Collocations

HAP/LAP. Corpus Linguistics.

Frequency lists and beyond

- Key insight: frequency lists are very helpful.
- **But**: top 10 frequent words in any corpora are the same.
- Exploit frequency data to get deeper insights.
  - Collocations.

Collocations

- What are collocations?
  - systematic co-occurrence of words in use.
  - words which are likely to occur in the context of another.
- Usually within a window of words.

MLWE

- Collocation is a type of Multi-Word Expression (MLWE)
- What is a MLWE?
  - Lexical unit larger than a word
  - Include the range of phenomena
  - Many terms (with slightly different semantics)
    - *chunk, cliché, collocation, extended lexical unit, fixed expression, formulaic sequence, idiom, idiomatic expression, lexical/lexicalized phrase, multi-word unit, phraseme, phraseologism, phraseological unit, phrasal lexical item, phrasal lexeme, prefabricated chunk, prefab*

MLWE

- Many types of MLWEs
  - Compounds: “disk drive”
  - Phrasal verbs: “make up”
  - Other phrases: “bacon and eggs”
- May be several words long
  - “international best practice”
- May be discontinuous:
  - “make [something] up”

MLWE: Compositionality

- MLWEs are often not *compositional*
  - An NLP expression is *compositional* if the meaning of the expression can be predicted from the meaning of the parts.
- Degrees of compositionality:
  - "fish and chips" (collocation): fully compositional, non-idiomatic.
  - "strong tea" (collocation) but *strong* used in a slightly different manner.
  - "black market": one or several words are idiomatic.
  - "red herring", "kick the bucket" (idioms): fully opaque, non-compositional.

MLWEs: more criteria

- **Non-substitutability**: We cannot substitute other words for the components of a MLWE, even if they have the same meaning in the context.
    - "Fast foot" $\Rightarrow$ "Quick food" ?
    - "White wine" $\Rightarrow$ "Yellow wine" ?
- **Non-modifiability**: many MLWEs can not be freely modified with additional lexical material or through grammatical transformations.
    - "The cat has got your tongue" $\Rightarrow$
      "The large, furry cat has your tongue" or
      "The cats have your tongues"

MLWEs and terminology

- Overlap between MLWEs and notions like
  - *term*
  - *technical term*
  - *terminological term*
- Usually, used when collocations are extracted from technical domains
  - terminology extraction

MLWEs: applications

- Statistical NLP (such as SMT)
  - word translates differently according to MLWE it occurs in
- Information Retrieval (IR)
  - index only "interesting" phrases
  - language models
- Lexicographers
  - frequent ways a word is used.
  - multiword detection (and inclusion in dictionaries)

Collocations: hands on

- Simple case: how to identify **contiguous, two words collocations**.
- First idea: find the most common two word sequences in text (**bigrams**)

$C(w^1, w^2)$	w^1	W^2
80,871	of	the
58,841	in	the
26,430	to	the
21,842	on	the
...	...	...

- The table (taken from Manning and Shütze, 1999) shows bigram frequency counts.
- Do not capture the collocations present in the text.
- Frequent bigrams represent common syntactic constructions.
- Many function words: grammatical words which always occur in sentences.

Collocations: hands on

Obtaining bigrams (`bigrams.py`)

Obtain bigrams from gutenberg corpus. Create a program to output bigrams:

- First line is total counts:
  N_bigrams TAB *N_tokens*
- Then, one line per bigram:
  wordA TAB *wordB* TAB *freq_big* TAB *freq_A* TAB *freq_B*

Example:

```
2447758    2543994
,    and 41331    192338   78770
of   the 18911    70040    125730
in   the 9793     31874    125730
;    and 7589     27837    78770
and  the 6432     78770    125730
```

Collocations: hands on

- **Problem**: Frequent bigrams give almost no information
- Workarounds:
  1. Take into account individual word frequencies. Measure association by chance.
  2. Filter collocations according to external factor.
     - Use POS tags to get, for instance, *adjective noun*, *noun noun*, etc.

Collocations: beyond frequencies

- There are many techniques to overcome the limitations of frequency.
- Hypothesis testing: whether two words occur more frequently than by chance
  - *t* test
  - chi square
  - likelihood ratios
- Mutual information (pointwise mutual information, *pmi*)
  - information theoretically motivated.
  - measure of the variable's mutual dependence.
- Dice coefficient

Contingency table

	$V = v$	$V \neq v$
$U = u$	O_{11}	O_{12}
$U \neq u$	O_{21}	O_{22}

- Count how many times the words u and v appear together,
- and also how many times each word occur by its own.
- For instance, for the words "box" and "black"

	$V = box$	$V \neq box$
$U = black$	123 *black box*	13,168 *e.g. black house*
$U \neq black$	1,810 *e.g white box*	4,951,883 *e.g. white house*

- We can derive many association measures using this matrix.

Contingency table: marginal frequencies

	$V = box$	$V \neq box$	
$U = black$	O_{11}	O_{12}	f_A
$U \neq black$	O_{21}	O_{22}	$f_{\neq A}$
	f_B	$f_{\neq B}$	N

- N = number of words in the corpus.
- Observed frequencies:
  - $f_A = O_{11} + O_{12}$ $\quad f_{\neq A} = O_{21} + O_{22}$
  - $f_B = O_{11} + O_{21}$ $\quad f_{\neq B} = O_{12} + O_{22}$
- Expected frequencies:
  - $E_{11} = \frac{f_A f_B}{N}$ $\quad E_{12} = \frac{f_A f_{\neq B}}{N}$
  - $E_{21} = \frac{f_{\neq A} f_B}{N}$ $\quad E_{22} = \frac{f_{\neq A} f_{\neq B}}{N}$

Association measures

Student t test	$\frac{O_{11}-E_{11}}{\sqrt{O_{11}}}$	
chi square	$\frac{N(O_{11}-E_{11})^2}{E_{11}E_{22}}$	$\frac{N(f_{AB}-\frac{f_A f_B}{N})^2}{f_A f_B f_{\neq A} f_{\neq B}}$
log likelihood	$2\sum_{ij} O_{ij} \log \frac{O_{ij}}{E_{ij}}$	omit $O_{ij}=0$ from formula
pmi	$\log \frac{p(A,B)}{p(A)p(B)} = \log \frac{O_{11}}{E_{11}}$	$= \log(N \frac{f_{AB}}{f_A f_B})$
Dice	$\frac{2O_{11}}{f_A+f_B}$	$= \frac{2f_{AB}}{f_A+f_B}$
Jaccard	$\frac{O_{11}}{O_{11}+O_{12}+O_{21}}$	

Collocations: hands on

Exercise (`pmi.py`)

Using bigram file,calculate *pmi* association measures. Discard bigrams whose frequency **is below 3**.

Additional issues

- Normalization: case insentitive, etc.
- Filter out non interesting words:
    - Named Entities, numbers, etc.

Using trigrams

- PMI for trigrams (as Perl NSP package)

  $$pmi(w_1, w_2, w_3) = 2 \log N_t + \log O_{111} - \log O_{1pp} - \log O_{p1p} - \log O_{pp1}$$

  where

  - N_t: total number of trigrams.
  - O_{111}: (w_1, w_2, w_3) trigram count.
  - O_{1pp}: number of trigrams starting with w_1.
  - O_{p1p}: number of trigrams starting with w_2.
  - O_{pp1}: number of trigrams starting with w_3.

Collocations: filter by POS

Tag Pattern	Example
A N	*linear function*
N N	*regression coefficients*
A A N	*Gaussian random variable*
A N N	*cumulative distribution function*
N A N	*mean squared error*
N N N	*class probability function*
N P N	*degrees of freedom*

- (Justeson and Katz, 1995): pass candidate phrases through a POS filter that are likely to be phrases. See patterns above.

Collocations: filter by POS

$C(w^1, w^2)$	w^1	w^2	Tag Pattern
11487	New	York	A N
7261	United	States	A N
5412	Los	Angeles	N N
3301	last	year	A N
3191	Saudi	Arabia	N N
2699	last	week	A N
2514	vice	president	A N
2378	Persian	Gulf	A N
2161	San	Francisco	N N
...	...	...	...

- The table shows the most highly ranked phrases after applying the filter.
- Surprisingly good results: only two are compositional ("last year", "last week")

Collocations: filter by POS

Exercise (`bigrams_pos.py`)

Obtain most frequent Adj Noun and Noun Noun bigrams from gutenberg corpus.