Corpus Linguistics (L415/L615)
Collocations, part 2: Practicalities

Markus Dickinson
Department of Linguistics, Indiana University
Fall 2015
Working with collocations

One question:

- What significant collocations are there that start with the word *sweet*?
- Specifically, what nouns tend to co-occur after *sweet*?

What do your intuitions say?
We have various options for calculating collocations

- Web interfaces
- Write our own program
- N-gram Statistics Package
  (http://www.d.umn.edu/~tpederse/nsp.html)
- UCS Toolkit (http://www.collocations.de)
Web interface

BYU interface

Can work with an online concordancer of the BNC, http://corpus.byu.edu/bnc/

1. Enter *sweet* in the Search String box.
   - Enter * in the Collocate box
   - Change the Sort option to be done by Relevance
     - This calculates & sorts collocates by MI scores

2. On the left side, check Compare Words
   - Enter *sweet* and some other word (e.g., *sour*)
   - This compares the collocates between the two words
After searching for *sweet*, let’s play with the Collocations options ...

Some other exercises:

- Search for *Christmas*. What is the most frequent word that collocates with it?
- What happens when you change the metric?
- What words collocate with *potato*? What words collocate with *couch potato*?
- Search for *Christmas* and *tree* in a 3-word window. Are there any occurrences where the words are not adjacent?
Perl scripting

We could write a Perl script to do the following:

1. Read in a corpus file (could be changed to read over a directory of files, if need be)
2. Store unigram and bigram counts as it reads the file in
3. Loop over all bigrams
4. For each bigram, calculate some metric (e.g., pointwise mutual information)
N-grams Statistics Package (NSP)

“The Ngram Statistics Package (NSP) is a suite of programs that aids in analyzing Ngrams in text files.” (from README)

▶ http://www.d.umn.edu/~tpederse/nsp.html

Two main files:

▶ `count.pl`: takes regular text files and generates a list of ngrams & their frequencies

▶ `statistic.pl`: takes ngram lists (output from `count.pl`) & runs a measure of association

Some example cases uses found here:
http://search.cpan.org/~tpederse/Text-NSP/doc/USAGE.pod

▶ See also the README which comes with the software
N-grams Statistics Package (NSP)

Other programs:

- **rank.pl**: takes 2 output files from statistic.pl & compares the orderings
- **kocos.pl**: $k^{th}$ order co-occurrences of words (e.g., due to *New York* & *New Jack*, *York* & *Jack* are 2$^{nd}$-order co-occurrences)
- **combig.pl**: ignores order (e.g., *fine wine* & *wine fine* treated the same)
- **huge-count.pl**: for larger corpora

You can type the name of a program with --help to see the options & usage notes
NSP package

Counting: `count.pl`

`leaves-of-grass.txt` is a plain text file (containing Walt Whitman’s *Leaves of Grass*)

Basic bigram counting:

```
> count.pl leaves.cnt leaves-of-grass.txt

> more leaves.cnt
152454
,<>the<>1290 18083 8976
of<>the<>1247 4237 8976
,<>I<>1053 18083 2933
,<>The<>978 18083 1309
,<>and<>831 18083 4866
in<>the<>569 1769 8976
```

You can also change the window size with `-window`
Count bigrams occurring 5 or more times & storing a histogram

```bash
> count.pl -frequency 5 -hist leaves-5.hist \ leaves-5.cnt leaves-of-grass.txt
```

```bash
> more leaves-5.hist
Total ngrams = 152454
Number of n-grams ... 1 time(s) = 60497 (39.68 percent)
Number of n-grams ... 2 time(s) = 8524 (11.18 percent)
Number of n-grams ... 3 time(s) = 3079 (6.06 percent)
Number of n-grams ... 4 time(s) = 1443 (3.79 percent)
```

A list of stop words to exclude can be given and indicated with -stop
Count trigrams:

> count.pl --ngram 3 leaves.tri leaves-of-grass.txt

> more leaves.tri
152453
,<>and<>the<>198 18083 4866 8976 831 1276 483
,<>I<>see<>171 18083 2933 394 1053 185 291
I<>see<>the<>117 2933 394 8976 291 365 132
I<>do<>not<>71 2933 217 780 75 173 129
Collocations: statistic.pl

Score bigram lists: log-likelihood ratios

- Note that we can restrict output by score or frequency

```shell
> statistic.pl -score 6.00 -frequency 5 ll.pm \ leaves.ll leaves.cnt
```

```shell
> more leaves.ll
152454
,<>The<>1 2816.6039 978 18083 1309
of<>the<>2 2417.4401 1247 4237 8976
I<>see<>3 1879.5817 291 2933 394
,<>And<>4 1416.4393 442 18083 537
...
I<>do<>43 320.1055 75 2933 217
these<>States<>44 318.0868 34 244 96
```
Score bigram list, keeping only top 10: dice coefficient

> statistic.pl -rank 10 dice.pm leaves.dice leaves.cnt
> more leaves.dice
...
En<>Masse<>1 1.0000 3 3 3
outer<>authorities<>1 1.0000 2 2 2
restrictions<>whatsoever<>1 1.0000 2 2 2
boot<>soles<>1 1.0000 2 2 2
AGREE<>THAT<>1 1.0000 2 2 2
pell<>mell<>1 1.0000 2 2 2
...

NB: there happen to be a lot of ties here
Collocations: statistic.pl

Score bigram lists, creating a formatted report: pointwise mutual information

```bash
> statistic.pl -format pmi.pm -precision 4 \ leaves.report leaves.cnt
> more leaves.report
Total sample size = 152454
```

<table>
<thead>
<tr>
<th>N-gram</th>
<th>Rank</th>
<th>PMI</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>favorite&lt;&gt;pupil</td>
<td>1</td>
<td>17.2180</td>
<td>1 1 1</td>
</tr>
<tr>
<td>EVEN&lt;&gt;IF</td>
<td>1</td>
<td>17.2180</td>
<td>1 1 1</td>
</tr>
<tr>
<td>foundest&lt;&gt;verifies</td>
<td>1</td>
<td>17.2180</td>
<td>1 1 1</td>
</tr>
<tr>
<td>fetid&lt;&gt;carbon</td>
<td>1</td>
<td>17.2180</td>
<td>1 1 1</td>
</tr>
<tr>
<td>stonecrop&lt;&gt;mixt</td>
<td>1</td>
<td>17.2180</td>
<td>1 1 1</td>
</tr>
<tr>
<td>impartially&lt;&gt;enfoldest</td>
<td>1</td>
<td>17.2180</td>
<td>1 1 1</td>
</tr>
<tr>
<td>racers&lt;&gt;racing</td>
<td>1</td>
<td>17.2180</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>
Available bigram measures:

- Dice Coefficient (dice)
- Fishers exact test - left sided (left)
- Fishers exact test - right sided (right)
- Fishers twotailed test - right sided (twotailed)
- Jaccard Coefficient (jaccard)
- Log-likelihood ratio (ll)
- Mutual Information (tmi)
- Odds Ratio (odds)
- Pointwise Mutual Information (pmi)
- Phi Coefficient (phi)
- Pearson’s Chi Squared Test (x2)
- Poisson Stirling Measure (ps)
- T-score (tscore)
Available trigram/4-gram measures

The available trigram measures are:

- Log-likelihood ratio (ll)
- Mutual Information (tmi)
- Pointwise Mutual Information (pmi)
- Poisson Stirling Measure (ps)

The only available 4-gram measure is:

- Log-likelihood ratio (ll)

Any of these measures can be used as follows, where XXXX is the name of the measure:

- statistic.pl XXXX output.txt input.txt
rank.pl “computes the Spearman’s rank correlation coefficient on the Ngrams that are common to both files” (from README)

▶ Give it two collocation files to compare

Here, we compare LL and PMI measures to a precision of 3 digits:

> rank.pl -precision 3 leaves.dice leaves.ll
Rank correlation coefficient = 0.487
k\textsuperscript{th} order co-occurrences: \texttt{kocos.pl}

\begin{verbatim}
> kocos.pl --literal soul --order 2 \ --trace soul.trace leaves.ll > soul.k2
> more soul.k2
own<>
in<>
poems<>
body<>
life<>
love<>
eyes<>
> more soul.trace
soul->my->own
soul->my->in
soul->my->poems
soul->my->body
...
\end{verbatim}
Another nice package is the UCS toolkit
  ▶ http://www.collocations.de/software.html

There are interfaces for both Perl and R
  ▶ The R interface provides more graphical output
  ▶ Tutorials & thorough documentation are available for both