CS114 Lecture 17

Computational Lexical Semantics

Word Similarity

April 8, 2013

Professor Meteer

Thanks for Jurafsky & Martin & Prof. Pustejovksy for slides

Word Similarity

- Synonymy is a binary relation
  - Two words are either synonymous or not
- We want a looser metric
  - Word similarity or
  - Word distance
- Two words are more similar
  - If they share more features of meaning
- Actually these are really relations between **senses**:
  - Instead of saying "bank is like fund"
  - We say
    - Bank1 is similar to fund3
    - Bank2 is similar to slope5
- We'll compute them over both words and senses

Why word similarity

- Information retrieval
- Question answering
- Machine translation
- Natural language generation
- Language modeling
- Automatic essay grading

Two classes of algorithms

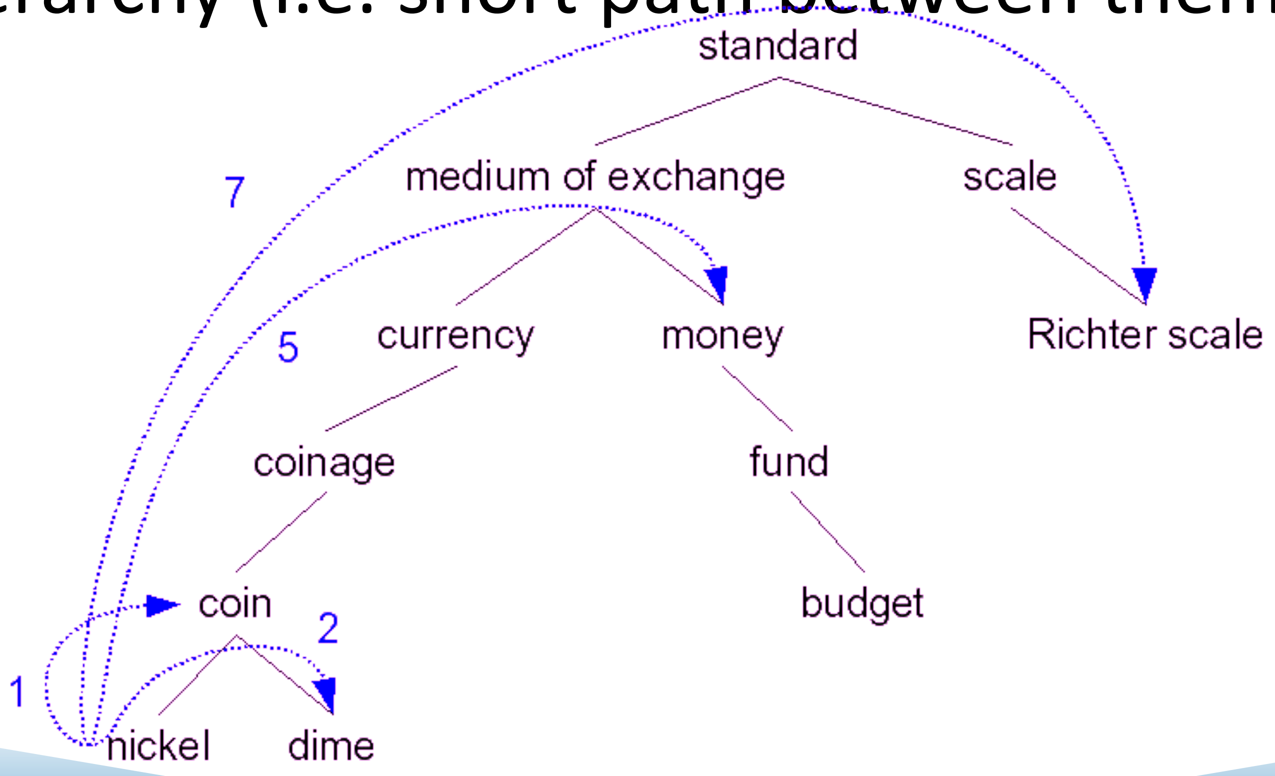
- Thesaurus-based algorithms
  - Based on whether words are "nearby" in Wordnet or MeSH
- Distributional algorithms
  - By comparing words based on their distributional context

Thesaurus-based word similarity

- We could use anything in the thesaurus
  - Meronymy
  - Glosses
  - Example sentences
- In practice
  - By "thesaurus-based" we just mean
    - Using the is-a/subsumption/hypernym hierarchy
- Word similarity versus word relatedness
  - Similar words are near-synonyms
  - Related could be related any way
    - Car, gasoline: related, not similar
    - Car, bicycle: similar

Path based similarity

- Two words are similar if nearby in thesaurus hierarchy (i.e. short path between them)



Refinements to path-based similarity

- pathlen(c1,c2) = number of edges in the shortest path in the thesaurus graph between the sense nodes c1 and c2
- simpath(c1,c2) = -log pathlen(c1,c2)
- wordsim(w1,w2) =
  - $\max_{c1 \in \text{senses}(w1), c2 \in \text{senses}(w2)} \text{sim}(c1,c2)$

Problem with basic path-based similarity

- Assumes each link represents a uniform distance
- *Nickel* to *money* seem closer than *nickel* to *standard*
- Instead:
  - Want a metric which lets us
  - Represent the cost of each edge independently

Information content similarity metrics

- Let’s define P(C) as:
  - The probability that a randomly selected word in a corpus is an instance of concept *c*
  - Formally: there is a distinct random variable, ranging over words, associated with each concept in the hierarchy
  - P(root)=1
  - The lower a node in the hierarchy, the lower its probability

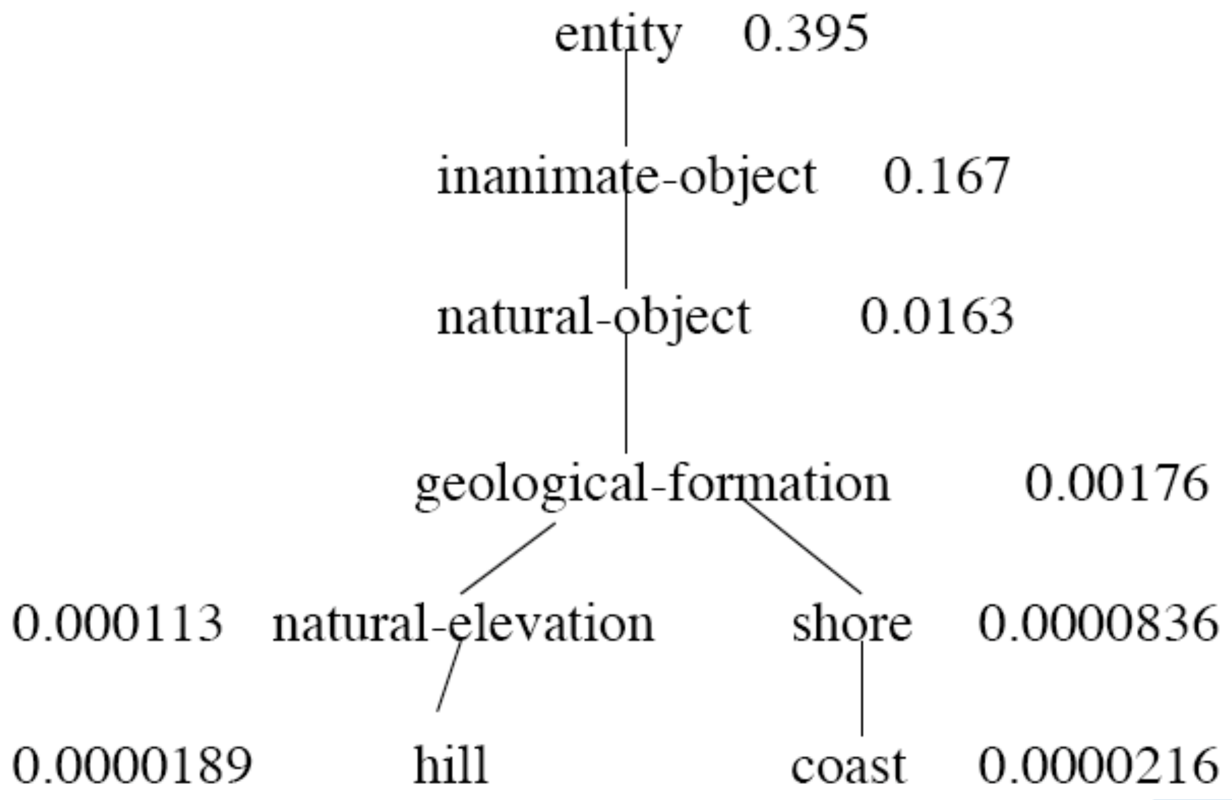
Information content similarity

- Train by counting in a corpus
  - 1 instance of "dime" could count toward frequency of *coin*, *currency*, *standard*, etc
- More formally:

$$P(c) = \frac{\sum_{w \in words(c)} count(w)}{N}$$

Information content similarity

- WordNet hieararchy augmented with probabilities P(C)



Information content: definitions

- Information content:
  - IC(c)=-logP(c)
- Lowest common subsumer
  - LCS(c1,c2) = the lowest common subsumer
    - I.e. the lowest node in the hierarchy
    - That subsumes (is a hypernym of) both c1 and c2
- We are now ready to see how to use information content IC as a similarity metric

Resnik method

- The similarity between two words is related to their common information
- The more two words have in common, the more similar they are
- Resnik: measure the common information as:
  - The info content of the lowest common subsumer of the two nodes
  - $\text{sim}_{\text{resnik}}(c1,c2) = -\log P(\text{LCS}(c1,c2))$

Dekang Lin method

- Similarity between A and B needs to do more than measure common information
- The more differences between A and B, the less similar they are:
  - Commonality: the more info A and B have in common, the more similar they are
  - Difference: the more differences between the info in A and B, the less similar
- Commonality: IC(Common(A,B))
- Difference: IC(description(A,B)-IC(common(A,B))

Dekang Lin method

- Similarity theorem: The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are

$$\text{simLin}(A,B) = \frac{\log P(\text{common}(A,B))}{\log P(\text{description}(A,B))}$$

- Lin furthermore shows (modifying Resnik) that info in common is twice the info content of the LCS

Lin similarity function

- $\text{SimLin}(c1,c2) = \dfrac{2 \times \log P(\text{LCS}(c1,c2))}{\log P(c1) + \log P(c2)}$

- $\text{SimLin}(\text{hill},\text{coast}) = \dfrac{2 \times \log P(\text{geological-formation})}{\log P(\text{hill}) + \log P(\text{coast})}$

- = .59

Extended Lesk

- Two concepts are similar if their glosses contain similar words
  - *Drawing paper*: **paper** that is **specially prepared** for use in drafting
  - *Decal*: the art of transferring designs from **specially prepared paper** to a wood or glass or metal surface
- For each *n*-word phrase that occurs in both glosses
  - Add a score of n^2
  - *Paper* and *specially prepared* for $1 + 4 = 5$…

Summary: thesaurus-based similarity

$$
\begin{aligned}
\text{sim}_{\text{path}}(c_1, c_2) &= -\log \text{pathlen}(c_1, c_2) \\
\text{sim}_{\text{Resnik}}(c_1, c_2) &= -\log P(\text{LCS}(c_1, c_2)) \\
\text{sim}_{\text{Lin}}(c_1, c_2) &= \frac{2 \times \log P(LCS(c_1, c_2))}{\log P(c_1) + \log P(c_2)} \\
\text{sim}_{\text{jc}}(c_1, c_2) &= \frac{1}{2 \times \log P(\text{LCS}(c_1, c_2)) - (\log P(c_1) + \log P(c_2))} \\
\text{sim}_{\text{eLesk}}(c_1, c_2) &= \sum_{r,q \in \text{RELS}} \text{overlap}(\text{gloss}(r(c_1)), \text{gloss}(q(c_2)))
\end{aligned}
$$

Problems with thesaurus-based methods

- We don’t have a thesaurus for every language
- Even if we do, many words are missing
- They rely on hyponym info:
  - Strong for nouns, but lacking for adjectives and even verbs
- Alternative
  - Distributional methods for word similarity

Distributional methods for word similarity

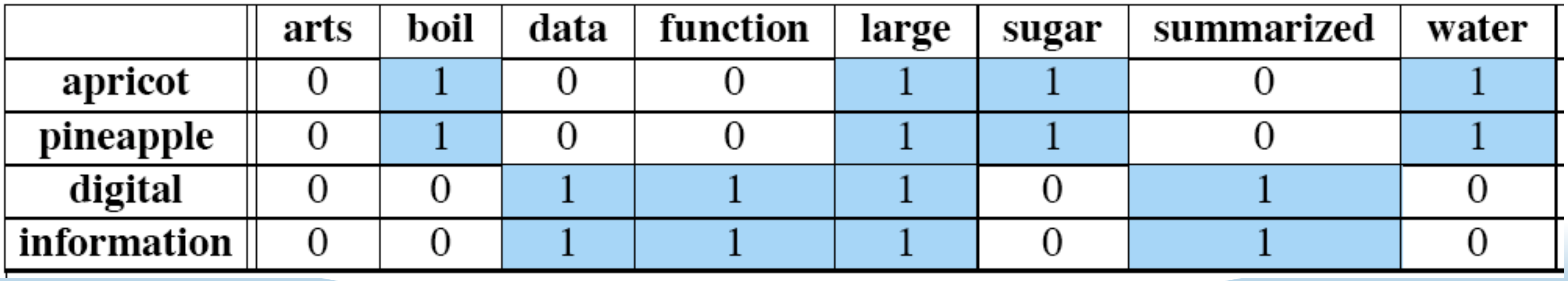
- Firth (1957): “You shall know a word by the company it keeps!”
- Nida example noted by Lin:
  - A bottle of ***tezgüino*** is on the table
  - Everybody likes ***tezgüino***
  - ***Tezgüino*** makes you drunk
  - We make ***tezgüino*** out of corn.
- Intuition:
  - just from these contexts a human could guess meaning of tezguino
  - So we should look at the surrounding contexts, see what other words have similar context.

Context vector

- Consider a target word *w*
- Suppose we had one binary feature f_i for each of the *N* words in the lexicon v_i
- Which means "word v_i occurs in the neighborhood of *w*"
- w=(f1,f2,f3,…,fN)
- If w=tezguino, v1 = bottle, v2 = drunk, v3 = matrix:
- w = (1,1,0,…)

Intuition

- Define two words by these sparse features vectors
- Apply a vector distance metric
- Say that two words are similar if two vectors are similar



	arts	boil	data	function	large	sugar	summarized	water
apricot	0	1	0	0	1	1	0	1
pineapple	0	1	0	0	1	1	0	1
digital	0	0	1	1	1	0	1	0
information	0	0	1	1	1	0	1	0

Distributional similarity

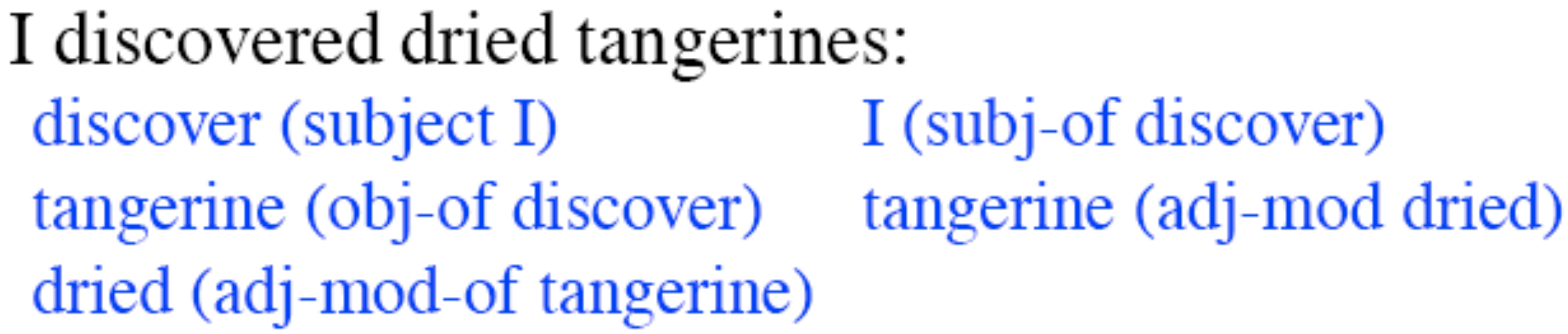
- So we just need to specify 3 things
  1. How the co-occurrence terms are defined
  2. How terms are weighted
     - (frequency? Logs? Mutual information?)
  3. What vector distance metric should we use?
     - Cosine? Euclidean distance?

Defining co-occurrence vectors

- Just as for WSD
- We could have windows
  - Bag-of-words
  - We generally remove **stopwords**
- But the vectors are still very sparse
- So instead of using ALL the words in the neighborhood
- How about just the words occurring in particular relations

Defining co-occurrence vectors

- Zellig Harris (1968)
  - The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entitites relative to other entities

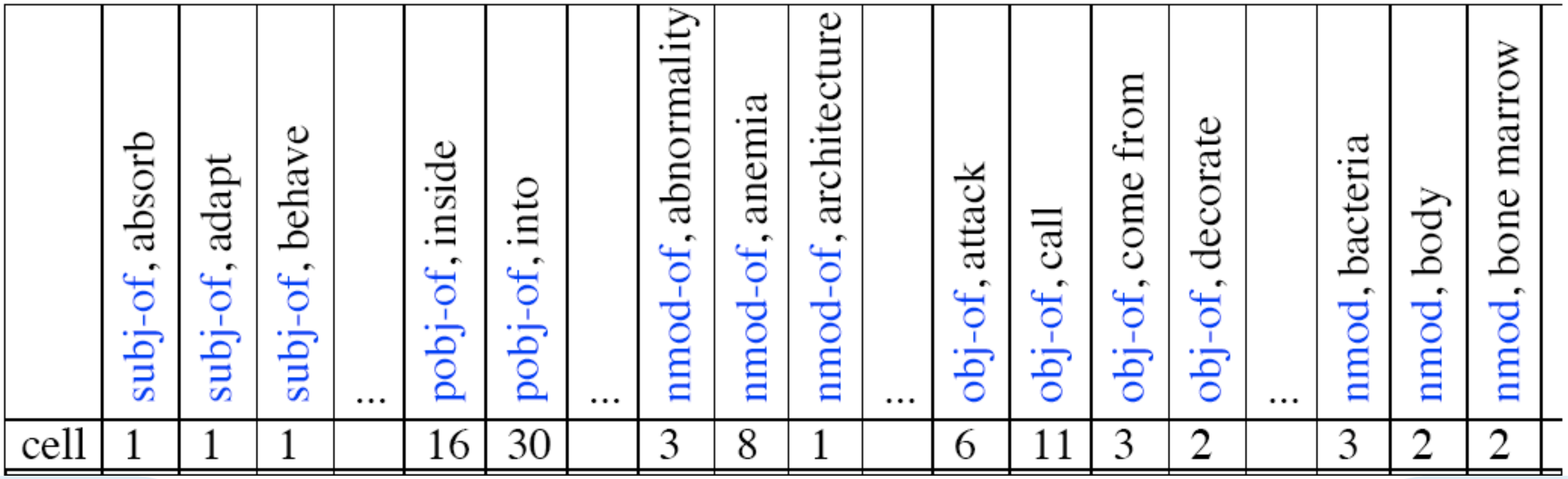


I discovered dried tangerines:

discover (subject I)	I (subj-of discover)
tangerine (obj-of discover)	tangerine (adj-mod dried)
dried (adj-mod-of tangerine)	

Co-occurrence vectors based on dependencies

- For the word “cell”: vector of NxR features
  - R is the number of dependency relations



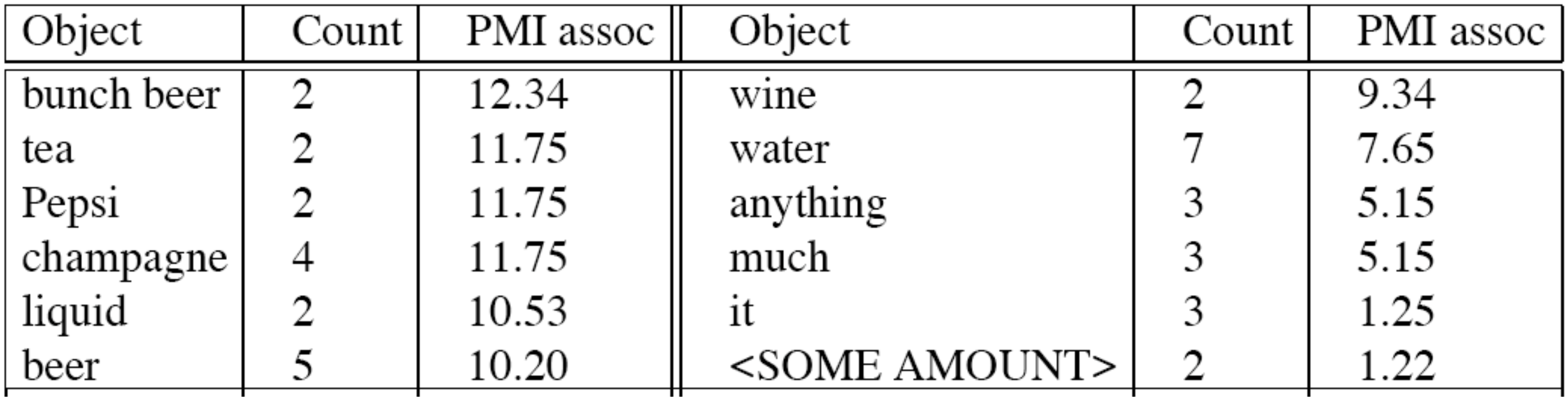
	subj-of, absorb	subj-of, adapt	subj-of, behave	…	pobj-of, inside	pobj-of, into	…	nmod-of, abnormality	nmod-of, anemia	nmod-of, architecture	…	obj-of, attack	obj-of, call	obj-of, come from	obj-of, decorate	…	nmod, bacteria	nmod, body	nmod, bone marrow
cell	1	1	1		16	30		3	8	1		6	11	3	2		3	2	2

2. Weighting the counts
("Measures of association with context")

- We have been using the frequency of some feature as its weight or value
- But we could use any function of this frequency
- Let's consider one feature
- f=(r,w') = (obj-of,*attack*)
- P(f|w)=count(f,w)/count(w)

- $\text{Assoc}_{\text{prob}}(w,f)=p(f|w)$

Intuition: why not frequency



Object	Count	PMI assoc	Object	Count	PMI assoc
bunch beer	2	12.34	wine	2	9.34
tea	2	11.75	water	7	7.65
Pepsi	2	11.75	anything	3	5.15
champagne	4	11.75	much	3	5.15
liquid	2	10.53	it	3	1.25
beer	5	10.20	<SOME AMOUNT>	2	1.22

- "drink it" is more common than "drink wine"
- But "wine" is a better "drinkable" thing than "it"
- Idea:
  - We need to control for change (expected frequency)
  - We do this by normalizing by the expected frequency we would get assuming independence

Weighting: Mutual Information

- **Mutual information:** between 2 random variables X and Y

$$I(X,Y) = \sum_x \sum_y P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- **Pointwise mutual information**: measure of how often two events x and y occur, compared with what we would expect if they were independent:

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

Essential Information Theory

- (Textbook section 4.10)
- Developed by Shannon in the 40s
  - Maximizing the amount of information that can be transmitted over an imperfect communication channel
  - Data compression (entropy)
  - Transmission rate (channel capacity)
- In Computational Linguistics
  - Underlies perplexity: measure of how well a particular grammar matches a particular language

Entropy

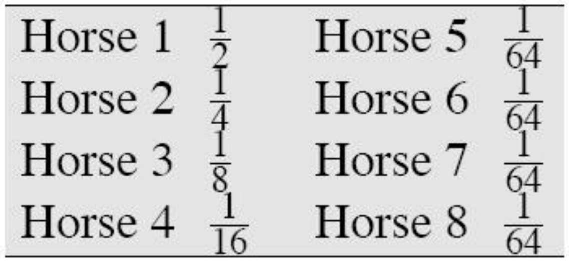
- X: discrete RV, p(X)
- Entropy (or self-information)

$$H(p) = H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

- Entropy measures the amount of information in a RV; it's the average length of the message needed to transmit an outcome of that variable using the optimal code
- Lower bound on the number of bits needed to encode a decision or piece of information

Example

- There are 8 horses in a race. You want to send a bet to your bookie. How many bits do you need?
  - Simplest is 3: 000,001,011, …
  - If we bet all day, the average number of bits is 3
- But assume this distribution of priors on the horse



Horse 1	$\frac{1}{2}$	Horse 5	$\frac{1}{64}$
Horse 2	$\frac{1}{4}$	Horse 6	$\frac{1}{64}$
Horse 3	$\frac{1}{8}$	Horse 7	$\frac{1}{64}$
Horse 4	$\frac{1}{16}$	Horse 8	$\frac{1}{64}$

Example (cont)

- Entropy of the random variable X that ranges over the horses

$$H(p) = H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

= -1/2log1/2 - 1/4log1/4 …

= 2 bits

- For example, we could encode the most likely horse with the code 0, the next with 10, the next with 110, 1110, …
- One bit is the most frequent. If we bet all day, the average would be 2 bits.

Joint Entropy

- The joint entropy of 2 RV X,Y is the amount of the information needed on average to specify both their values

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log p(X,Y)$$

Conditional Entropy

- The conditional entropy of a RV Y given another X, expresses how much extra information one still needs to supply on average to communicate Y given that the other party knows X

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X=x)$$

$$= -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log p(y|x)$$

$$= -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log p(y|x) \quad = -E\left(\log p(Y|X)\right)$$

Chain Rule

$$H(X,Y) = H(X) + H(Y \mid X)$$

$$H(X_{1},\dots,X_{n}) = H(X_{1}) + H(X_{2} \mid X_{1}) + \dots + H(X_{n} \mid X_{1},\dots X_{n-1})$$

Mutual Information

$$H(X,Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y)$$

$$H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) = I(X,Y)$$

- I(X,Y) is the mutual information between X and Y. It is the reduction of uncertainty of one RV due to knowing about the other, or the amount of information one RV contains about the other
- Textbook 20.7.2

Mutual Information (cont)

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

- I is 0 only when X,Y are independent: H(X|Y)=H(X)
- H(X)=H(X)-H(X|X)=I(X,X) Entropy is the self-information

Entropy and Linguistics

- Entropy is measure of uncertainty. The more we know about something the lower the entropy.
- If a language model captures more of the structure of the language, then the entropy should be lower.
- We can use entropy as a measure of the quality of our models

Weighting: Mutual Information

- **Pointwise mutual information**: measure of how often two events x and y occur, compared with what we would expect if they were independent:

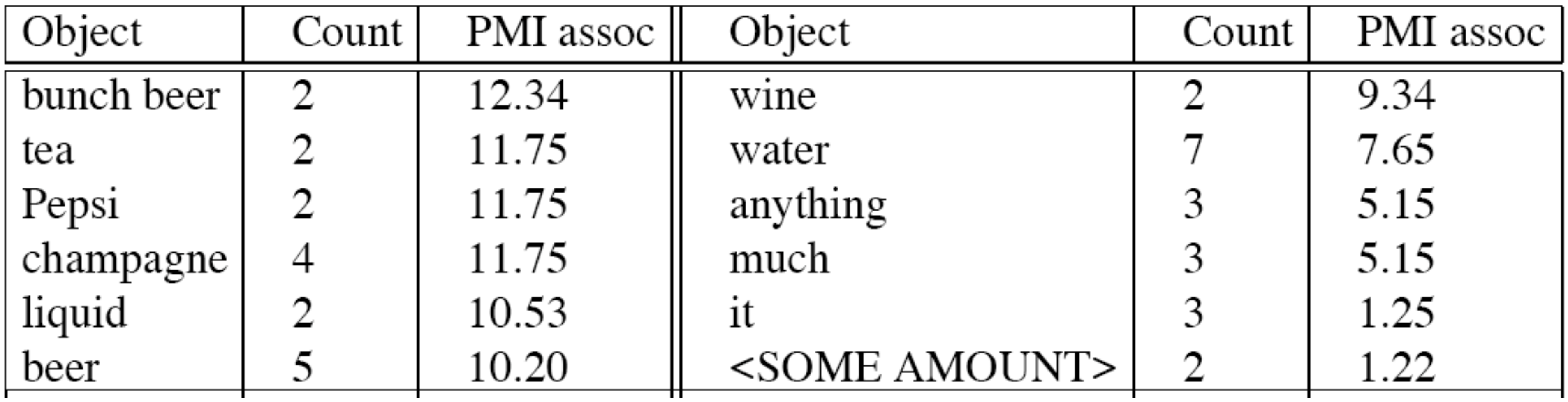
$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- PMI between a target word *w* and a feature *f* :

$$\text{assoc}_{\text{PMI}}(w,f) = \log_2 \frac{P(w,f)}{P(w)P(f)}$$

Mutual information intuition

- Objects of the verb *drink*



Object	Count	PMI assoc	Object	Count	PMI assoc
bunch beer	2	12.34	wine	2	9.34
tea	2	11.75	water	7	7.65
Pepsi	2	11.75	anything	3	5.15
champagne	4	11.75	much	3	5.15
liquid	2	10.53	it	3	1.25
beer	5	10.20	<SOME AMOUNT>	2	1.22

Lin is a variant on PMI

- **Pointwise mutual information**: measure of how often two events x and y occur, compared with what we would expect if they were independent:

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

- PMI between a target word *w* and a feature *f* :

$$\text{assoc}_{\text{PMI}}(w,f) = \log_2 \frac{P(w,f)}{P(w)P(f)}$$

- Lin measure: breaks down expected value for P(f) differently (r:relation, w' related word)

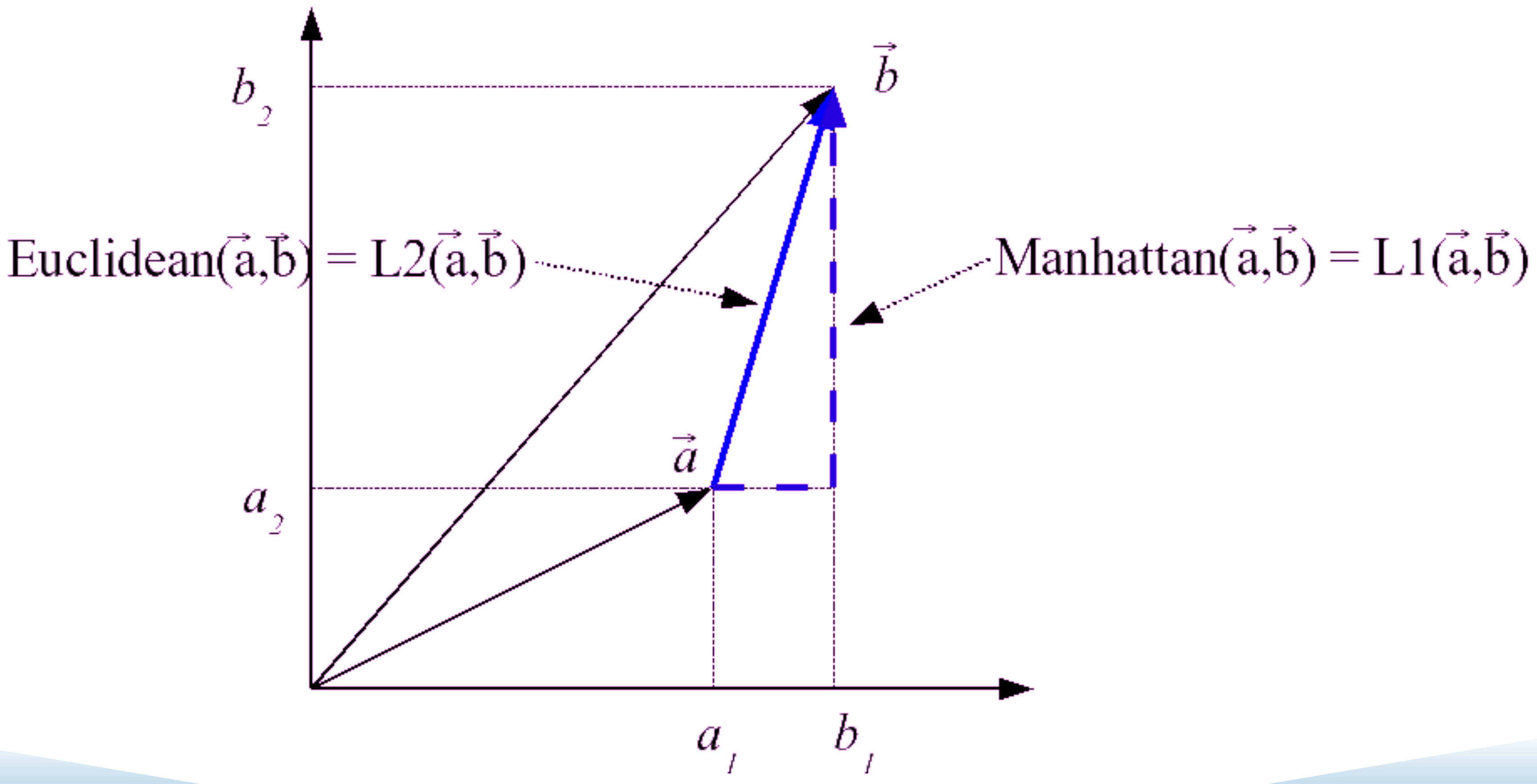
$$\text{assoc}_{\text{Lin}}(w,f) = \log_2 \frac{P(w,f)}{P(w)P(r|w)P(w'|w)}$$

Summary: weightings

- See Manning and Schuetze (1999) for more

$$
\begin{aligned}
\text{assoc}_{\text{prob}}(w,f) &= P(f|w) \\
\text{assoc}_{\text{PMI}}(w,f) &= \log_2 \frac{P(w,f)}{P(w)P(f)} \\
\text{assoc}_{\text{Lin}}(w,f) &= \log_2 \frac{P(w,f)}{P(w)P(r|w)P(w'|w)} \\
\text{assoc}_{\text{t-test}}(w,f) &= \frac{P(w,f)-P(w)P(f)}{\sqrt{P(f)P(w)}}
\end{aligned}
$$

3. Defining similarity between vectors


$\vec{b}$
b_2
$\vec{a}$
a_2
a_1
b_1
Euclidean($\vec{a}$,$\vec{b}$) = L2($\vec{a}$,$\vec{b}$)
Manhattan($\vec{a}$,$\vec{b}$) = L1($\vec{a}$,$\vec{b}$)

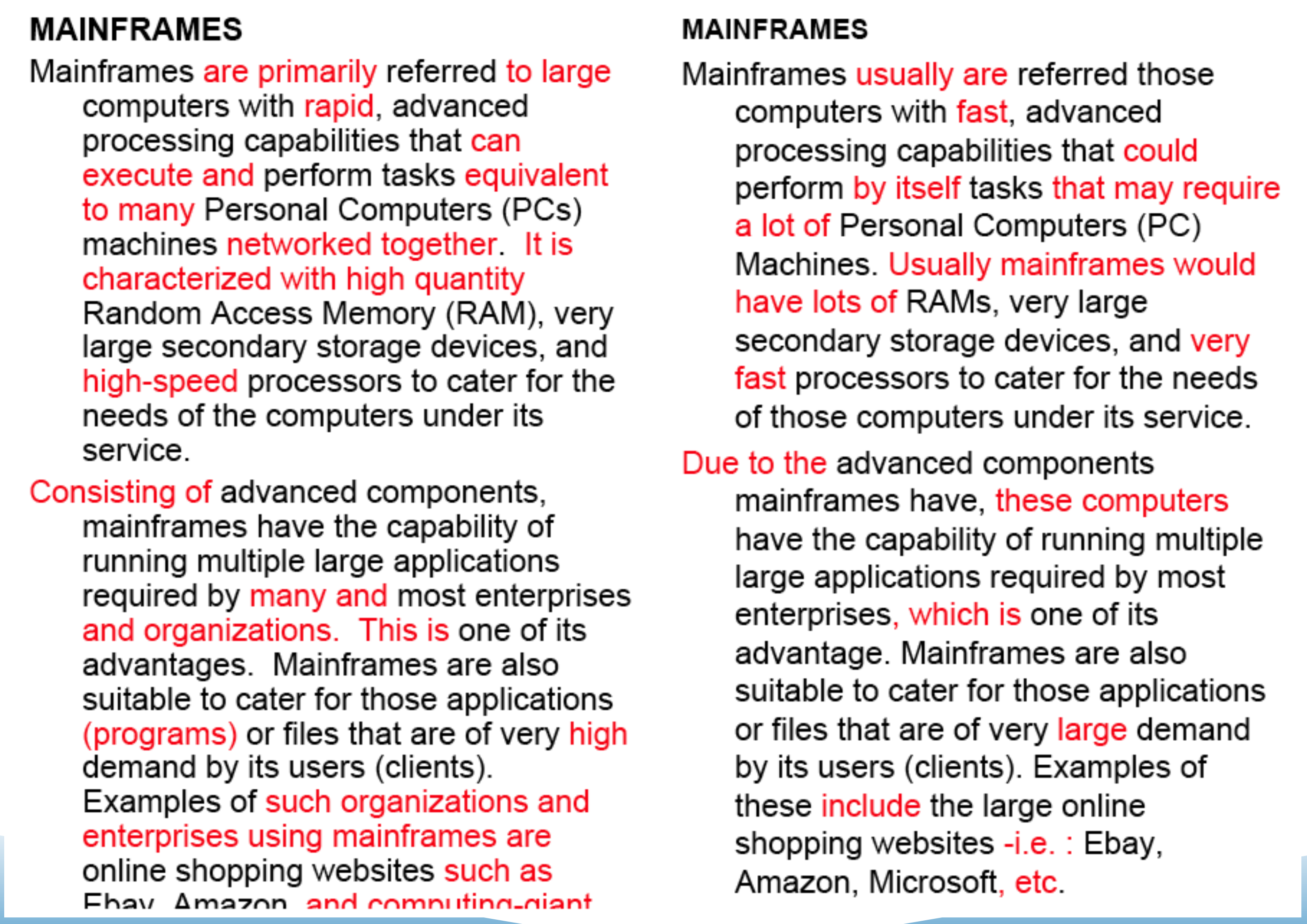
Summary of similarity measures

$$
\begin{aligned}
\text{sim}_{\text{cosine}}(\vec{v},\vec{w}) &= \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}} \\
\text{sim}_{\text{Jaccard}}(\vec{v},\vec{w}) &= \frac{\sum_{i=1}^{N} \min(v_i,w_i)}{\sum_{i=1}^{N} \max(v_i,w_i)} \\
\text{sim}_{\text{Dice}}(\vec{v},\vec{w}) &= \frac{2 \times \sum_{i=1}^{N} \min(v_i,w_i)}{\sum_{i=1}^{N} (v_i + w_i)} \\
\text{sim}_{\text{JS}}(\vec{v}||\vec{w}) &= D(\vec{v}|\tfrac{\vec{v}+\vec{w}}{2}) + D(\vec{w}|\tfrac{\vec{v}+\vec{w}}{2})
\end{aligned}
$$

Evaluating similarity

- Intrinsic Evaluation:
  - Correlation coefficient between algorithm scores
    - And word similarity ratings from humans
- Extrinsic (task-based, end-to-end) Evaluation:
    - Malapropism (spelling error) detection
    - WSD
    - Essay grading
    - Taking TOEFL multiple-choice vocabulary tests
    - Language modeling in some application

An example of detected plagiarism



MAINFRAMES

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high demand by its users (clients). Examples of such organizations and enterprises using mainframes are online shopping websites such as Ebay, Amazon, and computing-giant

MAINFRAMES

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand by its users (clients). Examples of these include the large online shopping websites -i.e. : Ebay, Amazon, Microsoft, etc.

Detecting hyponymy and other relations

- Could we discover new hyponyms, and add them to a taxonomy under the appropriate hypernym?
- Why is this important?
  - "**insulin**" and "**progesterone** are in WN 2.1, but "**leptin**" and "**pregnenolone**" are not.
  - "**combustibility**" and "**navigability**", but not "**affordability**", "**reusability**", or "**extensibility**".
  - "**HTML**" and "**SGML**", but not "**XML**" or "**XHTML**".
  - "**Google**" and "**Yahoo**", but not "**Microsoft**" or "**IBM**".
- This **unknown word problem** occurs throughout NLP

Hearst Approach

- "Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use."
- What does Gelidium mean? How do you know?

$$NP_0 \; such \; as \; NP_1\{, NP_2 \ldots, (and|or) NP_i\}, i \geq 1$$

implies the following semantics

$$\forall NP_i, i \geq 1, \text{hyponym}(NP_i, NP_0)$$

allowing us to infer

$$\text{hyponym}(\text{Gelidium}, \text{red algae})$$

Hearst's hand-built patterns

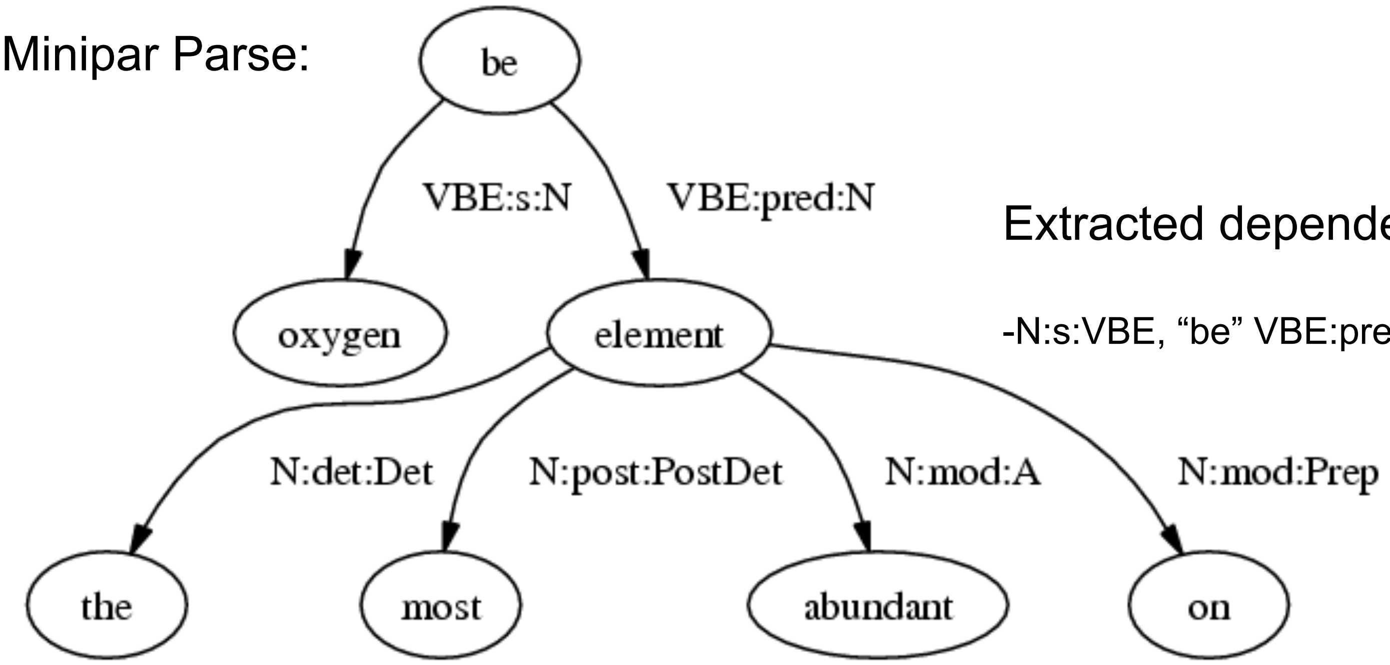
NP {, NP}*{,} (and|or) other NP_H	Temples, treasuries and other important civic buildings
NP_H such as {NP,}* (and|or) NP	Red algae such as Gelidium
such NP_H as {NP,}* (and|or) NP	Works by such authors as Herrick, Goldsmith and Shakespeare
NP_H {,} including {NP,}* (and|or) NP	All common-law countries, including Canada and England
NP_H {,} especially {NP,}* (and|or) NP	…most European countries, especially France, England, and Spain

Recording the Lexico-Syntactic Environment with MINIPAR Syntactic Dependency Paths

MINIPAR: A dependency parser (Lin, 1998)

Example Word Pair: “**oxygen / element**”
Example Sentence: “Oxygen is the most abundant element on the moon.”

Minipar Parse:



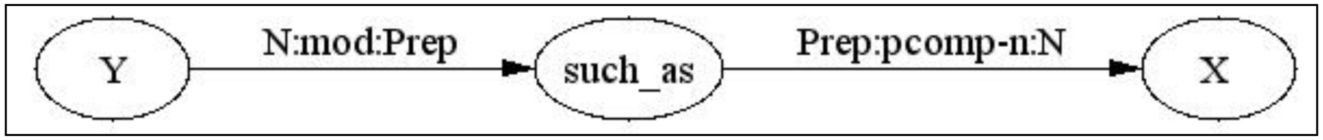
Extracted dependency path:

-N:s:VBE, “be” VBE:pred:N

Each of Hearst's patterns can be captured by a syntactic dependency path in MINIPAR:

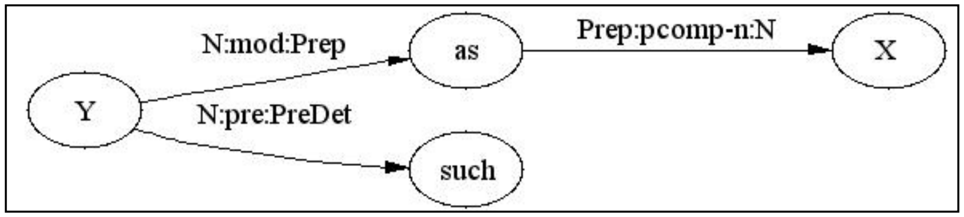
Hearst Pattern **MINIPAR Representation**

Y such as X…



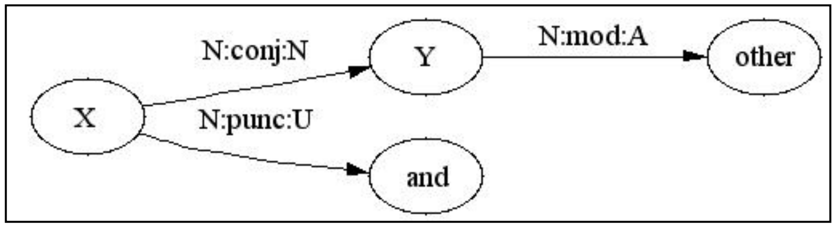
-N:pcomp-n:Prep,such_as,such_as,-Prep:mod:N

Such Y as X…



-N:pcomp-n:Prep,as,as,-Prep:mod:N,(such,PreDet:pre:N)}

X… and other Y



(and,U:punc:N),N:conj:N, (other,A:mod:N)

Algorithm

- Collect noun pairs from corpora
  - (752,311 pairs from 6 million words of newswire)
- Identify each pair as positive or negative example of hypernym-hyponym relationship
  - (14,387 yes, 737,924 no)
- Parse the sentences, extract patterns
  - (69,592 dependency paths occurring in >5 pairs)
- Train a hypernym classifier on these patterns
  - We could interpret each path as a binary classifier
  - Better: **logistic regression** with 69,592 features
    - (actually converted to 974,288 bucketed binary features)

Using Discovered Patterns to Find Novel Hyponym/Hypernym Pairs

Example of a discovered high-precision path:
-N:desc:V,call,call,-V:vrel:N: **“<hypernym> ‘called’ <hyponym>”**

Learned from cases such as:

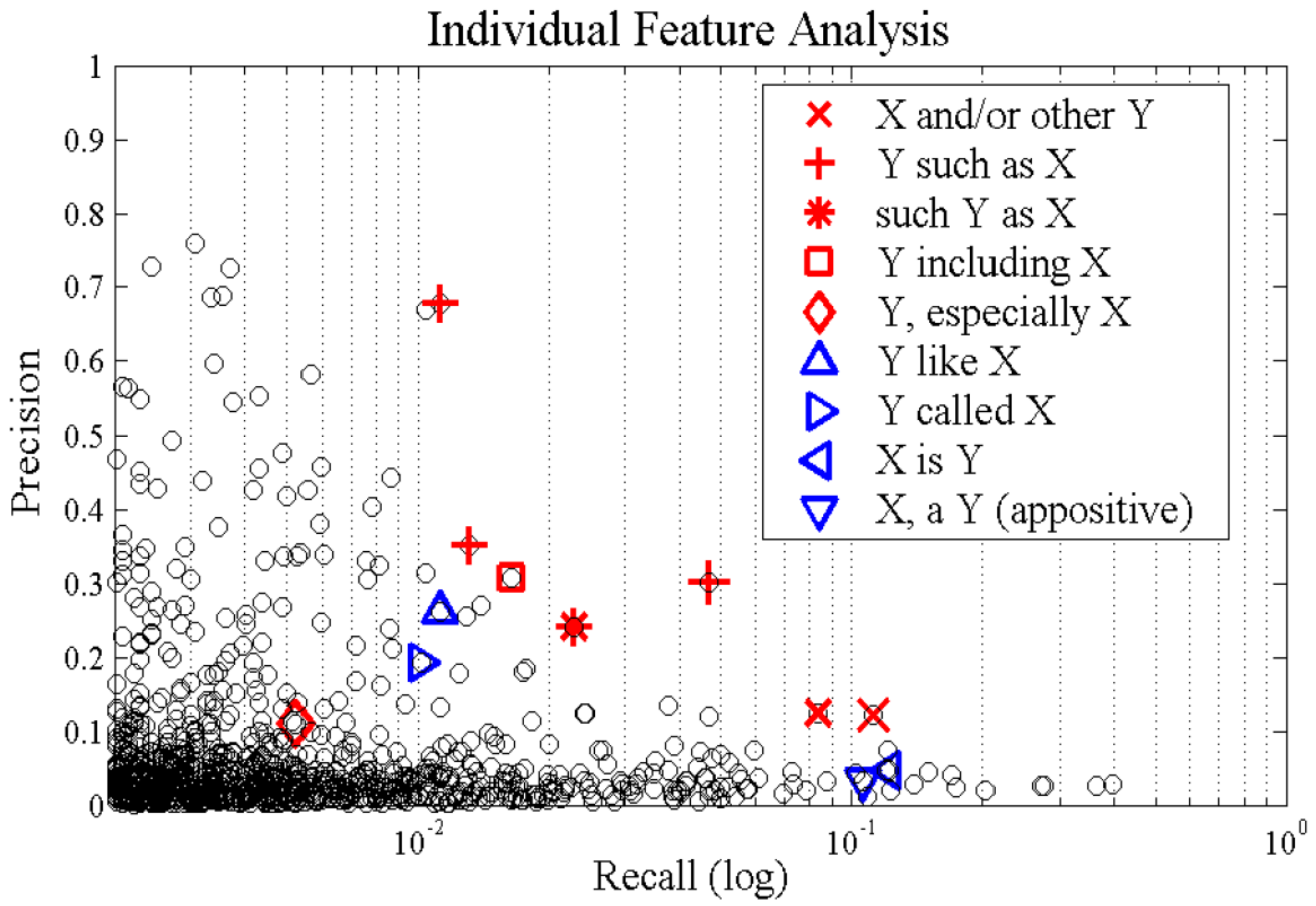
“sarcoma / cancer”: …an uncommon bone **cancer called osteogenic sarcoma** and to…
“deuterium / atom” ….heavy water rich in the doubly heavy hydrogen **atom called deuterium**.

May be used to discover new hypernym pairs not in WordNet:

“*efflorescence / condition*”: …and a **condition called efflorescence** are other reasons for…
“*’neal_inc / company*” …The **company, now called O'Neal Inc.**, was sole distributor of E-Ferol…
“*hat_creek_outfit / ranch*” …run a small **ranch called the Hat Creek Outfit**.
“*tardive_dyskinesia / problem*”: ... irreversible **problem called tardive dyskinesia**…
“*hiv-1 / aids_virus*” …infected by the **AIDS virus, called HIV-1**.
“*bateau_mouche / attraction*” …local sightseeing **attraction called the Bateau Mouche**...
“*kibbutz_malkiyya / collective_farm*” …an Israeli **collective farm called Kibbutz Malkiyya**…

But 70,000 patterns are better than one!

Using each pattern/feature as a binary classifier: Hypernym Precision / Recall



Review

Review

- Major topics for this section
  - Syntax
  - Parsing
  - Semantics
  - Lexical Semantics
- Use the slides to indicate what’s important and the book to describe it in more detail
  - If it’s in the book and not in the slides it won’t be on the test
  - but slides are bullet points and picture—use the book to know how to talk about these points

Syntax

- Know your basic phrase types
  - VP does not mean Vice President
- Terms to know
  - Derivation
  - Overgenerate
  - Syntactic grammars
  - Dependency grammars
  - Verb subcategorization

Syntax

- Key notions that we'll cover
  - Constituency
  - Grammatical relations and Dependency
    - Heads
- Key formalism
  - Context-free grammars
- Resources
  - Treebanks

Parsing Types

- CFGS
  - Top down, bottom up
  - CKY
  - Earley’s algorithm
- Probabalistic CFGs
- Unification Grammars
- Chunking
- Partial parsing

Semantics

- Synonyms vs. Similar vs. Related
- Wordnet
- Word Sense Disambiguation
  - Feature vectors
  - Collocational vs. bag of words
- Similarity metrics
  - Thesaurus-based vs. distributional
  - Context vectors
- Entropy and Mutual Information

Supervised Machine Learning Approaches

- Supervised machine learning approach:
  - a training corpus of words tagged in context with their sense
  - used to train a classifier that can tag words in new text
  - Just as we saw for part-of-speech tagging, statistical MT.
- Summary of what we need:
  - the **tag set** ("sense inventory")
  - the **training corpus**
  - A set of **features** extracted from the training corpus
  - A **classifier**

Thesaurus-based word similarity

- We could use anything in the thesaurus
  - Meronymy
  - Glosses
  - Example sentences
- In practice
  - By “thesaurus-based” we just mean
    - Using the is-a/subsumption/hypernym hierarchy
- Word similarity versus word relatedness
  - Similar words are near-synonyms
  - Related could be related any way
    - Car, gasoline: related, not similar
    - Car, bicycle: similar

Distributional similarity

- So we just need to specify 3 things
  1. How the co-occurrence terms are defined
  2. How terms are weighted
     - (frequency? Logs? Mutual information?)
  3. What vector distance metric should we use?
     - Cosine? Euclidean distance?

Conditional Entropy

- The conditional entropy of a RV Y given another X, expresses how much extra information one still needs to supply on average to communicate Y given that the other party knows X

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X=x)$$

$$= -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log p(y|x)$$

$$= -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log p(y|x) \quad = -E\left(\log p(Y|X)\right)$$

Weighting: Mutual Information

- **Mutual information:** between 2 random variables X and Y

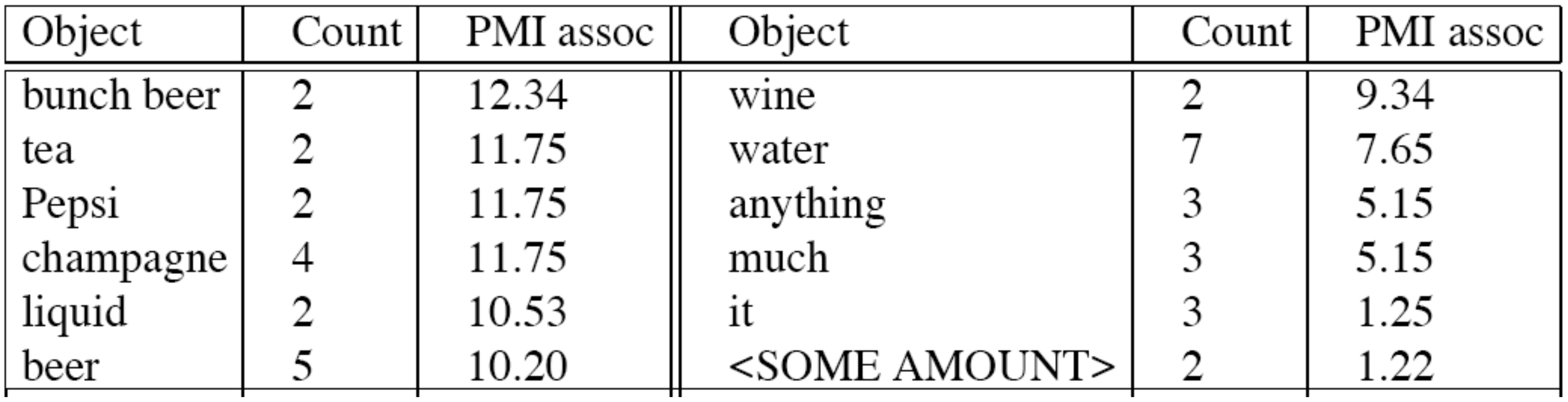
$$I(X,Y) = \sum_x \sum_y P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- **Pointwise mutual information**: measure of how often two events x and y occur, compared with what we would expect if they were independent:

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

Mutual information intuition

- Objects of the verb *drink*



Object	Count	PMI assoc	Object	Count	PMI assoc
bunch beer	2	12.34	wine	2	9.34
tea	2	11.75	water	7	7.65
Pepsi	2	11.75	anything	3	5.15
champagne	4	11.75	much	3	5.15
liquid	2	10.53	it	3	1.25
beer	5	10.20	<SOME AMOUNT>	2	1.22