Collocations

References

- Lakoff: Women, Fire and Dangerous Things: What Categories Reveal about the Mind (1990)
- Lakoff & Johnson: Metaphors We Live By (1980)
- Manning and Schutze: Foundations of Statistical Natural Language Processing (1999)
- Seretan: Induction of Syntactic Collocation Patterns from Generic Syntactic Relations (2005)
- Stevenson, et. al.: Statistical Measures of the Semi-Productivity of Light Verb Constructions (2004)

Agenda

- Introduction
- Finding Candidates
- Hypothesis Testing
- Applications

Collocations: Definitions

- Broad definition:

  Words strongly associated with each other
  - Example: hospital, insurance
- Narrow definition:

  Two or more (consecutive) words functioning as a (syntactic or semantic) unit
  - Example: dressed to the nines
  - Test: translate word-for-word into another language?
- Book (and I) will use a narrow-ish definition

Collocations: Features

- Book's definition:
  - Non-Compositionality
  - Non-Substitutability
  - Non-Modifiability
- Perhaps "limited" would be better than "non" in the above?

Definition: Non-Compositionality

- The meaning of the collocation cannot easily be derived from the meanings of the individual words
- Example: in broad daylight
  - Literal meaning: during the day
  - Subtle meaning: with no attempt to hide one's actions
- The whole unit must be considered in NLP tasks, such as Machine Translation

Definition: Non-Substitutability

- Cannot substitute an equivalent word for any in the collocation
- Example: in broad daylight
  - Can never be

    * wide daylight

Definition: Non-Modifiability

- The individual words in the collocation resist modification
- Example: in broad daylight
  - Never:

    * broad daylights

Example Classes

- Names
- Technical Terms
- “Light” Verb Constructions
- Phrasal verbs
- Noun Phrases
- Bromides, Stock Phrases, Idioms

Names

- Example: City University of New York
  - Different from: [a] university in the city of New York
- Names are candidates for transliteration
  - They are generally not subject to "normal" machine translation techniques
- Important NLP tasks:
  - Find all names
  - Find all aliases for names
    - Example: CUNY = City University of New York

Technical Terms

- Example: head gasket
  - Part of a car's engine
- Important NLP tasks:
  - Treat each occurrence of a technical term consistently
  - Translate each occurrence in only one fashion

Light Verb Constructions

- Example: to take a walk
- Features:
  - "Light" verb: do, give, have, make, take
  - Verb adds little semantic content to phrase
- Important NLP tasks:
  - Identify light verb constructions
  - Generation of light verbs correctly

    * She makes a walk = elle fait une promenade

Phrasal Verbs

- Example: we make up new stories
  - Make: light verb
  - Make up: invent
- Features: long distance dependencies
  - Example: we made that story up
- Important NLP tasks:
  - Identify phrasal verbs (despite distance!)
  - Translate phrasal verbs as a unit:

    * hicimos una historia para arriba = we made a story up

Noun Phrases

- Example: weapons of mass destruction
  - Also: weapons of mass deception
- Features:
  - Long NPs often become TLAs (esp. in SAE)
  - Possession follows a rule but is not obvious
    - Example: the weapon of mass destruction's payload
- Important NLP tasks:
  - Identify noun phrases as collocations

Bromides, Stock Phrases, Idioms

- Example: as blind as a bat
  - French, German: as myopic as a mole
- Features:
  - All of the canonical features (non-compositionality, etc.)
- Important NLP tasks:
  - Identify idioms
  - Translate them correctly

    The spirit is willing but the flesh is weak
    - Some idioms may be translated directly: my blood is boiling

Agenda

- Introduction
- Finding Candidates
- Hypothesis Testing
- Applications

Finding Candidates

- Basic premise:
  - If two (or more) words co-occur a lot, the co-occurrence is a collocation candidate
  - Candidate may (must?) still be hypothesis-tested
- Tools:
  - Frequency
  - Frequency with Tag Filters
  - Mean and Variance

Frequency

- Technique: unsmoothed bigram
  - Count the number of times a bigram co-occurs
  - Extract top counts and report them as candidates
- Results:
  - Corpus: New York Times
    - August – November, 1990
  - Extremely un-interesting

$C(w^1\ w^2)$	w^1	w^2
80871	of	the
58841	in	the
26430	to	the
21842	on	the
21839	for	the
18568	and	the
16121	that	the
15630	at	the
15494	to	be
13899	in	a
13689	of	a
13361	by	the
13183	with	the
12622	from	the
11428	New	York
10007	he	said
9775	as	a
9231	is	a
8753	has	been
8573	for	a

Frequency with Tag Filters Technique

- Technique: unsmoothed N-gram
  - Count the number of times a bigram co-occurs
  - Tag candidates for POS
  - Pass all candidates through POS filter, considering only ones matching filter
  - Extract top counts and report them as candidates

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*

Frequency with Tag Filters Results

$C(w^1\ w^2)$	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
2106	President	Bush	N N
2001	Middle	East	A N
1942	Saddam	Hussein	N N
1867	Soviet	Union	A N
1850	White	House	A N
1633	United	Nations	A N
1337	York	City	N N
1328	oil	prices	N N
1210	next	year	A N
1074	chief	executive	A N
1073	real	estate	A N

Mean and Variance Technique

- Technique: unsmoothed bigram
  - Produce all possible pairs in a window
  - Consider all pairs in window as candidates
  - Keep data about distance of one word from another
  - Count the number of time each candidate occurs
- Measures:
  - Mean: average offset (possibly negative)
    - Whether two words are related to each other
  - Variance: **?** (offset)
    - Variability in position of two words

Mean and Variance Illustration

- Candidate Generation example:
  - Window: 3

Sentence: *Stocks crash as rescue plan teeters*

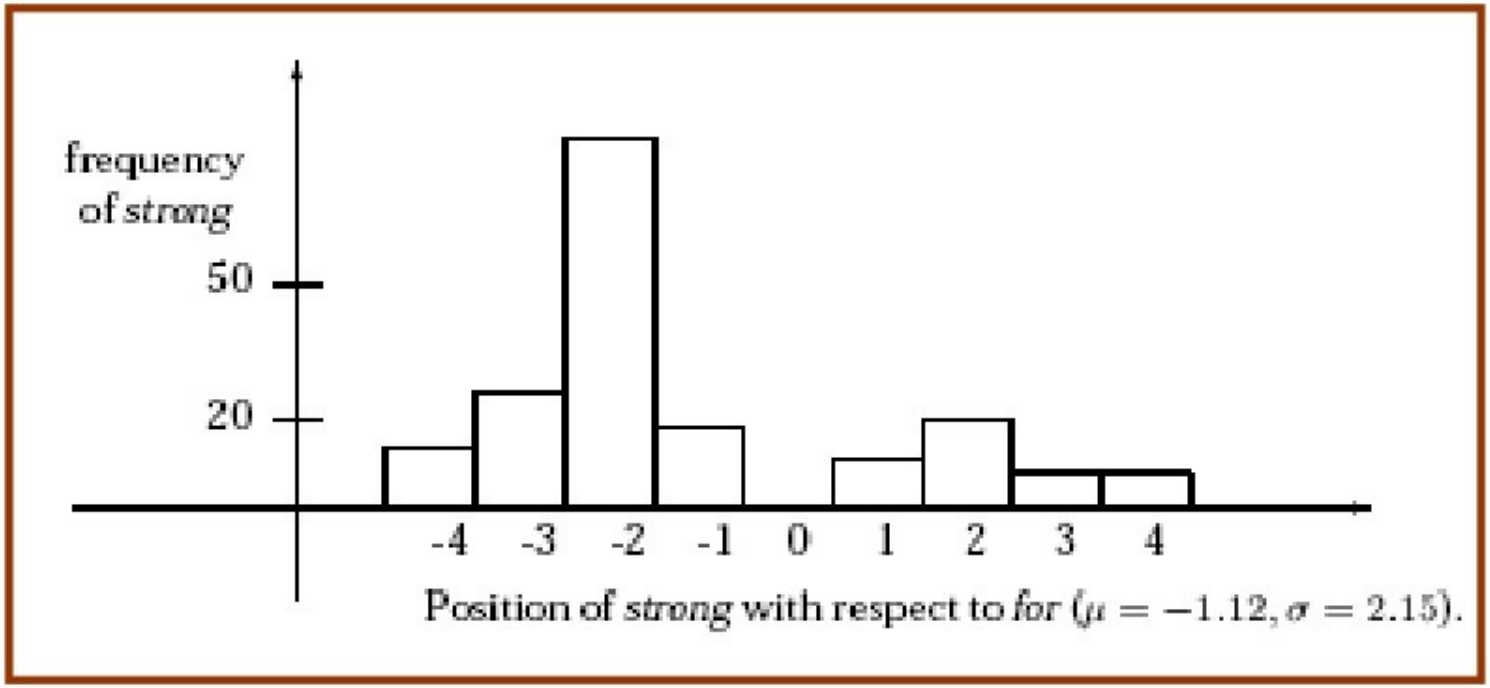
Bigrams:

stocks crash	*stocks as*	*stocks rescue*		
	crash as	*crash rescue*	*crash plan*	
		as rescue	*as plan*	*as teeters*
			rescue plan	*rescue teeters*
				plan teeters

- Used to find collocations with long-distance relationships

Mean and Variance Graphed Results

- Derived data can be placed in a histogram
  - Example: strong vs. for ("strong [business] support for"?)



Mean and Variance Collocations

σ	μ	Count	Word 1	Word 2
0.43	0.97	11657	New	York
0.48	1.83	24	previous	games
0.15	2.98	46	minus	points
0.49	3.87	131	hundreds	dollars
4.03	0.44	36	editorial	Atlanta
4.03	0.00	78	ring	New
3.96	0.19	119	point	hundredth
3.96	0.29	106	subscribers	by
1.07	1.45	80	strong	support
1.13	2.57	7	powerful	organizations
1.01	2.00	112	Richard	Nixon
1.05	0.00	10	Garrison	said

Table 5.5 Finding collocations based on mean and variance. Standard Deviation σ and mean μ of the distances between 12 word pairs.

Agenda

- Introduction
- Finding Candidates
- Hypothesis Testing
- Applications

Hypothesis Testing

- Basic premise:
  - Two (or more) words co-occur a lot
  - Is a candidate a true collocation, or a (not-at-all-interesting) phantom?
- Distribution tools:
  - Mutual Information / Pointwise Mutual Information (I)
  - t Test
  - χ^2 Test
  - Likelihood Ratios (λ)

[Pointwise] Mutual Information (I)

- Intuition:
  - Given a collocation (w_1, w_2) and an observation of w_1
  - $I(w_1; w_2)$ indicates how more likely it is to see w_2
  - The same measure also works in reverse (observe w_2)
- Assumptions:
  - Data is not sparse

Mutual Information Formula

$$I(x', y') = \log_2 \frac{P(x'y')}{P(x')P(y')}$$
$$= \log_2 \frac{P(x'|y')}{P(x')}$$
$$= \log_2 \frac{P(y'|x')}{P(y')}$$

- Measures:
  - $P(w_1)$ = unigram prob.
  - $P(w_1w_2)$ = bigram prob.
  - $P(w_2|w_1)$ = probability of w_2 given we see w_1
- Result:
  - Number indicating increased confidence that we will see w_2 after w_1
  - Number of bits in increased probability

Mutual Information Criticism

- A better measure of the independence of two words rather than the dependence of one word on another
- Horrible on [read: misidentifies] sparse data

The t test Intuition

- Intuition:
  - Compute chance occurrence and ensure observed is *significantly* higher
  - Take several permutations of the words in the corpus
  - How more frequent is the set of all possible permutations than what is observed?
- Assumptions:
  - H_0 is the null hypothesis (chance occurrence)
    - $P(w_1, w_2) = P(w_1)\,P(w_2)$
  - Distribution is "normal"

The t test Formula

$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}}$$

- Measures:
  - x = bigram count
  - ! = $H_0 = P(w_1)\,P(w_2)$
  - s^2 = bigram count (since p ~ p[1 – p])
  - N = total number of bigrams
- Result:
  - Number to look up in a table
  - Degree of confidence that collocation is not created by chance
    - ✔ = the confidence (%) with which one can reject H_0

The t test Sample Findings

t	$C(w^1)$	$C(w^2)$	$C(w^1\ w^2)$	w^1	w^2
4.4721	42	20	20	Ayatollah	Ruhollah
4.4721	41	27	20	Bette	Midler
4.4720	30	117	20	Agatha	Christie
4.4720	77	59	20	videocassette	recorder
4.4720	24	320	20	unsalted	butter
2.3714	14907	9017	20	first	made
2.2446	13484	10570	20	over	many
1.3685	14734	13478	20	into	them
1.2176	14093	14776	20	like	people
0.8036	15019	15629	20	time	last

Table 5.6 Finding collocations: The t test applied to 10 bigrams that occur with frequency 20.

The t test Criticism

- Words are not normally distributed
  - Can reject valid collocation
- Not good on sparse data
- Not good on data with large probabilities

χ^2 Intuition

- Pearson's chi-square test
- Intuition
  - Compare observed frequencies to expected frequencies for independence
  - Large difference = can reject H_0
- Assumptions
  - If sample is not small, the distribution is approximately χ^2

$\square^2$ General Formula

- Measures:
  - E_{ij} = Expected count of the bigram
  - O_{ij} = Observed count of the bigram
- Result
  - A number to look up in a table (like the t test)
  - Degree of confidence (✔) with which H_0

$$X^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

χ^2 Bigram Method and Formula

- Technique for Bigrams:
  - Arrange the bigrams in a 2x2 table with counts for each
  - Formula
    - O_{ij}: i = column; j = row

	$w_1 = new$	$w_1 \neq new$
$w_2 = companies$	8 (*new companies*)	4667 (e.g., *old companies*)
$w_2 \neq companies$	15820 (e.g., *new machines*)	14287181 (e.g., *old machines*)

$$\chi^2 = \frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11} + O_{12})(O_{11} + O_{21})(O_{12} + O_{22})(O_{21} + O_{22})}$$

$$\frac{14307668(8 \times 14287181 - 4667 \times 15820)^2}{(8 + 4667)(8 + 15820)(4667 + 14287181)(15820 + 14287181)} \approx 1.55$$

$\square^2$ Sample Findings

- Comparing corpora
  - Machine Translation
    - Comparison of (English) "cow" and (French) "vache" gives a
    - $\square^2 = 456400$
  - Similarity of two corpora

	cow	¬ *cow*
vache	59	6
¬ *vache*	8	570934

$\square^2$ Criticism

- Not good for small datasets

Intuition

- Likelihood Ratio
- Intuition
  - How much more likely is a given collocation than occurrence by chance?
- Assumptions
  - Formulate two hypotheses:
    - H_1 = words in collocation are independent
    - H_2 = words are dependent

Formula

- Measures
  - $L(H_x)$ = estimate of the distribution of hypothesis H_x

$$
\begin{aligned}
\log \lambda &= \log \frac{L(H_1)}{L(H_2)} \\
&= \log \frac{b(c_{12}, c_1, p) b(c_2 - c_{12}, N - c_1, p)}{b(c_{12}, c_1, p_1) b(c_2 - c_{12}, N - c_1, p_2)} \\
&= \log L(c_{12}, c_1, p) + \log L(c_2 - c_{12}, N - c_1, p) \\
&\quad - \log L(c_{12}, c_1, p_1) - \log L(c_2 - c_{12}, N - c_1, p_2)
\end{aligned}
$$

✦ Sample Findings

$-2 \log \lambda$	$C(w^1)$	$C(w^2)$	$C(w^1 w^2)$	w^1	w^2
1291.42	12593	932	150	most	powerful
99.31	379	932	10	politically	powerful
82.96	932	934	10	powerful	computers
80.39	932	3424	13	powerful	force
57.27	932	291	6	powerful	symbol
51.66	932	40	4	powerful	lobbies
51.52	171	932	5	economically	powerful
51.05	932	43	4	powerful	magnet
50.83	4458	932	10	less	powerful
50.75	6252	932	11	very	powerful
49.36	932	2064	8	powerful	position
48.78	932	591	6	powerful	machines
47.42	932	2339	8	powerful	computer
43.23	932	16	3	powerful	magnets
43.10	932	396	5	powerful	chip
40.45	932	3694	8	powerful	men
36.36	932	47	3	powerful	486
36.15	932	268	4	powerful	neighbor
35.24	932	5245	8	powerful	political
34.15	932	3	2	powerful	cudgels

Table 5.12 Bigrams of *powerful* with the highest scores according to Dunning's likelihood ratio test.

Agenda

- Introduction
- Finding Candidates
- Hypothesis Testing
- Applications

Applications

- Collocations are useful in:
  - Comparison of Corpora
  - Parsing
  - New Topic Detection
  - Computational Lexicography
  - Natural Language Generation
  - Machine Translation

Comparison of Corpora

- Compare corpora to determine:
  - Document clustering (for information retrieval)
  - Plagiarism
- Comparison techniques:
  - Competing hypotheses:
    - Documents are dependent
    - Documents are independent
  - Compare hypotheses using ✦, etc.

	corpus 1	corpus 2
word 1	60	9
word 2	500	76
word 3	124	20
	...	

Parsing

- When parsing, we may get more accurate data by treating a collocation as a unit (rather than individual words)
  - Using the CMU parser, things can go wrong
  - Example: [hand to hand] is a unit in:

```
(S (NP They)
    (VP engaged
        (PP in hand)
        (PP to
            (NP hand combat))))
```

New Topic Detection

- When new topics are reported, the count of collocations associated with those topics increases
  - When topics become old, the count drops

ratio	1990	1989	w^1	w^2
0.0241	2	68	Karim	Obeid
0.0372	2	44	East	Berliners
0.0372	2	44	Miss	Manners
0.0399	2	41	17	earthquake
0.0409	2	40	HUD	officials
0.0482	2	34	EAST	GERMANS
0.0496	2	33	Muslim	cleric
0.0496	2	33	John	Le
0.0512	2	32	Prague	Spring
0.0529	2	31	Among	individual

Table 5.13 Damerau's frequency ratio test. Ten bigrams that occurred twice in the 1990 New York Times corpus, ranked according to the (inverted) ratio of relative frequencies in 1989 and 1990.

Computational Lexicography

- As new multi-word expressions become part of the language, they can be detected
  - Existing collocations can be acquired
- Can also be used for cultural identification
  - Examples:
    - My friend got an A in his class
    - My friend took an A in his class
    - My friend made an A in his class
    - My friend earned an A in his class

Natural Language Generation

- Problem:
  - Given two (or more) possible productions, which is more feasible?
  - Productions usually involve synonyms or near-synonyms
  - Languages generally favour one production

t	$C(w)$	$C(\textit{strong}\ w)$	$C(\textit{powerful}\ w)$	word
3.1622	933	0	10	computers
2.8284	2337	0	8	computer
2.4494	289	0	6	symbol
2.4494	588	0	6	machines
2.2360	2266	0	5	Germany
2.2360	3745	0	5	nation
2.2360	395	0	5	chip
2.1828	3418	4	13	force
2.0000	1403	0	4	friends
2.0000	267	0	4	neighbor
7.0710	3685	50	0	support
6.3257	3616	58	7	enough
4.6904	986	22	0	safety
4.5825	3741	21	0	sales
4.0249	1093	19	1	opposition
3.9000	802	18	1	showing
3.9000	1641	18	1	sense
3.7416	2501	14	0	defense
3.6055	851	13	0	gains
3.6055	832	13	0	criticism

Table 5.7 Words that occur significantly more often with *powerful* (the first ten words) and *strong* (the last ten words).

Machine Translation

- Collocation-complete problem?
  - Must find all used collocations
  - Must parse collocation as a unit
  - Must translate collocation as a unit
  - In target language production, must select among many plausible alternatives

Thanks!

- Questions?