





# Word Sense Disambiguation (WSD)

- Given
  - A word in context
  - A fixed inventory of potential word senses
  - Decide which sense of the word this is
- Why? Machine translation, QA, speech synthesis
- What set of senses?
  - English-to-Spanish MT: set of Spanish translations
  - Speech Synthesis: homographs like *bass* and *bow*
  - In general: the senses in a thesaurus like WordNet



# Two variants of WSD task

- Lexical Sample task
  - Small pre-selected set of target words (*line, plant*)
  - And inventory of senses for each word
  - **Supervised machine learning: train a classifier for each word**
- All-words task
  - Every word in an entire text
  - A lexicon with senses for each word
  - Data sparseness: can't train word-specific classifiers



# WSD Methods

- Supervised Machine Learning
- Thesaurus/Dictionary Methods
- Semi-Supervised Learning





# 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
- Summary of what we need:
  - the **tag set** (“sense inventory”)
  - the **training corpus**
  - A set of **features** extracted from the training corpus
  - A **classifier**



# Supervised WSD 1: WSD Tags

- What's a tag?  
A dictionary sense?
- For example, for WordNet an instance of “bass” in a text has 8 possible tags or labels (bass1 through bass8).



## 8 senses of “bass” in WordNet

1. bass - (the lowest part of the musical range)
2. bass, bass part - (the lowest part in polyphonic music)
3. bass, basso - (an adult male singer with the lowest voice)
4. sea bass, bass - (flesh of lean-fleshed saltwater fish of the family Serranidae)
5. freshwater bass, bass - (any of various North American lean-fleshed freshwater fishes especially of the genus *Micropterus*)
6. bass, bass voice, basso - (the lowest adult male singing voice)
7. bass - (the member with the lowest range of a family of musical instruments)
8. bass - (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)





# Inventory of sense tags for *bass*

WordNet Sense	Spanish Translation	Roget Category	Target Word in Context
bass <sup>4</sup>	lubina	FISH/INSECT	... fish as Pacific salmon and striped <b>bass</b> and...
bass <sup>4</sup>	lubina	FISH/INSECT	... produce filets of smoked <b>bass</b> or sturgeon...
bass <sup>7</sup>	bajo	MUSIC	... exciting jazz <b>bass</b> player since Ray Brown...
bass <sup>7</sup>	bajo	MUSIC	... play <b>bass</b> because he doesn't have to solo...



# Supervised WSD 2: Get a corpus

- Lexical sample task:
  - *Line-hard-serve* corpus - 4000 examples of each
  - *Interest* corpus - 2369 sense-tagged examples
- All words:
  - **Semantic concordance**: a corpus in which each open-class word is labeled with a sense from a specific dictionary/thesaurus.
    - SemCor: 234,000 words from Brown Corpus, manually tagged with WordNet senses
    - SENSEVAL-3 competition corpora - 2081 tagged word tokens



# SemCor

<wf pos=PRP>**He**</wf>

<wf pos=VB lemma=recognize wnsn=4 lexsns=2:31:00::>**recognized**</wf>

<wf pos=DT>**the**</wf>

<wf pos=NN lemma=gesture wnsn=1 lexsns=1:04:00::>**gesture**</wf>

<punc>.</punc>



# Supervised WSD 3: Extract feature vectors

## Intuition from Warren Weaver (1955):

“If one examines the words in a book, one at a time as through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of the words...

But if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say  $N$  words on either side, then if  $N$  is large enough one can unambiguously decide the meaning of the central word...

The practical question is : “What minimum value of  $N$  will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?”



# Feature vectors

- A simple representation for each observation  
(each instance of a target word)
  - **Vectors** of sets of feature/value pairs
  - Represented as a ordered list of values
  - These vectors represent, e.g., the window of words around the target



# Two kinds of features in the vectors

- **Collocational** features and **bag-of-words** features
  - **Collocational**
    - Features about words at **specific** positions near target word
      - Often limited to just word identity and POS
  - **Bag-of-words**
    - Features about words that occur anywhere in the window (regardless of position)
      - Typically limited to frequency counts



# Examples

- Example text (WSJ):

An electric guitar and **bass** player stand off to one side not really part of the scene

- Assume a window of +/- 2 from the target



## Examples

- Example text (WSJ)

An electric **guitar and bass** **player stand** off to one side not really part of the scene,

- Assume a window of +/- 2 from the target





# Collocational features

- Position-specific information about the words and collocations in window

- guitar and bass player stand

$$[w_{i-2}, \text{POS}_{i-2}, w_{i-1}, \text{POS}_{i-1}, w_{i+1}, \text{POS}_{i+1}, w_{i+2}, \text{POS}_{i+2}, w_{i-2}^{i-1}, w_i^{i+1}]$$

[guitar, NN, and, CC, player, NN, stand, VB, and guitar, player stand]

- word 1,2,3 grams in window of  $\pm 3$  is common



# Bag-of-words features

- “an unordered set of words” – position ignored
- Counts of words occur within the window.
- First choose a vocabulary
- Then count how often each of those terms occurs in a given window
  - sometimes just a binary “indicator” 1 or 0



## Co-Occurrence Example

- Assume we've settled on a possible vocabulary of 12 words in "bass" sentences:

*[fishing, big, sound, player, fly, rod, pound, double, runs, playing, guitar, band]*

- The vector for:

*guitar and bass player stand*

$[0,0,0,1,0,0,0,0,0,0,1,0]$





# Classification: definition

- *Input:*
  - a word  $w$  and some features  $f$
  - a fixed set of classes  $C = \{c_1, c_2, \dots, c_J\}$
- *Output:* a predicted class  $c \in C$



# Classification Methods: Supervised Machine Learning

- *Input:*
  - a word  $w$  in a text window  $d$  (which we'll call a "document")
  - a fixed set of classes  $C = \{c_1, c_2, \dots, c_J\}$
  - A training set of  $m$  hand-labeled text windows again called "documents"  $(d_1, c_1), \dots, (d_m, c_m)$
- *Output:*
  - a learned classifier  $\gamma: d \rightarrow c$



# Classification Methods: Supervised Machine Learning

- Any kind of classifier
  - Naive Bayes
  - Logistic regression
  - Neural Networks
  - Support-vector machines
  - k-Nearest Neighbors
  - ...







# Naive Bayes Intuition

- Simple (“naive”) classification method based on Bayes rule
- Relies on very simple representation of document
  - Bag of words



# I'll introduce classification with an even simpler supervised learning task

Let's classify a movie review as positive (+) or negative (-)

Suppose we have lots of reviews labeled as (+) or (-) and I give you a new review.

Given: the words in this new movie review

Return: one of 2 classes: + or -





# The bag of words representation

$Y$  (

seen	2
sweet	1
whimsical	1
recommend	1
happy	1
...	...

)

=  $C$





# Bayes' Rule Applied to Documents and Classes

- For a document  $d$  and a class  $c$

$$P(c | d) = \frac{P(d | c)P(c)}{P(d)}$$



# Naive Bayes Classifier (I)

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(c | d)$$

MAP is “maximum a posteriori” = most likely class

$$= \operatorname{argmax}_{c \in C} \frac{P(d | c)P(c)}{P(d)}$$

Bayes Rule

$$= \operatorname{argmax}_{c \in C} P(d | c)P(c)$$

Dropping the denominator



## Naive Bayes Classifier (II)

$$C_{MAP} = \operatorname{argmax}_{c \in C} P(d | c)P(c)$$

$$= \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

Document  $d$   
represented as  
features  
 $x_1 \dots x_n$



# Naive Bayes Classifier (IV)

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

$O(|X|^n \cdot |C|)$  parameters

Could only be estimated if a very, very large number of training examples was available.

How often does this class occur?

We can just count the relative frequencies in a corpus





# Multinomial Naive Bayes Independence Assumptions

$$P(x_1, x_2, \dots, x_n | c)$$

- **Bag of Words assumption:** Assume position doesn't matter
- **Conditional Independence:** Assume the feature probabilities  $P(x_i | c_j)$  are independent given the class  $c$ .

$$P(x_1, \dots, x_n | c) = P(x_1 | c) \cdot P(x_2 | c) \cdot P(x_3 | c) \cdot \dots \cdot P(x_n | c)$$



# Multinomial Naive Bayes Classifier

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c) P(c)$$

$$c_{NB} = \operatorname{argmax}_{c \in C} P(c_j) \prod_{x \in X} P(x | c)$$



# Applying Multinomial Naive Bayes Classifiers to Text Classification

positions = all word positions in test document

$$c_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i \in \text{positions}} P(x_i | c_j)$$







# Learning the Multinomial Naive Bayes Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{\text{doccount}(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$



# Parameter estimation

$$\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

fraction of times word  $w_i$  appears  
among all words in documents of topic  $c_j$

- Create mega-document for topic  $j$  by concatenating all docs in this topic
  - Use frequency of  $w$  in mega-document



# Problem with Maximum Likelihood

- What if we have seen no training documents with the word *fantastic* and classified in the topic **positive** (*thumbs-up*)?

$$\hat{P}(\text{"fantastic"} \mid \text{positive}) = \frac{\text{count}(\text{"fantastic"}, \text{positive})}{\sum_{w \in V} \text{count}(w, \text{positive})} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$c_{MAP} = \operatorname{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$





# Laplace (add-1) smoothing for Naïve Bayes

$$\hat{P}(w_i | c) = \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)}$$
$$= \frac{\text{count}(w_i, c) + 1}{\left( \sum_{w \in V} \text{count}(w, c) \right) + |V|}$$



# Multinomial Naïve Bayes: Learning

- From training corpus, extract *Vocabulary*
- Calculate  $P(c_j)$  terms
  - For each  $c_j$  in  $C$  do
    - $docs_j \leftarrow$  all docs with class =  $c_j$
- Calculate  $P(w_k | c_j)$  terms
  - $Text_j \leftarrow$  single doc containing all  $docs_j$
  - For each word  $w_k$  in *Vocabulary*
    - $n_k \leftarrow$  # of occurrences of  $w_k$  in  $Text_j$

$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

$$P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$







# Applying Naive Bayes to WSD

- $P(c)$  is the prior probability of that sense
  - Counting in a labeled training set.
- $P(w|c)$  conditional probability of a word given a particular sense
  - $P(w|c) = \text{count}(w,c)/\text{count}(c)$
- We get both of these from a tagged corpus like SemCor
  
- Can also generalize to look at other features besides words.
  - Then it would be  $P(f|c)$ 
    - Conditional probability of a feature given a sense



$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w|c) = \frac{\text{count}(w,c)+1}{\text{count}(c)+|V|}$$

	Doc	Words	Class
Training	1	fish smoked fish	f
	2	fish line	f
	3	fish haul smoked	f
	4	guitar jazz line	g
Test	5	line guitar jazz jazz	?

**Priors:**

$$P(f) = \frac{3}{4}$$

$$P(g) = \frac{1}{4}$$

$V = \{\text{fish, smoked, line, haul, guitar, jazz}\}$

**Conditional Probabilities:**

$$P(\text{line}|f) = (1+1) / (8+6) = 2/14$$

$$P(\text{guitar}|f) = (0+1) / (8+6) = 1/14$$

$$P(\text{jazz}|f) = (0+1) / (8+6) = 1/14$$

$$P(\text{line}|g) = (1+1) / (3+6) = 2/9$$

$$P(\text{guitar}|g) = (1+1) / (3+6) = 2/9$$

$$46 \quad P(\text{jazz}|g) = (1+1) / (3+6) = 2/9$$

**Choosing a class:**

$$P(f|d5) = \frac{3}{4} * \frac{2}{14} * \left(\frac{1}{14}\right)^2 * \frac{1}{14} \approx 0.00003$$

$$P(g|d5) = \frac{1}{4} * \frac{2}{9} * \left(\frac{2}{9}\right)^2 * \frac{2}{9} \approx 0.0006$$





# WSD Evaluations and baselines

- Best evaluation: **extrinsic ('end-to-end', 'task-based') evaluation**
  - Embed WSD algorithm in a task and see if you can do the task better!
- What we often do for convenience: **intrinsic evaluation**
  - Exact match **sense accuracy**
    - % of words tagged identically with the human-manual sense tags
  - Usually evaluate using **held-out data** from same labeled corpus
- Baselines
  - Most frequent sense
  - The Lesk algorithm





# Most Frequent Sense

- WordNet senses are ordered in frequency order
- So “most frequent sense” in WordNet = “take the first sense”
- Sense frequencies come from the *SemCor* corpus

Freq	Synset	Gloss
338	plant <sup>1</sup> , works, industrial plant	buildings for carrying on industrial labor
207	plant <sup>2</sup> , flora, plant life	a living organism lacking the power of locomotion
2	plant <sup>3</sup>	something planted secretly for discovery by another
0	plant <sup>4</sup>	an actor situated in the audience whose acting is rehearsed but seems spontaneous to the audience



# Ceiling

- Human inter-annotator agreement
  - Compare annotations of two humans
  - On same data
  - Given same tagging guidelines
- Human agreements on all-words corpora with WordNet style senses
  - 75%-80%





# The Simplified Lesk algorithm

- Let's disambiguate “**bank**” in this sentence:

The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

- given the following two WordNet senses:

bank <sup>1</sup>	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	“he cashed a check at the bank”, “that bank holds the mortgage on my home”
bank <sup>2</sup>	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 currents”



# The Simplified Lesk algorithm

Choose sense with most word overlap between gloss and context  
(not counting function words)

The **bank** can guarantee **deposits** will eventually cover future tuition costs because it invests in adjustable-rate **mortgage** securities.

bank <sup>1</sup>	Gloss:	a financial institution that accepts <b>deposits</b> and channels the money into lending activities
	Examples:	“he cashed a check at the bank”, “that bank holds the <b>mortgage</b> on my home”
bank <sup>2</sup>	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 currents”



# The Corpus Lesk algorithm

- Assumes we have some sense-labeled data (like SemCor)
- Take all the sentences with the relevant word sense:  
*These short, "streamlined" meetings usually are sponsored by local **banks**<sup>1</sup>, Chambers of Commerce, trade associations, or other civic organizations.*
- Now add these to the gloss + examples for each sense, call it the “signature” of a sense.
- Choose sense with most word overlap between context and signature.



# Corpus Lesk: IDF weighting

- Instead of just removing function words
  - Weigh each word by its 'promiscuity' across documents
  - Down-weights words that occur in every 'document' (gloss, example, etc)
  - These are generally function words, but is a more fine-grained measure
- Weigh each overlapping word by **inverse document frequency**



# Corpus Lesk: IDF weighting

- Weigh each overlapping word by **inverse document frequency**
  - $N$  is the total number of documents
  - $df_i$  = “document frequency of word  $i$ ”
  - = # of documents with word  $i$

$$\text{idf}_i = \log \left( \frac{N}{df_i} \right)$$

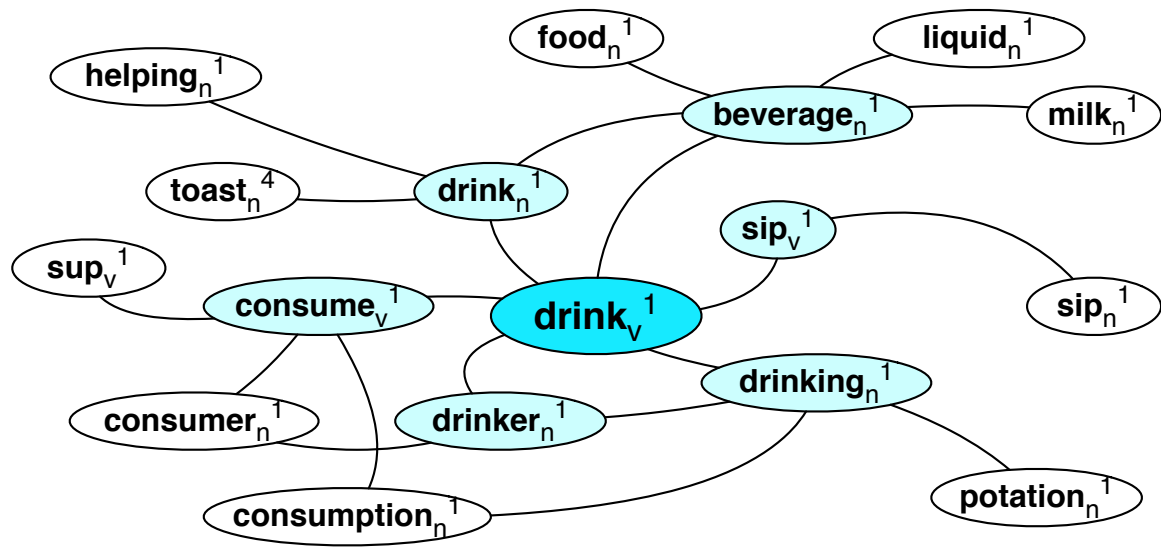
$$\text{score}(\text{sense}_i, \text{context}_j) = \sum_{w \in \text{overlap}(\text{signature}_i, \text{context}_j)} \text{idf}_w$$





# Graph-based methods

- First, WordNet can be viewed as a graph
  - senses are nodes
  - relations (hypernymy, meronymy) are edges
  - Also add edge between word and unambiguous gloss words





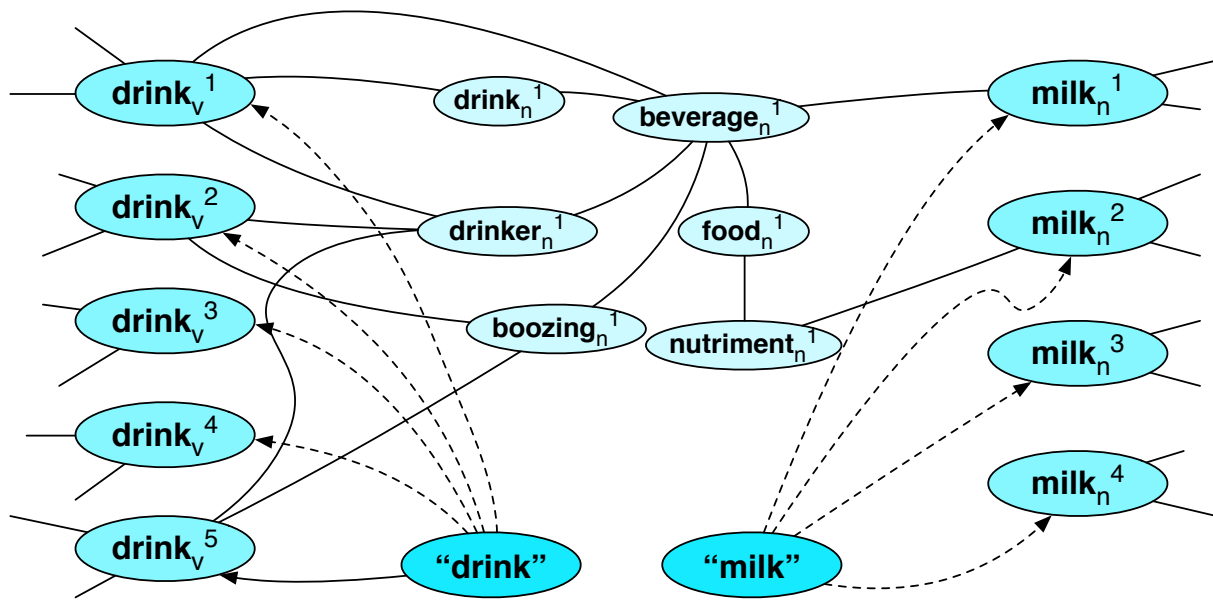
# How to use the graph for WSD

- Insert target word and words in its sentential context into the graph, with directed edges to their senses

“She drank some milk”

- Now choose the *most central* sense

Add some probability to “drink” and “milk” and compute node with highest “pagerank”







# Semi-Supervised Learning

**Problem:** supervised and dictionary-based approaches require large hand-built resources

What if you don't have so much training data?

**Solution:** Bootstrapping

Generalize from a very small hand-labeled seed-set.



# Bootstrapping

- For bass
  - Rely on “One sense per collocation” rule
    - A word reoccurring in collocation with the same word will almost surely have the same sense.
  - the word `p1ay` occurs with the music sense of bass
  - the word `f1sh` occurs with the fish sense of bass



# Sentences extracting using “fish” and “play”

We need more good teachers – right now, there are only a half a dozen who can **play** the free **bass** with ease.

An electric guitar and **bass player** stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.

The researchers said the worms spend part of their life cycle in such **fish** as Pacific salmon and striped **bass** and Pacific rockfish or snapper.

And it all started when **fishermen** decided the striped **bass** in Lake Mead were too skinny.

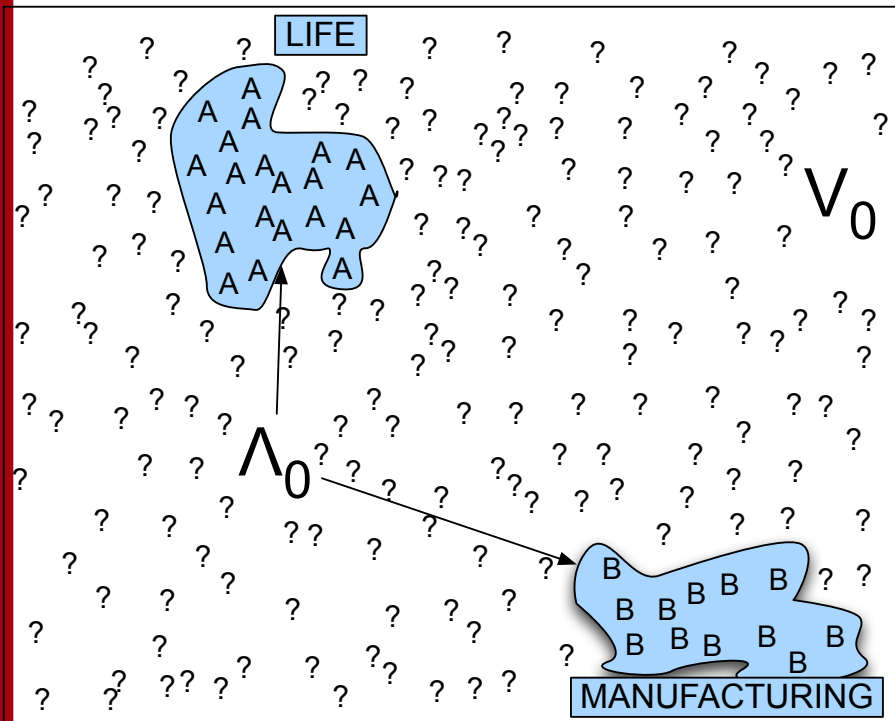


# Summary: generating seeds

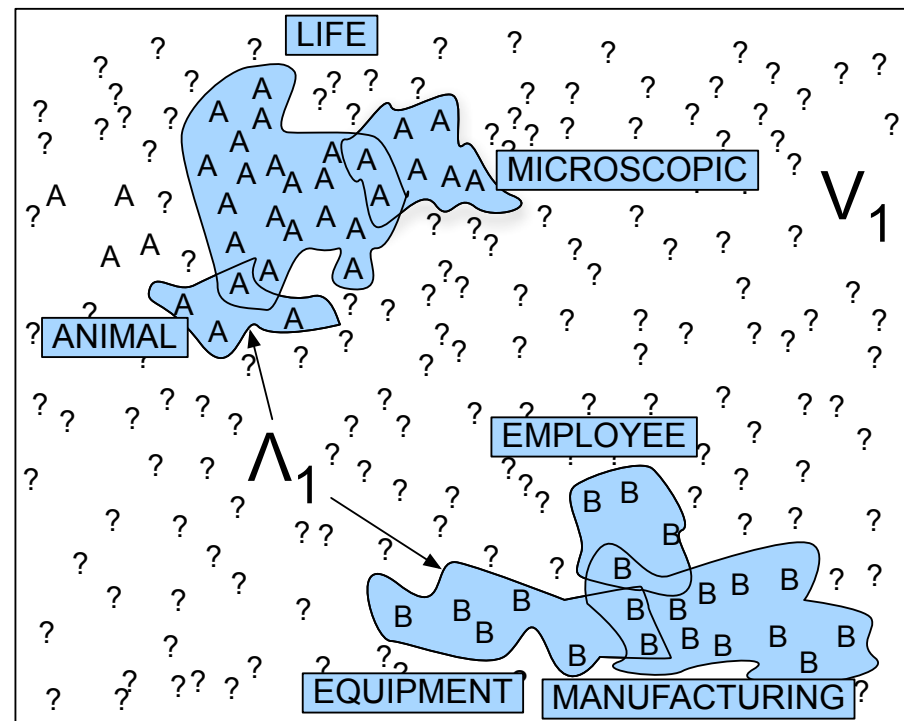
- 1) Hand labeling
- 2) “One sense per collocation”:
  - A word reoccurring in collocation with the same word will almost surely have the same sense.
- 3) “One sense per discourse”:
  - The sense of a word is highly consistent within a document - Yarowsky (1995)
  - (At least for non-function words, and especially topic-specific words)



# Stages in the Yarowsky bootstrapping algorithm for the word “plant”



(a)



(b)





# Summary

- Word Sense Disambiguation: choosing correct sense in context
- Applications: MT, QA, etc.
- Three classes of Methods
  - Supervised Machine Learning: Naive Bayes classifier
  - Thesaurus/Dictionary Methods
  - Semi-Supervised Learning
- Main intuition
  - There is lots of information in a word's context
  - Simple algorithms based just on word counts can be surprisingly good