

Corpus Linguistics (L615)

Application #2: Collocations

Markus Dickinson

Department of Linguistics, Indiana University
Spring 2009

Corpora for lexicography

- ▶ Can extract authentic & typical examples, with frequency information
- ▶ With sociolinguistic meta-data, can get an accurate description of usage and, with monitor corpora, its change over time
- ▶ Can complement intuitions about meanings

The study of loanwords, for example, can be bolstered by corpus studies

Collocations

Defining a collocation
Krishnamurthy

Calculating
collocations

Practical work

Collocations are characteristic co-occurrence patterns of two (or more) lexical items

- ▶ Tend to occur with greater than random chance
- ▶ The meaning tends to be more than the sum of its parts

These are extremely hard to define by intuition:

- ▶ Pro: Corpora have been able to reveal connections previously unseen
- ▶ Con: It's not always clear what the theoretical basis of collocations are

Collocations

Defining a collocation
Krishnamurthy

Calculating
collocations

Practical work

A **colligation** is a slightly different concept:

- ▶ collocation of a node word with a particular class of words (e.g., determiners)

Colligations often create “noise” in a list of collocations

- ▶ e.g., *this house* because *this* is so common on its own, and determiners appear before nouns
- ▶ Thus, people sometimes use stop words to filter out non-collocations

“People disagree on collocations”

- ▶ Intuition does not seem to be a completely reliable way to figure out what a collocation is
- ▶ Many collocations are overlooked: people notice unusual words & structures, but not ordinary ones

But what your collocations are depends on exactly how you calculate them

- ▶ There is some notion that they are more than the sum of their parts

So, how can we practically define a collocation? ...

What a collocation is

Collocations are expressions of two or more words that are in some sense conventionalized as a group

- ▶ *strong tea* (cf. ??*powerful tea*)
- ▶ *international best practice*
- ▶ *kick the bucket*

In examining collocations, we are placing an importance on the context: “You shall know a word by a company it keeps” (Firth 1957)

- ▶ In other words, there are lexical properties that more general syntactic properties do not capture

This slide and the next 3 adapted from Manning and Schütze (1999), *Foundations of Statistical Natural Language Processing*

Prototypically, collocations meet the following criteria:

- ▶ Non-compositional: meaning of *kick the bucket* not composed of meaning of parts
 - ▶ More subtly: *red/white hair* sort of composable, but the color of *red/white* here is different than usual
- ▶ Non-substitutable: *orange hair* just as accurate as *red hair*, but we don't say it
- ▶ Non-modifiable: often we cannot modify a collocation, even though we normally could modify one of those words: ??*kick the red bucket*

The previous properties are good tests, but hard to verify with corpus data

(At least) two tests we can use with corpora:

- ▶ Is the collocation translated word-by-word into another language?
 - ▶ e.g., Collocation *make a decision* is not translated literally into French
- ▶ Do the two words co-occur more frequently together than we would otherwise expect?
 - ▶ e.g., *of the* is frequent, but both words are frequent, so we might expect this

Collocations come in different guises:

- ▶ Light verbs: verbs convey very little meaning but must be the right one:
 - ▶ *make a decision* vs. **take a decision*, *take a walk* vs. **make a walk*
- ▶ Phrasal verbs: main verb and particle combination, often translated as a single word:
 - ▶ *to tell off*, *to call up*
- ▶ Proper nouns: slightly different than others, but each refers to a single idea (e.g., *Brooks Brothers*)
- ▶ Terminological expressions: technical terms that form a unit (e.g., *hydraulic oil filter*)

Semantic prosody = “a form of meaning which is established through the proximity of a consistent series of collocates” (Louw 2000)

- ▶ These are typically negative: e.g., *peddle*, *ripe for*, *get ONESELF VERBED*
- ▶ The idea is that you can tell the semantic prosody of a word by the types of words it frequently co-occurs with

This type of co-occurrence often leads to general semantic preferences

- ▶ e.g., *utterly*, *totally*, etc. typically have a feature of ‘absence or change of state’

Collocation: from *silly ass* to lexical sets

Krishnamurthy 2000

Firth 1957: “You shall know a word by the company it keeps”

- ▶ Collocational meaning is a *syntagmatic* type of meaning, not a conceptual one
- ▶ e.g., in this framework, one of the meanings of *night* is the fact that it co-occurs with *dark*

An example is that *ass* is associated with a set of adjectives (think of *goose* if you prefer)

- ▶ *silly, obstinate, stupid, awful*
- ▶ We can see a **lexical set** associated with this word

Lexical sets and collocations can vary across genres, subcorpora, etc.

Notes on a collocation's definition

Krishnamurthy 2000

We often look for words which are adjacent to make up a collocation, but this is not always true

- ▶ e.g., *computers run*, but these 2 words may only be in the same proximity.

We can also speak of upward/downward collocations:

- ▶ *downward*: involves a more frequent node word A with a less frequent collocate B
- ▶ *upward*: weaker relationship, tending to be more of a grammatical property

Where collocations fit into corpus linguistics:

1. Pattern recognition: recognize lexical and grammatical units
2. Frequency list generation: rank words
3. Concordancing: observe word behavior
4. Collocations: take concordancing a step further ...

Calculating collocations

(The slides from here on out are based on Manning & Schütze (M&S) 1999)

The simplest thing to do to find collocations is to use frequency counts: two words appearing together a lot are a collocation

The problem is that we get lots of uninteresting pairs of function words (M&S 1999, table 5.1)

$C(w_1, w_2)$	w_1	w_2
80871	of	the
58841	in	the
26430	to	the
21842	on	the

To remove frequent pairings which are uninteresting, we can use a POS filter (Justeson and Katz 1995)

- ▶ only examine word sequences which fit a particular part-of-speech pattern:

A N, N N, A A N, A N N, N A N, N N N, N P N

A N *linear function*

N A N *mean squared error*

N P N *degrees of freedom*

- ▶ Crucially, all other sequences are removed

P D *of the*

MV V *has been*

Some results after tag filtering (M&S 1999, table 5.3)

$C(w_1, w_2)$	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

⇒ Fairly simple, but surprisingly effective

- ▶ This would need to be refined to handle verb-particle collocations
- ▶ Also, kind of inconvenient to write out patterns you want

Longer distance connections

Two words may commonly go together, but they may not be strictly collocational, i.e., they may not be right next to each other, as in *knock* and *door*:

- (1) she knocked on his door
- (2) they knocked at the door
- (3) 100 women knocked on Donaldson's door
- (4) a man knocked on the metal front door

So, how can we tell if they're related?

Offsets: mean & variance

Generally, words that go together appear near each other, so we can examine the offset between the two words, *knocked* and *door*

- ▶ Mean, or average offset: $\bar{x} = \frac{\sum_i d_i}{n}$, where d_i is each offset, and n is the total number of examples

$$(5) \quad \bar{x} = \frac{3+3+5+5}{4} = 4.0$$

- ▶ Variance (s^2) measures how far off each offset is from the mean:

$$(6) \quad s^2 = \frac{\sum_i^n (d_i - \bar{d})^2}{n-1} \approx 1.33$$

- ▶ Standard deviation (s) is the square root of variance (s^2) and is about 1.15 in this case

A low deviation means that the mean is a pretty accurate indicator of the distance

Longer distance connections (2)

It is a lot of calculations to look at the offsets for every possible pair of words

- ▶ Restrict the search to be within a window of a set number of words, e.g., 5

The standard deviation gives us useful information—i.e., the words restrict one another in position

- ▶ But if we search for “bigrams at a distance,” then we can use all the other techniques we’ll talk about

For now, though, we’ll focus on words next to each other ...

Determining strength of collocation

We want to compare the likelihood of 2 words next to other being being a chance event vs. being a surprise

- ▶ Do the two words appear next to each other more than we might expect, based on what we know about their individual frequencies?
 - ▶ Is this an accidental pairing or not?
- ▶ We will look at different techniques which define this differently
- ▶ The more data we have, the more confident we will be in our assessment of a collocation or not

We'll look at bigrams, but techniques work for words within five words of each other, translation pairs, phrases, etc.

(Pointwise) Mutual Information

One way to see if two words are strongly connected is to compare

- ▶ the probability of the two words appearing together if they are independent ($p(w_1)p(w_2)$)
- ▶ the actual probability of the two words appearing together ($p(w_1 w_2)$)

The pointwise mutual information is a measure to do this:

$$(7) I(w_1, w_2) = \log \frac{p(w_1 w_2)}{p(w_1)p(w_2)}$$

Pointwise Mutual Information Equation

Our probabilities ($p(w_1 w_2)$, $p(w_1)$, $p(w_2)$) are all basically calculated in the same way:

$$(8) p(x) = \frac{C(x)}{N}$$

- ▶ N is the number of words in the corpus
- ▶ The number of bigrams \approx the number of unigrams

$$(9) I(w_1, w_2) = \log \frac{p(w_1 w_2)}{p(w_1)p(w_2)}$$
$$= \log \frac{\frac{C(w_1 w_2)}{N}}{\frac{C(w_1)}{N} \frac{C(w_2)}{N}}$$
$$= \log \left[N \frac{C(w_1 w_2)}{C(w_1)C(w_2)} \right]$$

Mutual Information example

We want to know if *Ayatollah Ruhollah* is a collocation in a data set we have:

- ▶ $C(\textit{Ayatollah}) = 42$
- ▶ $C(\textit{Ruhollah}) = 20$
- ▶ $C(\textit{AyatollahRuhollah}) = 20$
- ▶ $N = 14,307,668$

$$(10) I(\textit{Ayatollah}, \textit{Ruhollah}) = \log_2 \frac{\frac{20}{N}}{\frac{42}{N} \times \frac{20}{N}} = \log_2 N \frac{20}{42 \times 20} \approx 18.38$$

To see how good a collocation this is, we need to compare it to others

Problems for Mutual Information

The formula we have also has the following equivalencies:

$$(11) I(w_1, w_2) = \log \frac{p(w_1 w_2)}{p(w_1)p(w_2)} = \log \frac{P(w_1|w_2)}{P(w_1)} = \log \frac{P(w_2|w_1)}{P(w_2)}$$

Mutual information tells us how much more information we have for a word, knowing the other word

- ▶ But a decrease in uncertainty isn't quite right ...

A few problems:

- ▶ Sparse data: infrequent bigrams for infrequent words get high scores
- ▶ Tends to measure independence (value of 0) better than dependence
- ▶ Doesn't account for how often the words do **not** appear together (M&S 1999, table 5.15)

Motivating Contingency Tables

What we can instead get at is: which bigrams are likely, out of a range of possibilities?

Looking at the Arthur Conan Doyle story *A Case of Identity*, we find the following possibilities for one particular bigram:

- ▶ *sherlock* followed by *holmes*
- ▶ *sherlock* followed by some word other than *holmes*
- ▶ some word other than *sherlock* preceding *holmes*
- ▶ two words: the first not being *sherlock*, the second not being *holmes*

These are all the relevant situations for examining this bigram

Contingency Tables

We can count up these different possibilities and put them into a contingency table (or 2x2 table)

	B = holmes	B \neq holmes	Total
A = sherlock	7	0	7
A \neq sherlock	39	7059	7098
Total	46	7059	7105

The *Total* row and *Total* column are the **marginals**

- ▶ The values in this chart are the observed frequencies (f_o)

Observed bigram probabilities

Because each cell indicates a bigram, divide each of the cells by the total number of bigrams (7105) to get probabilities:

	holmes	¬ holmes	Total
sherlock	0.00099	0.0	0.00099
¬ sherlock	0.00549	0.99353	0.99901
Total	0.00647	0.99353	1.0

The marginal probabilities indicate the probabilities for a given word, e.g., $p(\textit{sherlock}) = 0.00099$ and $p(\textit{holmes}) = 0.00647$

Expected bigram probabilities

If we assumed that *sherlock* and *holmes* are independent—i.e., the probability of one is unaffected by the probability of the other—we would get the following table:

	holmes	¬ holmes	Total
sherlock	0.00647 x 0.00099	0.99353 x 0.00099	0.00099
¬ sherlock	0.00647 x 0.99901	0.99353 x 0.99901	0.99901
Total	0.00647	0.99353	1.0

- ▶ This is simply $p_e(w_1, w_2) = p(w_1)p(w_2)$

Expected bigram frequencies

Multiplying by 7105 (the total number of bigrams) gives us the expected number of times we should see each bigram:

	holmes	¬ holmes	Total
sherlock	0.05	6.95	7
¬ sherlock	45.5	7052.05	7098
Total	46	7059	7105

- ▶ The values in this chart are the expected frequencies (f_e)

Pearson's chi-square test

The chi-square (χ^2) test measures how far the observed values are from the expected values:

$$(12) \chi^2 = \sum \frac{(f_o - f_e)^2}{f_e}$$

$$\chi^2 = \frac{(7-0.05)^2}{0.05} + \frac{(0-6.95)^2}{6.95} + \frac{(39-45.5)^2}{45.5} + \frac{(7059-7052.05)^2}{7052.05}$$

$$(13) \quad = 966.05 + 6.95 + 1.048 + 0.006$$

$$= 974.05$$

If you look this up in a table, you'll see that it's unlikely to be chance

NB: The χ^2 test does not work well for rare events, i.e., $f_e < 6$

The question is:

- ▶ 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?

Calculating collocations: web interface

For today's practice, let's work with an online concordancer of the BNC, <http://corpus.byu.edu/bnc/>

- ▶ Enter *sweet* in the Search String box.
 - ▶ We can use this search to get our bearings.
 - ▶ Better yet, change the SORT option to be done by RELEVANCE ... This calculates & sorts collocates by MI scores
- ▶ Or, on the left side, check COMPARE WORDS
 - ▶ Enter *sweet* and some other word (e.g., *sour*)
 - ▶ This calculates collocates with each word

Calculating collocations: Perl script

I wrote a Perl script which does the following:

1. Reads in a corpus file (could be changed to read over a directory of files, if need be)
2. Stores unigram and bigram counts as it reads the file in
3. Loops over all bigrams
4. For each bigram, calculates the pointwise mutual information score

You'll note that I simply wrote a function (pmi) which calculate pointwise mutual information

- ▶ This could be replaced by any function that someone (else) writes to calculate a score
- ▶ Or, you could output the data without a score and input that into a statistical package
 - ▶ e.g., we could change the output into a comma-separated value (CSV) file, readable by excel and other software

Testing our intuitions

Let's play around with collocations:

- ▶ Trying different corpora on jones
- ▶ Hypothesizing better/worse collocations
- ▶ Trying to implement a POS filter in our Perl code
- ▶ ... or any other ideas we get ...