCky)
- 2006: 1,000,000 million words (Google Web 1T 5-Grams)

The Google Web 1T 5-Gram database
Brants and Franz (2006)

- Not the full 1 trillion words of English Web text, but ...
- Frequency counts for bigrams, trigrams, 4-grams and 5-grams extracted from this corpus
  - thresholds: \( f \geq 200 \) for terms, \( f \geq 40 \) for n-grams
- Multiple compressed text files with total size of 24.4 GiB
- No linguistic pre-processing (case-folding, lemmatization, POS tagging, parsing, word sense disambiguation, ...)
- Little boilerplate cleanup (“from collectibles to cars”)

<table>
<thead>
<tr>
<th>word 1</th>
<th>word 2</th>
<th>word 3</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>supplement</td>
<td>depend</td>
<td>on</td>
<td>193</td>
</tr>
<tr>
<td>supplement</td>
<td>depending</td>
<td>on</td>
<td>174</td>
</tr>
<tr>
<td>supplement</td>
<td>depends</td>
<td>entirely</td>
<td>94</td>
</tr>
<tr>
<td>supplement</td>
<td>depends</td>
<td>on</td>
<td>338</td>
</tr>
<tr>
<td>supplement</td>
<td>derived</td>
<td>from</td>
<td>2668</td>
</tr>
<tr>
<td>supplement</td>
<td>des</td>
<td>coups</td>
<td>77</td>
</tr>
<tr>
<td>supplement</td>
<td>described</td>
<td>in</td>
<td>200</td>
</tr>
</tbody>
</table>

excerpt from file 3gm-0088.gz
Some applications of Google Web1T5

- broad-coverage word-level n-gram models (of course . . . )
  - machine translation
  - speech recognition
  - predictive typing
  - . . .
- replacement for Google API in knowledge mining tools
- spelling correction (Bergsma et al. 2009)
- linguistic steganography (Chang and Clark 2010)
- near-synonym choice (Islam and Inkpen 2010)
- prediction of fMRI neural activation (Mitchell et al. 2008)
- testbed for n-gram search engines and analysis software (Stein et al. 2010; Sekine and Dalwani 2010; Lin et al. 2010)
  - e.g. http://www.netspeak.org/ (University of Weimar)

Advantages of Google Web1T5

1. **Size**
   - more data are better data (Church and Mercer 1993)
   - three orders of magnitude larger than current Web corpora, four orders of magnitude larger than BNC
   - much better coverage of words and esp. phrases
   - data-driven NLP scales logarithmically (Banko and Brill 2001)

2. **Pre-compiled n-gram frequency data**
   - frequency counts for a trillion words of text need massive computing power and clever algorithms

Limitations of Google Web1T5

1. **Lack of linguistic annotation**
   - *part-time* is split into a trigram (*part, -*, *time*)
   - cannot search for *can/N* or verb-object combinations

2. **Frequency thresholds**
   - precise co-occurrence frequencies only for bigrams

3. **Lack of normalisation**
   - case-folding, deletion of non-words, numbers, URLs, . . .

4. **No indexing for interactive search**
   - queries require linear scan of many GiB of compressed text
   - no suitable open-source indexing software available

5. **Pre-compiled n-gram frequency data**
   - corpus linguists more interested in frequencies of patterns, association strength, collocations, distributional similarity
   - cannot use tagger, parser, . . . without original corpus data

6. **Web language** (sex, lolcats and advertising)
Web1T5 as a resource for corpus linguistics

1. What can Web1T5 do for corpus linguists?
   - frequencies of words and phrases
   - collocation analysis
   - distributional semantics

2. Software implementation
   - how to compute lemmatised frequencies, association scores and distributional similarity
   - challenges: efficiency, limitations of Web1T5

3. Evaluating the quality of Web1T5
   - anecdotal evidence and pet peeves
   - direct comparison of frequencies and association scores
   - task-based evaluation: multiword extraction and distributional semantics

Requirements

A Web1T5 indexing software for corpus linguists should

- be flexible and powerful enough to support queries on multi-word patterns, collocation analysis, distributional semantics, word frequency distributions, . . .
- be open-source (or at least available free of charge)
- be easy to install and run (no GPU computing . . .)
- run on commodity hardware (e.g. a €5000 server)
- be fast enough for occasional interactive exploration
- connect to other analysis tools (Excel, R, . . .)

Web1T5-Easy architecture

<table>
<thead>
<tr>
<th>word 1</th>
<th>word 2</th>
<th>word 3</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>supplement</td>
<td>depend</td>
<td>on</td>
<td>193</td>
</tr>
<tr>
<td>supplement</td>
<td>depending</td>
<td>on</td>
<td>174</td>
</tr>
<tr>
<td>supplement</td>
<td>depends</td>
<td>entirely</td>
<td>94</td>
</tr>
<tr>
<td>supplement</td>
<td>depends</td>
<td>on</td>
<td>338</td>
</tr>
<tr>
<td>supplement</td>
<td>derived</td>
<td>from</td>
<td>2668</td>
</tr>
<tr>
<td>supplement</td>
<td>described</td>
<td>in</td>
<td>77</td>
</tr>
</tbody>
</table>

- This looks very much like a relational database table
- So why not just put the data into an off-the-shelf RDBMS?
  - built-in indexing for quick access
  - powerful query language SQL
- I’m not the only one to come up with this idea . . .
  (Evert 2010; Lam 2010)
Web1T5-Easy architecture

<table>
<thead>
<tr>
<th>word</th>
<th>id</th>
<th>id 1</th>
<th>id 2</th>
<th>id 3</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>depend</td>
<td>6094</td>
<td>5095</td>
<td>6094</td>
<td>14</td>
<td>193</td>
</tr>
<tr>
<td>depending</td>
<td>3571</td>
<td>5095</td>
<td>3571</td>
<td>14</td>
<td>174</td>
</tr>
<tr>
<td>depends</td>
<td>3846</td>
<td>5095</td>
<td>3846</td>
<td>4585</td>
<td>94</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>5095</td>
<td>3846</td>
<td>14</td>
<td>338</td>
</tr>
<tr>
<td>on</td>
<td>14</td>
<td>5095</td>
<td>4207</td>
<td>27</td>
<td>2668</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>5095</td>
<td>2298</td>
<td>62481</td>
<td>77</td>
</tr>
<tr>
<td>supplement</td>
<td>5095</td>
<td>5095</td>
<td>1840</td>
<td>11</td>
<td>200</td>
</tr>
</tbody>
</table>

- Use numeric ID coding as in IR / large-corpus query engines
- More efficient to store, index and sort in RDBMS
- Frequency-sorted lexicon is beneficial for variable-length coding of integer IDs

Which RDBMS?

Requirements

- Web1T5-Easy is more than an interactive end-user GUI
- Preferably on dedicated RDBMS not shared with other users
- Indexing expensive → want to share pre-compiled database

My choice: SQLite [www.sqlite.org]

- Lightweight embedded SQL engine & RDBMS
- Database stored in single, platform-independent file
- Available for C, C++, Java, C#, Perl, Python, PHP, R, …

But it’s all SQL & Perl, so you can substitute any other RDBMS!

(for all other technical details see Evert 2010)

Database encoding procedure

Pre-processing (normalisation, filtering, …)  
⇓
Numeric ID coding & database insertion [1d 23h]  
⇓
Collapse duplicate rows (from normalisation) [6d 7h]  
⇓
Indexing of each n-gram position [3d 2h]  
⇓
Statistical analysis for query optimisation [not useful]  
⇓
Build database of co-occurrence frequencies [ca. 3d]  
⇓

211 GiB

Carried out in spring 2009 on quad-core Opteron 2.6 GHz with 16 GiB RAM — should be faster on state-of-the-art server with latest version of SQLite.

Querying the database

It’s easy to search the database for patterns like

association … Xal Y

with a “simple” SQL query:

```
SELECT w3, w4, SUM(f) AS freq FROM ngrams
WHERE w1 IN (SELECT id FROM vocab WHERE w='association')
AND w3 IN (SELECT id FROM vocab WHERE w LIKE '%al')
GROUP BY w3, w4 ORDER BY freq DESC;
```

Web1T5-Easy implements a more user-friendly query language:

association ? %al *
Web1T5-Easy demo

https://cogsci.uni-osnabrueck.de/~korpora/Web1T5/ (but currently broken)

Web1T5-Easy query performance

<table>
<thead>
<tr>
<th>Web1T5-Easy query</th>
<th>cold cache</th>
<th>warm cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>corpus linguistics</td>
<td>0.11s</td>
<td>0.01s</td>
</tr>
<tr>
<td>web as corpus</td>
<td>1.29s</td>
<td>0.44s</td>
</tr>
<tr>
<td>time of *</td>
<td>2.71s</td>
<td>1.09s</td>
</tr>
<tr>
<td>%ly good fun</td>
<td>181.03s</td>
<td>24.37s</td>
</tr>
<tr>
<td>[sit,sits,sat,sitting] * ? chair</td>
<td>1.16s</td>
<td>0.31s</td>
</tr>
<tr>
<td>* linguistics (association ranking)</td>
<td>11.42s</td>
<td>0.05s</td>
</tr>
<tr>
<td>university of * (association ranking)</td>
<td>1.48s</td>
<td>0.48s</td>
</tr>
</tbody>
</table>

(64-bit Linux server with 2.6 GHz AMD Opteron CPUs, 16 GiB RAM and fast local hard disk; based on timing information from the public Web interface.)

Approximating lemmatized frequency counts

- Web1T5 frequencies based on unnormalized word forms
- Web1T5-Easy can perform case-folding normalization during indexing (default)
- Approximate lemmatized frequency counts by morphological query expansion
  
<table>
<thead>
<tr>
<th>query</th>
<th>( \hat{f} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>hear sound</td>
<td>36,304</td>
</tr>
<tr>
<td>[hear,hears,heard,hearing]</td>
<td></td>
</tr>
<tr>
<td>[sound,sounds]</td>
<td>95,453</td>
</tr>
</tbody>
</table>

- lazy approach: use TreeTagger lexicon, or extract from BNC
- pooled frequency counts with SQL aggregates (GROUP BY)
Collocations

- Collocation: frequent co-occurrence within short span of up to 5 words (Firth 1957; Sinclair 1966, 1991)
  - plays important role in lexicography, corpus linguistics, language description, word sense disambiguation, . . .
  - collocation database is also a sparse representation of a distributional semantic model (term-term matrix)
- Web1T5 only provides co-occurrence frequencies for immediately adjacent bigrams (e.g. * day and day *)

- Approximate counts for distance \( n \) from \( n + 1 \)-gram table

  \[
  \text{day} ? ? * \quad \text{and} \quad * ? ? \text{day}
  \]

  ➠ quasi-collocations

Quasi-collocations database

- Web1T5-Easy: pre-compiled database of quasi-collocations
  - brute-force, multi-pass algorithm
  - runtime approx. 3 days on server with 16 GiB RAM
- Flexible collocational span L4, . . . , L1 / R1, . . . , R4
  - separate count for each collocate and position
  - co-occurrence frequency in user-defined span and association scores are calculated on the fly
  - benefits from tight integration of Perl & SQLite

- Standard association measures: \( X^2 \), \( G^2 \), \( t \), MI, Dice

Distributional semantics

- Distributional hypothesis (Harris 1954): meaning of a word can be inferred from its distribution across contexts
  - “You shall know a word by the company it keeps!” — (Firth 1957)

- Reality check: What is the mystery word?
  - He handed her her glass of XXXXX.
  - Nigel staggered to his feet, face flushed from too much XXXXX.
  - Malbec, one of the lesser-known XXXXX grapes, responds well to Australia’s sunshine.
  - I dined off bread and cheese and this excellent XXXXX.
  - The drinks were delicious: blood-red XXXXX as well as light, sweet Rhenish.

  ➠ XXXXX = claret

  ➠ all examples from BNC (carefully selected & slightly edited)
## Distributional Semantics

- A computer can (sometimes) do the same, with sufficient amounts of corpus data and full collocational profiles.

<table>
<thead>
<tr>
<th></th>
<th>get</th>
<th>see</th>
<th>use</th>
<th>hear</th>
<th>eat</th>
<th>kill</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$w_1$</td>
<td>$w_2$</td>
<td>$w_3$</td>
<td>$w_4$</td>
<td>$w_5$</td>
<td>$w_6$</td>
</tr>
<tr>
<td>knife</td>
<td>51</td>
<td>20</td>
<td>84</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>52</td>
<td>58</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>26</td>
</tr>
<tr>
<td>???</td>
<td>115</td>
<td>83</td>
<td>10</td>
<td>42</td>
<td>33</td>
<td>17</td>
</tr>
<tr>
<td>boat</td>
<td>59</td>
<td>39</td>
<td>23</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cup</td>
<td>98</td>
<td>14</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>pig</td>
<td>12</td>
<td>17</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>banana</td>
<td>11</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>18</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ \text{sim}(???, \text{knife}) = 0.770 \]

<table>
<thead>
<tr>
<th></th>
<th>get</th>
<th>see</th>
<th>use</th>
<th>hear</th>
<th>eat</th>
<th>kill</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$w_1$</td>
<td>$w_2$</td>
<td>$w_3$</td>
<td>$w_4$</td>
<td>$w_5$</td>
<td>$w_6$</td>
</tr>
<tr>
<td>knife</td>
<td>51</td>
<td>20</td>
<td>84</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>52</td>
<td>58</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>26</td>
</tr>
<tr>
<td>???</td>
<td>115</td>
<td>83</td>
<td>10</td>
<td>42</td>
<td>33</td>
<td>17</td>
</tr>
<tr>
<td>boat</td>
<td>59</td>
<td>39</td>
<td>23</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cup</td>
<td>98</td>
<td>14</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>pig</td>
<td>12</td>
<td>17</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>banana</td>
<td>11</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>18</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ \text{sim}(???, \text{pig}) = 0.939 \]

<table>
<thead>
<tr>
<th></th>
<th>get</th>
<th>see</th>
<th>use</th>
<th>hear</th>
<th>eat</th>
<th>kill</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$w_1$</td>
<td>$w_2$</td>
<td>$w_3$</td>
<td>$w_4$</td>
<td>$w_5$</td>
<td>$w_6$</td>
</tr>
<tr>
<td>knife</td>
<td>51</td>
<td>20</td>
<td>84</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>52</td>
<td>58</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>26</td>
</tr>
<tr>
<td>???</td>
<td>115</td>
<td>83</td>
<td>10</td>
<td>42</td>
<td>33</td>
<td>17</td>
</tr>
<tr>
<td>boat</td>
<td>59</td>
<td>39</td>
<td>23</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cup</td>
<td>98</td>
<td>14</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>pig</td>
<td>12</td>
<td>17</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>27</td>
</tr>
<tr>
<td>banana</td>
<td>11</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>18</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ \text{sim}(???, \text{cat}) = 0.961 \]
Distributional semantics with Web1T5

- Basis of distributional semantic model (DSM): term-term **co-occurrence matrix** of collocational profiles
  - very sparse: e.g. 250k × 100k matrix with 24.2 billion cells, but only 245.4 million cells (≈ 1%) have nonzero values
- We’ve already computed collocational profiles
  - 32 GiB collocations database = sparse co-occurrence matrix
  - export for further processing with 250k most frequent word forms as target terms (rows) and 100k mid-frequency word forms as feature terms (columns)
- DSM implemented in R (experimental **wordspace** package)
  - column-compressed sparse matrix
  - t-score feature weights with sqrt transformation
  - cosine similarity measure (converted to angle = distance)
  - dim. reduction with randomized SVD (Halko et al.
  - needs 20 GiB RAM and half a day (or else a weekend)

DSM with Web1T5: nearest neighbours

**Neighbours of linguistics** (cosine angle):
- sociology (24.6), sociolinguistics (24.6), criminology (29.5), anthropology (30.8), mathematics (31.2), phonetics (33.1), phonology (33.2), philology (33.2), literatures (33.5), gerontology (35.3), proseminar (35.5), geography (35.8), humanities (35.9), archaeology (35.9), science (36.5), . . .

**Neighbours of spaniel** (cosine angle):
- terrier (23.0), schnauzer (26.5), pinscher (27.0), weimaraner (28.3), keeshond (29.1), pomeranian (29.4), pekingese (29.6), bichon (30.1), vizsla (30.5), labradoodle (30.6), apso (31.1), spaniels (32.0), frise (32.0), yorkie (32.1), sheepdog (32.3), dachshund (32.4), retriever (32.7), whippet (32.9), havanese (33.1), westie (33.5), mastiff (34.6), dandie (34.7), chihuahua (34.9), dalmation (35.0), elkhound (35.0), . . .

DSM with Web1T5: semantic map

(data from ESSLLI 2008 shared task on concrete noun categorization)

Semantic map (Web1T5)

Evaluating the quality of Web1T5
### Anecdotal Evidence

**Insufficient boilerplate removal & de-duplication:**

- from * to *
- from collectibles to cars: 9,443,572
- from collectables to cars: 8,844,838
- from time to time: 5,678,941
- from left to right: 793,957
- from start to finish: 749,705
- from a to z: 572,917
- from year to year: 486,669
- from top to bottom: 372,935

"Traditional" Web corpora are better:

- Google ≈ 121,000,000 hits
- Google.de ≈ 119,600,000 hits
- Web 1T 5-Grams: 18,288,410 hits
- ukWaC: 3 hits
- BNC: 0 hits

---

### Linguistic Evaluation of Web 1T 5-Grams

- Compare Web1TS with British National Corpus (Aston and Burnard 1998) and ukWaC Web corpus (Baroni *et al.* 2009)
- **Method 1:** Direct comparison of frequency counts
  - expect good correlation, but better coverage from Web1TS
  - Baroni *et al.* (2009) use a similar approach to compare their ukWaC Web corpus against the BNC
  - same for association scores (bigrams, collocations)
- **Method 2:** Task-based evaluation
  - do applications benefit from the Web1TS data?
  - multiword extraction: English particle verbs (VPC, Baldwin 2008) and light verb constructions (LVC, Tu and Roth 2011)
  - standard shared tasks for distributional models, such as TOEFL synonyms and WordSim-353 (Finkelstein *et al.* 2002)

---

### Comparison of frequency counts

- Scatterplots of (log) frequencies in different corpora
  - BNC vs. ukWaC vs. Web 1T 5-Grams
  - only include items that occur in all three corpora
    - not interested in coverage / idiosyncrasies
  - correlation \( r \) from regression model \( f_{ukWaC} \sim \beta \cdot f_{BNC} \) etc.
- **Test data sets**
  - Basic English words (lemmatised vs. word form in Web1TS)
  - inflected forms of Basic English words
  - binary compound nouns extracted from WordNet 3.0
  - English particle verbs from VPC task (adjacent bigrams)
  - English particle verbs (co-occurrence in L0/R3 window)
Comparison of frequency counts

Basic English (lemmatised vs. word forms)
(dashed lines indicate acceptable frequency difference within one order of magnitude)

Basic English (inflected forms)
(dashed lines indicate acceptable frequency difference within one order of magnitude)

Comparison of frequency counts

Binary compound nouns (WordNet)
(dashed lines indicate acceptable frequency difference within one order of magnitude)

Particle verbs (adjacent bigrams)
(dashed lines indicate acceptable frequency difference within one order of magnitude)
Evaluation on English VPC extraction task
(Baldwin 2008)

- English verb-particle constructions (VPC) consisting of
  head verb + one obligatory prepositional particle
    - *hand in*, *back off*, *wake up*, *set aside*, *carry on*, ...
- Data set of 3,078 candidate VPC types
  - extracted from written part of BNC with combination of
tagger-, chunker-, and parser-based methods
- Manually annotated as compositional / non-compositional
  - baseline: 14.3% non-compositional VPC (440 / 3078)
  - compositional: *carry around*, *fly away*, *refer back*, ...
  - further distinction of transitive/intransitive VPC not used
- Evaluation: candidate ranking from BNC/ukWaC/Web1TS
  - surface co-occurrence (L0,R3) + POS filter (except Web1T5)
  - Web1T5 without/with morphological expansion
  - using best association measure for each corpus
    (X^2, X^2, t, G^2, Dice)

Evaluation Multiword Extraction

Do association measures scale badly?
fitted to BNC data

- Precision (%)
- Recall (%)
- BNC (miss: 185)
- ukWaC (miss: 157)
- Web1TS (miss: 3)
- Web1TS lemma (miss: 3)
- BNC untagged (miss: 39)
Do association measures scale badly?
fitted to Web1T5 data

What’s wrong with Web1T5 quasi-collocations?

Evaluation on English LVC extraction task
(Tu and Roth 2011)

- English light verb constructions (LVC) consisting of verb (semantically bleached) + object noun (often deverbal)
  > take a walk, give a speech, have a look, make a call, ...
- Data set of 2,162 candidate LVC tokens
  > extracted from BNC with parser and various heuristics (e.g. object NP must have deverbal head noun)
  > only for verbs do, get, give, have, make and take
- Manually annotated as LVC / non-LVC in sentence context
  > reduced to 891 verb + head noun types for this experiment
  > type considered a LVC if at least 50% of its tokens are LVC
  > baseline: 39.2% LVC (349 / 891 candidate types)
- Evaluation: candidate ranking from BNC/ukWaC/Web1T5
  > surface co-occurrence (L3,R3) + POS filter (except Web1T5)
  > association measure: $G^2$ with POS filter, MI without
Evaluation on English LVC extraction task
(Tu and Roth 2011)

- Distributional semantic model built from Web1T5 can be evaluated in various shared tasks (e.g. ESSLLI 2008)

- Here: direct comparison with semantic similarity ratings (WordSim-353, Finkelstein et al. 2002)
  - 353 noun-noun pairs with "relatedness" ratings
  - rated on scale 0-10 by 16 test subjects
  - closely related: money/cash, soccer/football, type/kind, ...
  - unrelated: king/cabbage, noon/string, sugar/approach, ...

- Correlation with DSM similarity in BNC/Wikipedia/Web1T5
  - DSM parameters: term-term matrix, (L2,R2) surface context, \( T \) weighting, cosine similarity, SVD to 300 dimensions
  - lemma vs. POS-disambiguated lemma on BNC and Wikipedia
  - word forms on Web1T5

Comparison of association scores for English LVC
Evaluating distributional similarity in Web1T5
correlation with human relatedness ratings (Finkelstein et al. 2002)

![Scatter plots showing cosine similarity between BNC, Wikipedia, and Web1T5 corpora](image)

**Work in progress**

- Find out what's really wrong with the Web 1T 5-grams
  - qualitative error analysis: which words and pairs are off?
  - further experiments on scaling of association measures, direct comparison of frequencies and association score, etc.
  - esp. usefulness of morphological expansion
  - linguistic quality of Web data (topics, slang, ...)

- Software improvements (Web1T5-Easy 2.0)
  - adapt to Web1T5 European edition (Brants and Franz 2009)
  - better customisation (e.g. normalisation, tagged data)
  - consistent Unicode support, more flexible Web GUI
  - include distributional model in open-source code

- Partial POS-tagging and lemmatisation of n-grams possible?

**That's all folks!**

http://webascorpus.sf.net/Web1T5-Easy/

Try the online demo at http://cogsci.uos.de/~korpora/Web1T5/
— currently offline —

Thanks for listening!
References II


Evert, Stefan (2010). Google Web 1T5 n-grams made easy (but not for the computer). In Proceedings of the 6th Web as Corpus Workshop (WAC-6), Los Angeles, CA.


References III


Lin, Dekang; Church, Kenneth; Ji, Heng; Sekine, Satoshi; Yarowsky, David; Bergsma, Shane; Patil, Kailash; Pitler, Emily; Lathbury, Rachel; Rao, Vikram; Dalwani, Kapil; Narsale, Sushant (2010). New tools for web-scale n-grams. In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010), Valletta, Malta. European Language Resources Association (ELRA).


References IV


