CS447: Natural Language Processing

http://courses.engr.illinois.edu/cs447

Lecture 13:
Word Sense Disambiguation

Julia Hockenmaier

juliahmr@illinois.edu

3324 Siebel Center

Midterm Exam

When: Tomorrow, Thursday, Oct 12, **6:30pm — 8pm**

Where: DCL 1320

What: Closed book exam:

- You are not allowed to use any cheat sheets, computers, calculators, phones etc.
  (you shouldn't have to anyway)
- Only the material covered in lectures
- Bring a pen (black/blue) or pencil

- **Short questions — we expect short answers!**

Last Wednesday's key concepts

Distributional hypothesis

Distributional similarities:

- word-context matrix
- representing words as vectors
- positive PMI
- computing the similarity of word vectors

Word senses

What does *'bank'* mean?

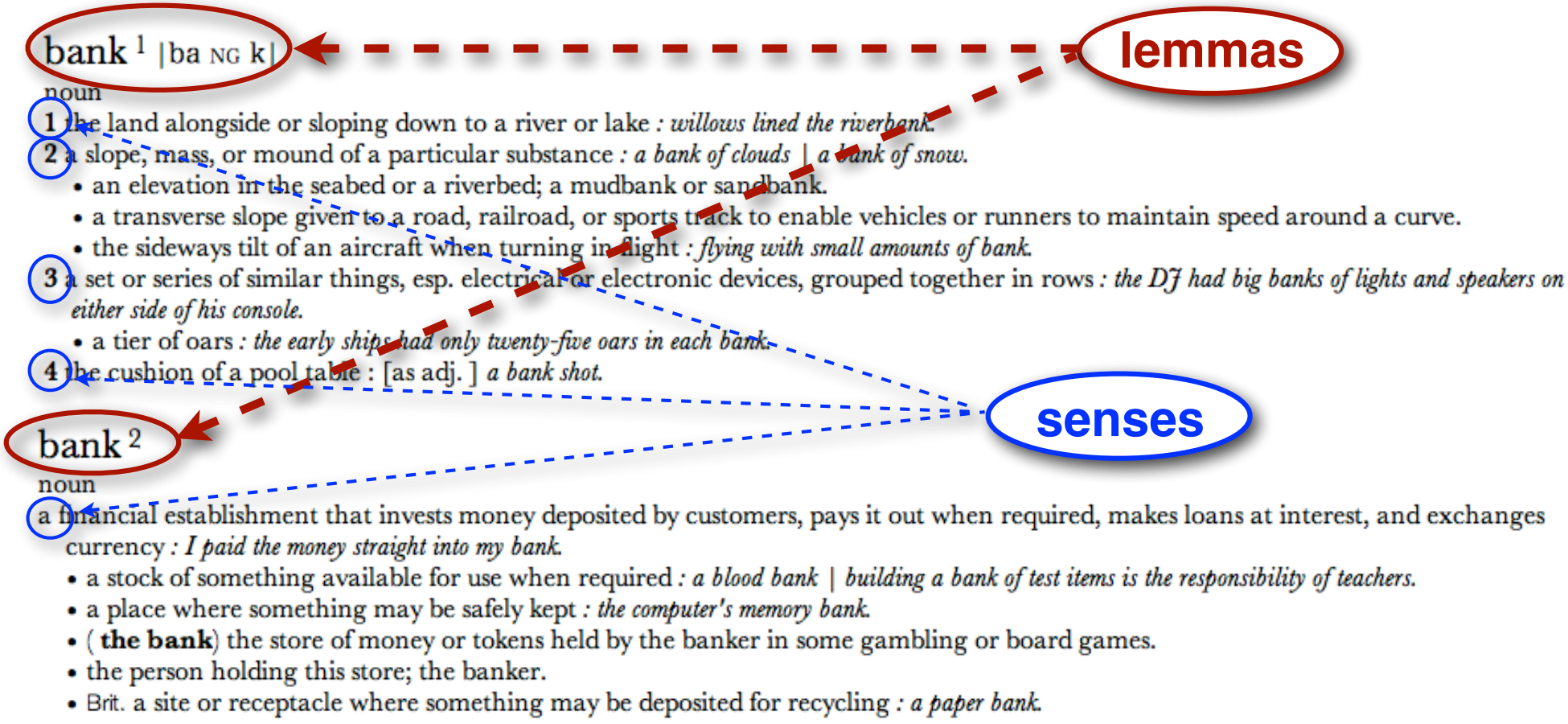
- **a financial institution**
  (US banks have raised interest rates)

- **a particular branch of a financial institution**
  (the bank on Green Street closes at 5pm)

- **the bank of a river**
  (In 1927, the bank of the Mississippi flooded)

- **a 'repository'**
  (I donate blood to a blood bank)

Lexicon entries



Some terminology

Word forms: *runs, ran, running; good, better, best*

Any, possibly inflected, form of a word
(i.e. what we talked about in morphology)

Lemma (citation/dictionary form): *run*

A basic word form (e.g. infinitive or singular nominative noun) that is used to represent all forms of the same word.
(i.e. the form you'd search for in a dictionary)

Lexeme: RUN(V), GOOD(A), BANK[1](N), BANK[2](N)

An abstract representation of a word (and all its forms), with a part-of-speech and a set of related word senses.
(Often just written (or referred to) as the lemma, perhaps in a ***different*** **FONT**)

Lexicon:

A (finite) list of lexemes

Trying to make sense of senses

Polysemy:

A lexeme is polysemous if it has different *related senses*

bank = financial institution or building

Homonyms:

Two lexemes are homonyms if their *senses are unrelated*, but they happen to have the **same spelling and pronunciation**

bank = (financial) bank or (river) bank

Relations between senses

Symmetric relations:

Synonyms: *couch/sofa*

Two lemmas with the **same** sense

Antonyms: *cold/hot, rise/fall, in/out*

Two lemmas with the **opposite** sense

Hierarchical relations:

Hypernyms and **hyponyms**: *pet/dog*

The hyponym *(dog)* is **more specific** than the hypernym *(pet)*

Holonyms and **meronyms:** *car/wheel*

The meronym *(wheel)* is a **part of** the holonym *(car)*

WordNet

WordNet

Very large lexical database of English:
110K nouns, 11K verbs, 22K adjectives, 4.5K adverbs
(WordNets for many other languages exist or are under construction)

Word senses grouped into synonym sets ("synsets")
linked into a conceptual-semantic hierarchy
81K noun synsets, 13K verb synsets, 19K adj. synsets, 3.5K adv synsets
Avg. # of senses: 1.23 nouns, 2.16 verbs, 1.41 adj, 1.24 adverbs

Conceptual-semantic relations: hypernym/hyponym
also holonym/meronym
Also lexical relations, in particular lemmatization

Available at http://wordnet.princeton.edu

A WordNet example

WordNet Search - 3.0 - WordNet home page - Glossary - Help

Word to search for: bass Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Noun

- S: (n) **bass** (the lowest part of the musical range)
- S: (n) **bass**, bass part (the lowest part in polyphonic music)
- S: (n) **bass**, basso (an adult male singer with the lowest voice)
- S: (n) sea bass, **bass** (the lean flesh of a saltwater fish of the family Serranidae)
- S: (n) freshwater bass, **bass** (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
- S: (n) **bass**, bass voice, basso (the lowest adult male singing voice)
- S: (n) **bass** (the member with the lowest range of a family of musical instruments)
- S: (n) **bass** (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

Adjective

- S: (adj) **bass**, deep (having or denoting a low vocal or instrumental range) *"a deep voice"; "a bass voice is lower than a baritone voice"; "a bass clarinet"*

WordNet home page

Hierarchical synset relations: nouns

Hypernym/hyponym (between concepts)
The more general '***meal***' is a hypernym of the more specific '***breakfast***'

Instance hypernym/hyponym (between concepts and instances)
Austen is an instance hyponym of ***author***

Member holonym/meronym (groups and members)
professor is a member meronym of (a university's) ***faculty***

Part holonym/meronym (wholes and parts)
wheel is a part meronym of (is a part of) ***car.***

Substance meronym/holonym (substances and components)
flour is a substance meronym of (is made of) ***bread***

Hierarchical synset relations: verbs

Hypernym/troponym (between events):
travel/fly, walk/stroll
Flying is a troponym of *traveling:*
it denotes **a specific manner** of *traveling*

Entailment (between events):
snore/sleep
Snoring **entails (presupposes)** *sleeping*

WordNet Hypernyms and Hyponyms

- S: (n) **bass** (the lowest part of the musical range)
  - *direct hypernym* / *inherited hypernym* / *sister term*
    - S: (n) pitch (the property of sound that varies with variation in the frequency of vibration)
      - S: (n) sound property (an attribute of sound)
        - S: (n) property (a basic or essential attribute shared by all members of a class) "*a stud*
          - S: (n) attribute (an abstraction belonging to or characteristic of an entity)
            - S: (n) abstraction, abstract entity (a general concept formed by extracting
              - S: (n) entity (that which is perceived or known or inferred to have
- S: (n) **bass**, bass part (the lowest part in polyphonic music)
  - *direct hyponym* / ***full hyponym***
    - S: (n) ground bass (a short melody in the bass that is constantly repeated)
    - S: (n) figured bass, basso continuo, continuo, thorough bass (a bass part written out in full and acc
  - *direct hypernym* / *inherited hypernym* / *sister term*
    - S: (n) part, voice (the melody carried by a particular voice or instrument in polyphonic music) "*he*
      - S: (n) tune, melody, air, strain, melodic line, line, melodic phrase (a succession of notes forr
        - S: (n) music (an artistic form of auditory communication incorporating instrumental c
          - S: (n) auditory communication (communication that relies on hearing)
            - S: (n) communication (something that is communicated by or to or betwe
              - S: (n) abstraction, abstract entity (a general concept formed by extr
                - S: (n) entity (that which is perceived or known or inferred to

Thesaurus-based similarity

Thesaurus-based word similarity

Instead of using distributional methods, rely on a resource like WordNet to compute word similarities.

Problem: each word may have multiple entries in WordNet, depending on how many senses it has.
We often just assume that the similarity of two words is equal to the similarity of their two most similar senses.

NB: There are a few recent attempts to combine neural embeddings with the information encoded in resources like WordNet. Here, we'll just go quickly over some classic approaches.

Thesaurus-based word similarity

Basic idea:

A thesaurus like WordNet contains all the information needed to compute a semantic distance metric.

Simplest instance: compute distance in WordNet

$$sim(s, s') = -log\ pathlen(s, s')$$

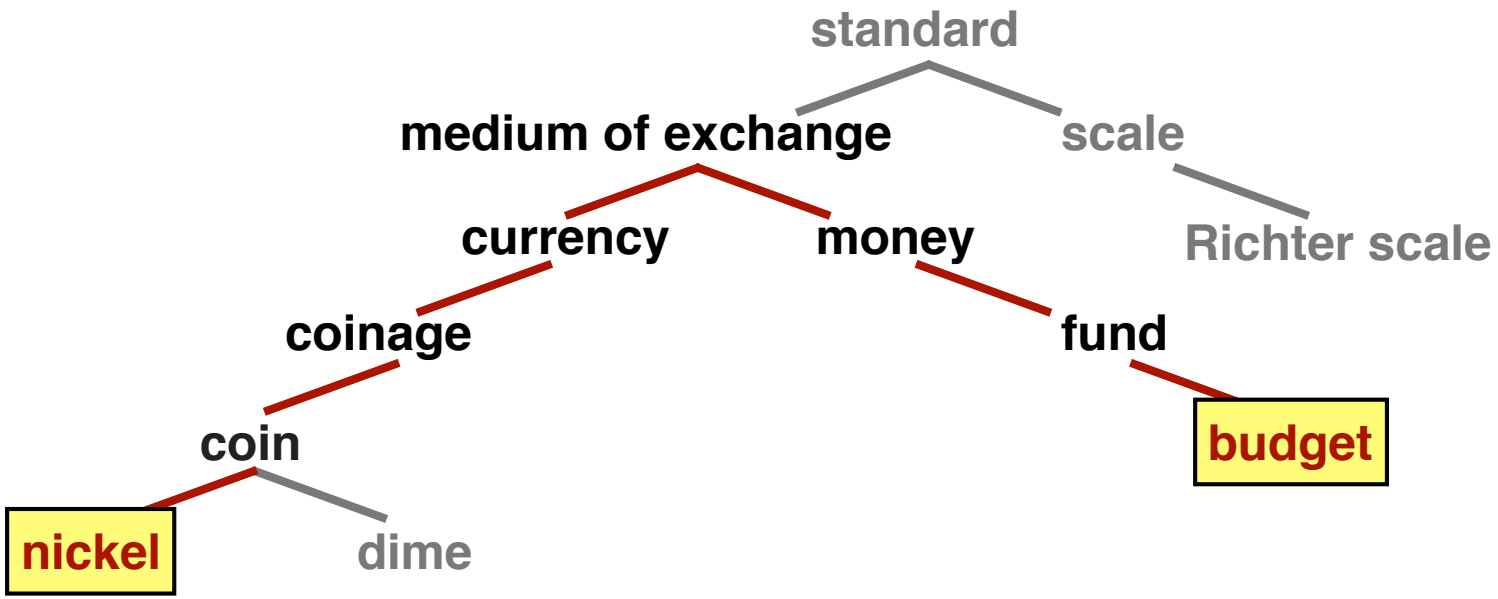
pathlen(s,s'): number of edges in shortest path between *s* and *s'*

Note: WordNet nodes are synsets (=word senses).
Applying this to words w, w':

$$sim(w, w') = \max_{\substack{s \in Senses(w) \\ s' \in Senses(w')}} sim(s, s')$$

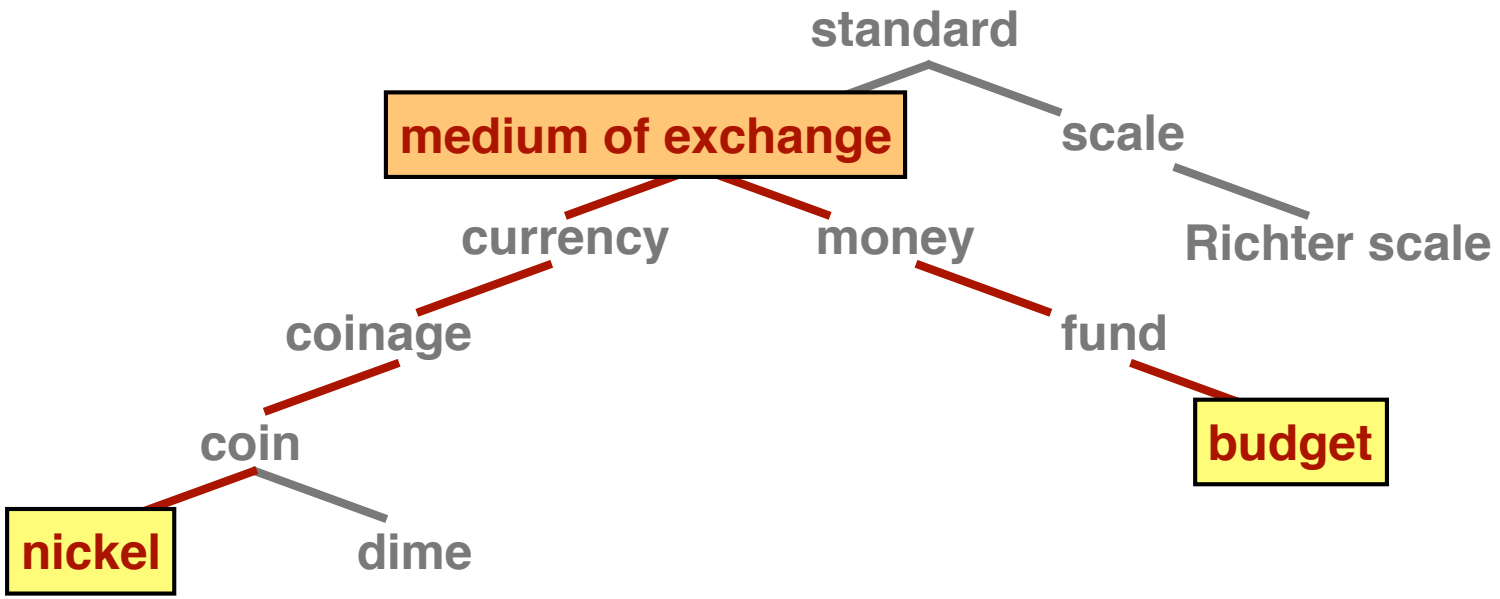
WordNet path lengths

The **path length** (distance) ***pathlen(s, s')***
between two senses s, s' is the length of the (shortest) path between them

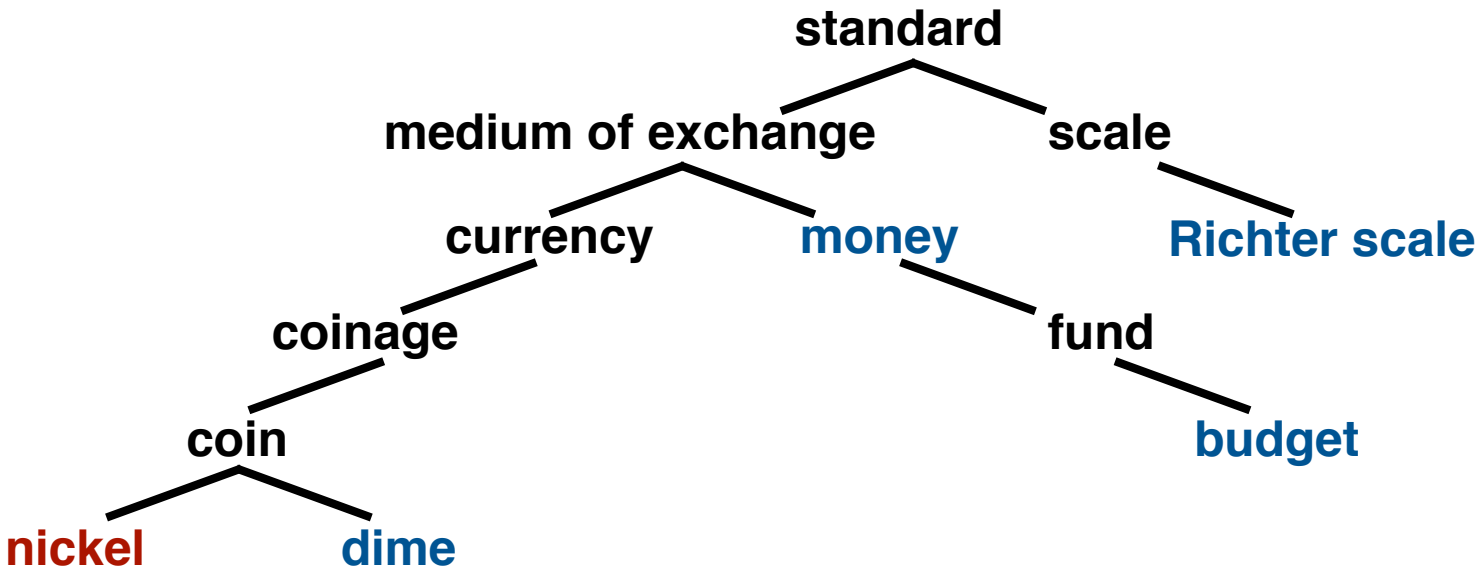


The lowest common subsumer

The **lowest common subsumer** (ancestor) ***LCS(s, s')*** of two senses s, s' is the lowest common ancestor node in the hierarchy



WordNet path lengths



A few examples:

pathlen(nickel, dime) = 2
pathlen(nickel, money) = 5
pathlen(nickel, budget) = 7

But do we really want the following?

pathlen(nickel, coin) < pathlen(nickel, dime)
pathlen(nickel, Richter scale) = pathlen(nickel, budget)

Information-content similarity

Basic idea: Add **corpus statistics** to thesaurus hierarchy

For each concept/sense *s* (synset node in WordNet), define:

- ***words(s):*** the set of words subsumed by (=below) *s.*

All words will be subsumed by the root of the hierarchy

- ***P(s):*** the probability that a random word in the corpus is an instance of *s*

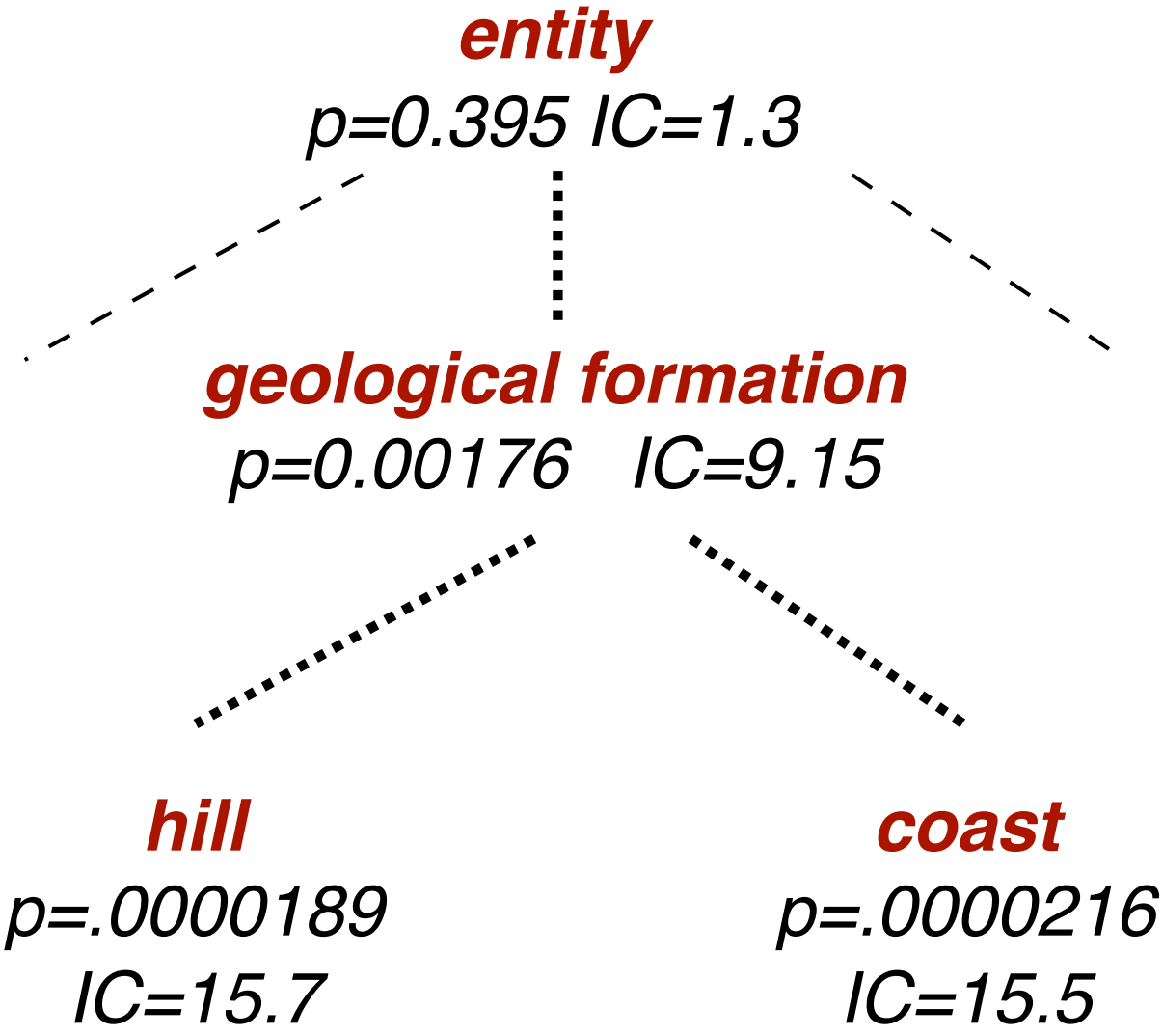
$$P(s) = \frac{\sum_{w \in words(s)} c(w)}{N}$$

(Either use a sense-tagged corpus, or count each word as one instance of each of its possible senses)

- This defines the **Information content** of a sense s:

IC(s) = –log P(s)

P(s) and IC(s): examples



Using LCS to compute similarity

Resnik (1995)'s similarity metric:

$$sim_{Resnik}(s,s') = -log\ P(\ LCS(s,\ s')\ )$$

The underlying intuition:

- If $s_{LCS} = LCS(s,s')$ is the root of the hierarchy, $P(s_{LCS})=1$
- The lower s_{LCS} is in the hierarchy, the more specific it is, and the lower $P(s_{LCS})$ will be.

LCS(car, banana) = physical entity
LCS(nickel, dime) = coin

Problem: this does not take into account how different s,s' are

LCS(thing, object) = physical entity = LCS(car, banana)

Better similarity metrics

Lin (1998)'s similarity:

$$sim_{Lin}(s,s') = 2\times \log P(s_{LCS}) / [\log P(s) + \log P(s')]$$

Jiang & Conrath (1997) 's distance

$$dist_{JC}(s,s') = 2\times \log P(s_{LCS}) - [\log P(s) + \log P(s')]$$

$$sim_{JC}(s,s') = 1/dist_{JC}(s, s')$$

(NB: you don't have to memorize these for the exam…)

Problems with thesaurus-based similarity

We need to have a thesaurus!
(not available for all languages)

We need to have a thesaurus that contains the words we're interested in.

We need a thesaurus that captures a rich hierarchy of hypernyms and hyponyms.

Most thesaurus-based similarities depend on the specifics of the hierarchy that is implement in the thesaurus.

Learning hyponym relations

If we don't have a thesaurus, can we learn that Corolla is a kind of car?

Certain **phrases and patterns** indicate hyponym relations:

Hearst(1992)

Enumerations: *cars* ***such as*** *the Corolla, the Civic, and the Vibe,*

Appositives: *the Corolla* **,** *a popular car…*

We can also **learn these patterns** if we have some **seed examples of hyponym relations** (e.g. from WordNet):

1. *Take all hyponym/hypernym pairs from WordNet (e.g. car/vehicle)*
2. *Find all sentences that contain both, and identify patterns*
3. *Apply these patterns to new data* to get new hyponym/hypernym pairs

Word Sense Disambiguation

What does this word mean?

This **plant** needs to be watered each day.

⇒ **living plant**

This **plant** manufactures 1000 widgets each day.

⇒ **factory**

Word Sense Disambiguation (WSD):

Identify the sense of content words (nouns, verbs, adjectives) in context (assuming a fixed inventory of word senses)

Applications: machine translation, question answering, information retrieval, text classification

The data

Sense	Training Examples (Keyword in Context)
?	... company said the *plant* is still operating
?	Although thousands of *plant* and animal species
?	... zonal distribution of *plant* life . ...
?	... to strain microscopic *plant* life from the ...
?	vinyl chloride monomer *plant* , which is ...
?	and Golgi apparatus of *plant* and animal cells
?	... computer disk drive *plant* located in ...
?	... divide life into *plant* and animal kingdom
?	... close-up studies of *plant* life and natural
?	... Nissan car and truck *plant* in Japan is ...
?	... keep a manufacturing *plant* profitable without
?	... molecules found in *plant* and animal tissue
?	... union responses to *plant* closures . ...
?	... animal rather than *plant* tissues can be
?	... many dangers to *plant* and animal life
?	company manufacturing *plant* is in Orlando ...
?	... growth of aquatic *plant* life in water ...
?	automated manufacturing *plant* in Fremont ,
?	... Animal and *plant* life are delicately
?	discovered at a St. Louis *plant* manufacturing
?	computer manufacturing *plant* and adjacent ...
?	... the proliferation of *plant* and animal life
?	... ...

WSD evaluation

Evaluation metrics:

- **Accuracy:** How many instances of the word are tagged with their correct sense?
- **Precision and recall:** How many instances of each sense did we predict/recover correctly?

Baseline accuracy:

- Choose the **most frequent sense** per word

WordNet: take the first (=most frequent) sense

- **Lesk algorithm** (see below)

Upper bound accuracy:

- **Inter-annotator agreement**: how often do two people agree

~75-80% for all words task with WordNet, ~90% for simple binary tasks

- **Pseudo-word task:** Replace all occurrences of words w_a and w_b (*door*, *banana*) with a nonsense word w_{ab} (*banana-door*).

Dictionary-based WSD: Lesk algorithm

(Lesk 1986)

Dictionary-based methods

We often don't have a labeled corpus, but we might have a **dictionary/thesaurus** that contains **glosses** and **examples**:

bank1
Gloss: a financial institution that accepts deposits and channels the money into lending activities
Examples: *"he cashed the check at the bank",*
"that bank holds the mortgage on my home"

bank2
Gloss: sloping land (especially the slope beside a body of water)
Examples: *"they pulled the canoe up on the bank",*
"he sat on the bank of the river and watched the current"

The Lesk algorithm

Basic idea: Compare the context with the dictionary definition of the sense.

Assign the dictionary sense whose gloss and examples are most similar to the context in which the word occurs.

Compare the **signature of a word** **in context**
with the **signatures of its senses** **in the dictionary**
Assign the sense that is **most similar** to the context

Signature = set of content words
(in examples/gloss or in context)
Similarity = size of intersection of context signature and sense signature

Sense signatures (dictionary)

***bank1*:**

Gloss: a financial institution that accepts deposits and channels the money into lending activities

Examples: *"he cashed the check at the bank", "that bank holds the mortgage on my home"*

Signature(bank1) = {*financial, institution, accept, deposit, channel, money, lend, activity, cash, check, hold, mortgage, home*}

***bank2*:**

Gloss: sloping land (especially the slope beside a body of water)

Examples: *"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the current"*

Signature(bank2) = {*slope, land, body, water, pull, canoe, sit, river, watch, current*}

Signature of target word

Test sentence:
*"The **bank** refused to give me a loan."*

Simplified Lesk: Overlap between sense signature and (simple) signature of the target word:

Target signature = words in context: **{refuse, give, loan}**

Original Lesk: Overlap between sense signature and augmented signature of the target word

Augmented target signature with signatures of words in context
{refuse, reject, request,... , **give**, gift, donate,... **loan,** money, borrow,...}

Lesk algorithm: Summary

The Lesk algorithm requires an electronic dictionary of word senses (e.g. WordNet) and a lemmatizer.

It does not use any machine learning,
but it is still a useful baseline.

WSD as a learning problem

WSD as a learning problem

Supervised:

- You have a (large) **corpus annotated with word senses**
- Here, WSD is a **standard supervised learning** task

Semi-supervised (bootstrapping) approaches:

- You only have **very little annotated data** (and a lot of raw text)
- Here, WSD is a **semi-supervised learning** task

WSD as a (binary) classification task

If *w* has two different senses, we can treat WSD for *w* as a binary classification problem:

Does this occurrence of *w* have sense A or sense B?

- If *w* has multiple senses, we are dealing with a multiclass classification problem.

We can use labeled training data to train a classifier.

- Labeled = each instance of *w* is marked as A or B.
- This is a kind of supervised learning

Designing a WSD classifier

We represent each occurrence of the word *w* as a feature vector ***w***

- Now the elements of ***w*** capture the *specific* context of the token *w*
- In distributional similarities, ***w*** provides a summary of all the contexts in which *w* occurs in the training corpus.

Implementing a WSD classifier

Basic insight: The **sense of a word** in a context depends on the **words in its context**.

Features:

- **Which words in context:** all words, all/some content words
- **How large is the context?** sentence, prev/following 5 words
- Do we represent context as **bag of words** (unordered set of words) or do we care about the **position** of words (preceding/following word)?
- Do we care about **POS tags**?
- Do we represent words as they occur in the text or as their **lemma** (dictionary form)?

Decision lists

A decision list is an ordered list of yes-no questions

***bass1* = fish vs. *bass2* = music:**

1. Does '*fish*' occur in window? - Yes. => bass1
2. Is the previous word '*striped* '? - Yes. => bass1
3. Does '*guitar*' occur in window? - Yes. => bass2
4. Is the following word '*player*'? - Yes. => bass2

Learning a decision list for a word with two senses:

- Define a **feature set**: what kind of questions do you want to ask?
- Enumerate all features (questions) the training data gives answers for
- Score each feature: $$score(f_i) = \left| \log \left(\frac{P(sense_1|f_i)}{P(sense_2|f_i)} \right) \right|$$

Semi-supervised: Yarowsky algorithm

The task:

Learn a **decision list classifier** for each ambiguous word (e.g. "*plant": living/factory?*) from lots of **unlabeled sentences**.

Features used by the classifier:

- **Collocations:** *"plant life", "manufacturing plant"*
- **Nearby (± 2-10) words:** *"animal ", "automate"*

Assumption 1: **One-sense-per-collocation**

"plant" in *"plant life"* always refers to *living* plants

Assumption 2: One-sense-per-discourse

A text talks either about living plants or about factories.

Yarowsky's training regime

1. **Initialization:**
   - Label a few seed examples.
   - Train an initial classifier on these seed examples
2. **Relabel:**
   - Label all examples with current classifier.
   - Put all examples that are labeled with high confidence into a new labeled data set.
   - Optional: apply one-sense-per-discourse to correct mistakes and get additional labels
3. **Retrain:**
   - Train a new classifier on the new labeled data set.
4. Repeat 2. and 3. until convergence.

Initial state: few labels

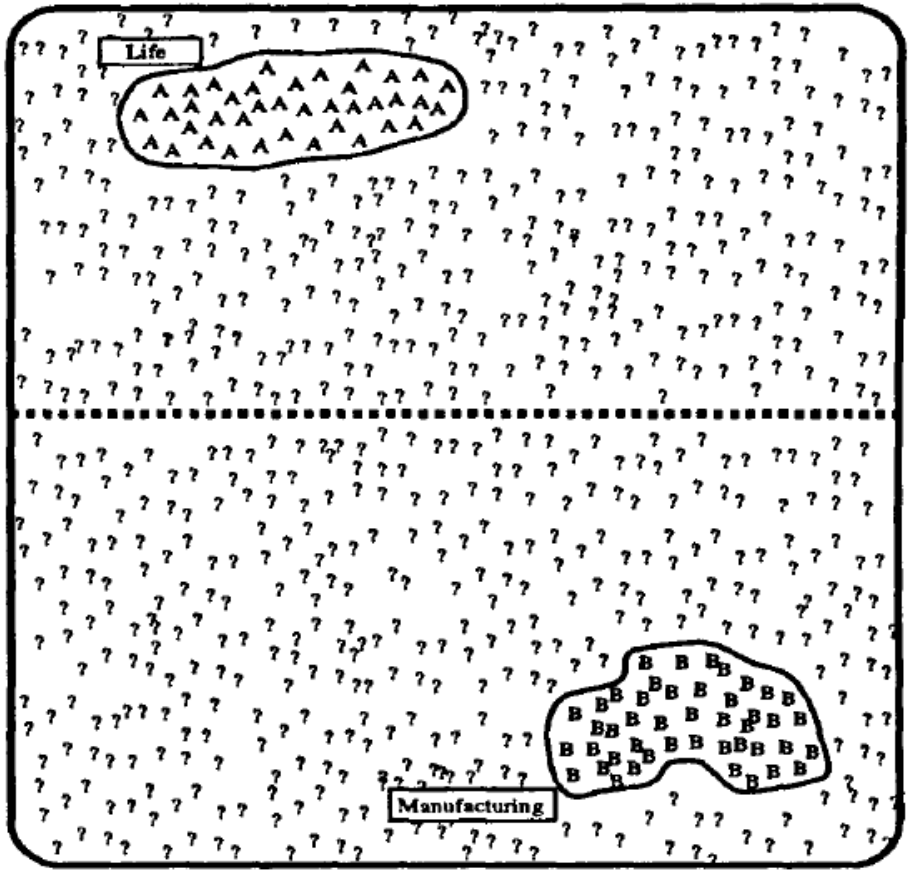


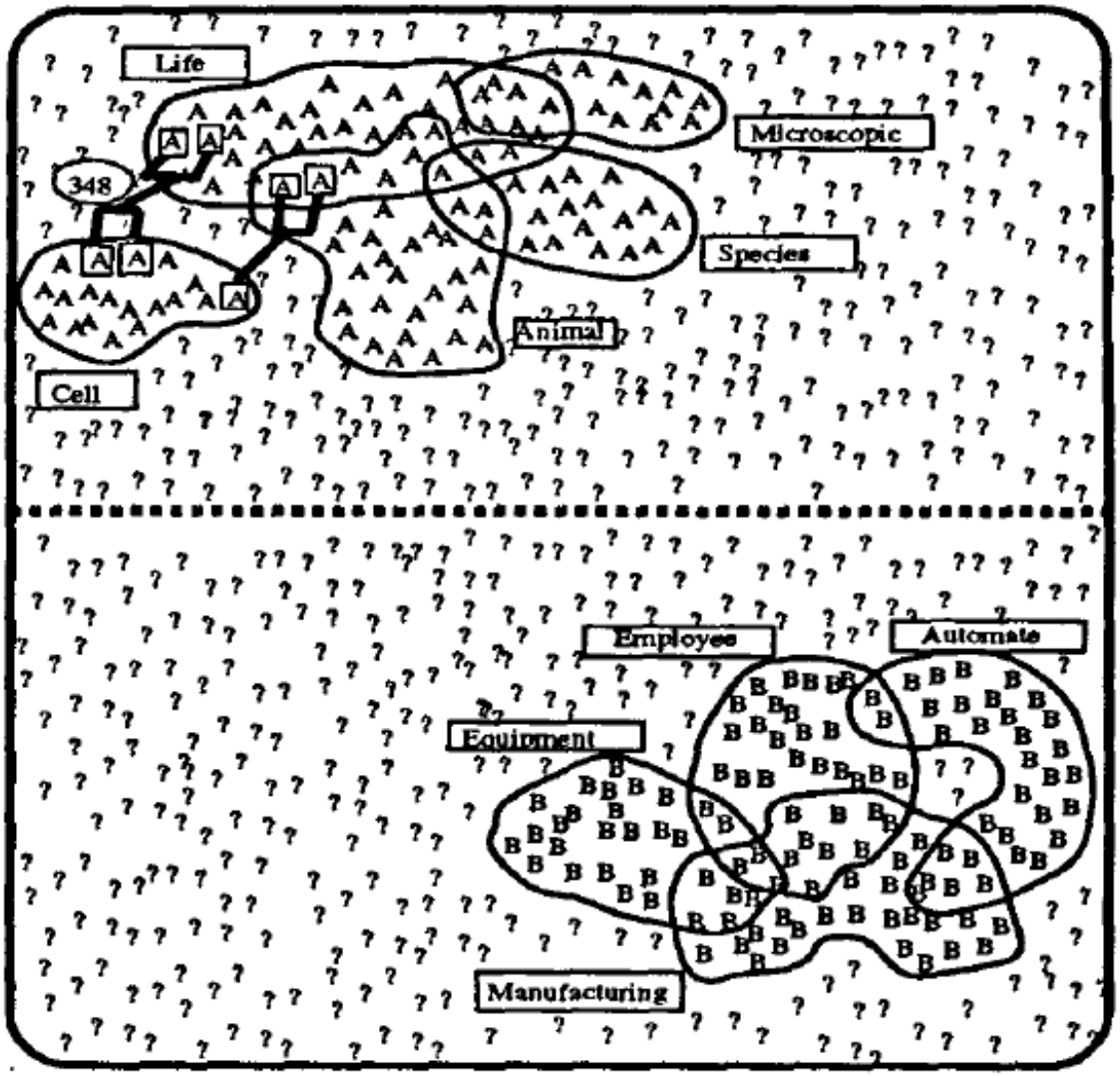
Figure 1: Sample Initial State

A = SENSE-A training example
B = SENSE-B training example
? = currently unclassified training example
Life = Set of training examples containing the collocation "life".

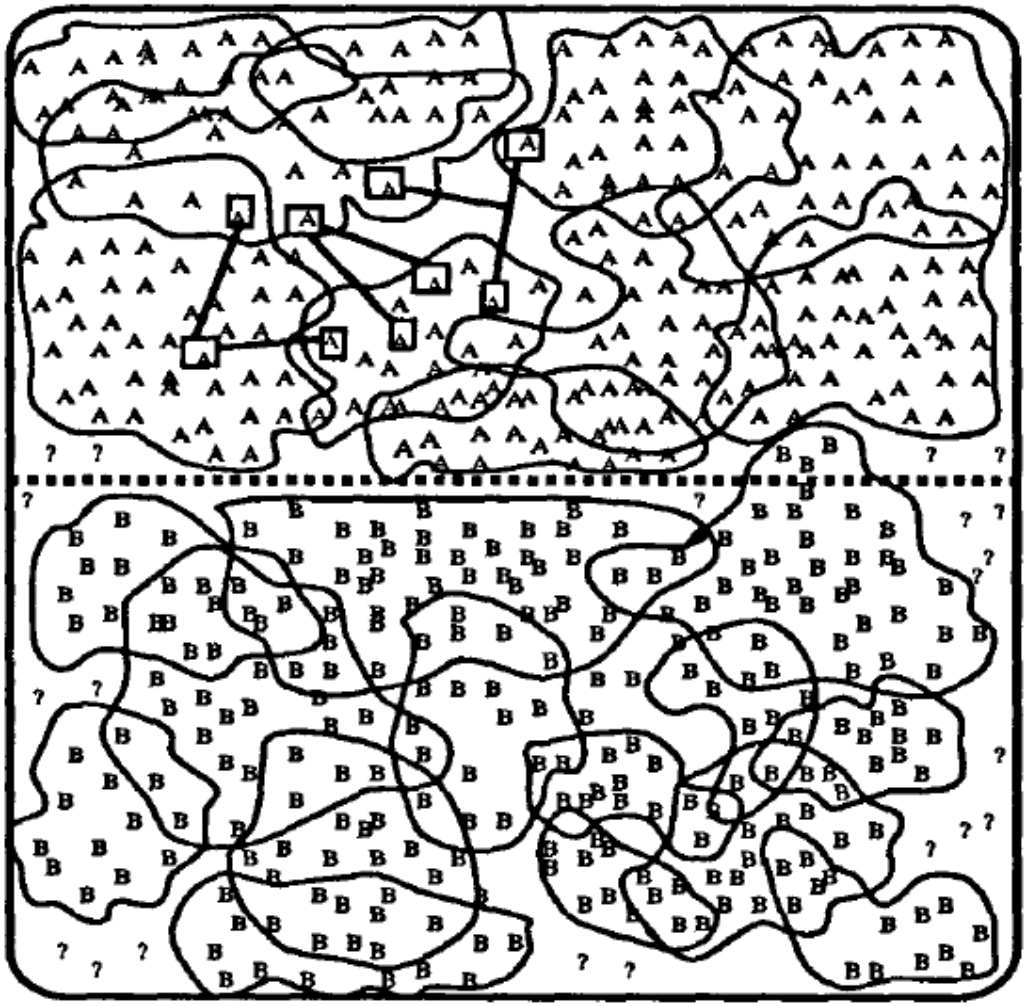
The initial decision list

Initial decision list for *plant* (abbreviated)		
LogL	Collocation	Sense
8.10	*plant* life	⇒ A
7.58	manufacturing *plant*	⇒ B
7.39	life (within ±2-10 words)	⇒ A
7.20	manufacturing (in ±2-10 words)	⇒ B
6.27	animal (within ±2-10 words)	⇒ A
4.70	equipment (within ±2-10 words)	⇒ B
4.39	employee (within ±2-10 words)	⇒ B
4.30	assembly *plant*	⇒ B
4.10	*plant* closure	⇒ B
3.52	*plant* species	⇒ A
3.48	automate (within ±2-10 words)	⇒ B
3.45	microscopic *plant*	⇒ A
	...	

Intermediate state: more labels



Final state: almost everything labeled



Initial vs. final decision lists

Initial decision list for *plant* (abbreviated)		
LogL	Collocation	Sense
8.10	***plant* life**	⇒ A
7.58	**manufacturing *plant***	⇒ B
7.39	**life (within ±2-10 words)**	⇒ A
7.20	**manufacturing (in ±2-10 words)**	⇒ B
6.27	animal (within ±2-10 words)	⇒ A
4.70	equipment (within ±2-10 words)	⇒ B
4.39	employee (within ±2-10 words)	⇒ B
4.30	assembly *plant*	⇒ B
4.10	*plant* closure	⇒ B
3.52	*plant* species	⇒ A
3.48	automate (within ±2-10 words)	⇒ B
3.45	microscopic *plant*	⇒ A
	...	

Final decision list for *plant* (abbreviated)		
LogL	Collocation	Sense
10.12	*plant* growth	⇒ A
9.68	car (within ±*k* words)	⇒ B
9.64	*plant* height	⇒ A
9.61	union (within ±*k* words)	⇒ B
9.54	equipment (within ±*k* words)	⇒ B
9.51	assembly *plant*	⇒ B
9.50	nuclear *plant*	⇒ B
9.31	flower (within ±*k* words)	⇒ A
9.24	job (within ±*k* words)	⇒ B
9.03	fruit (within ±*k* words)	⇒ A
9.02	*plant* species	⇒ A
...	...	

Summary: Yarowsky algorithm

Semi-supervised approach for WSD.

Basic idea:

- start with some minimal seed knowledge to get a few labeled examples as training data
- train a classifier
- apply this classifier to new examples
- add the most confidently classified examples to the training data
- use heuristics (one-sense-per-discourse) to add even more labeled examples to the training data
- retrain the classifier, ….

Today's key concepts

Word senses

- polysemy, homonyms
- hypernyms, hyponyms
- holonyms, meronyms

WordNet

- as a resource
- to compute thesaurus-based similarities

Word Sense disambiguation

- Lesk algorithm
- As a classification problem
- Yarowsky algorithm