# Natural Language Processing
# CS 6320
*Lecture 14*
*Word Sense Disambiguation*

Instructor: Sanda Harabagiu

# 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***

# Word Senses

- The meaning of a word distinguished in a given context

- *Word sense representations*
  - *With respect to a dictionary*

    chair = a seat for one person, with a support for the back; "he put his coat over the back of the chair and sat down"

    chair = the position of professor; "he was awarded an endowed chair in economics"

  - *With respect to the translation in a second language*

    chair = chaise

    chair = directeur

  - *With respect to the context where it occurs (discrimination)*

    *"Sit on a chair" "Take a seat on this chair"*

    *"The chair of the Math Department" "The chair of the meeting"*

# Two variants of WSD task

1. *Lexical Sample task*
   - *Small pre-selected set of target words (line, plant)*
   - *And inventory of senses for each word*
   - ***<u>Supervised machine learning</u>: train a classifier for each word***
2. *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*
  - based on a labeled training set
  - the learning system has:
    - a training set of feature-encoded inputs AND
    - their appropriate sense label (category)

- *Thesaurus/Dictionary Methods*
  - use of external lexical resources such as dictionaries and thesauri
  - discourse properties

- *Semi-Supervised Learning*
  - the learning system has:
    - a training set of feature-encoded inputs BUT
    - NOT their appropriate sense label (category)

# 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*

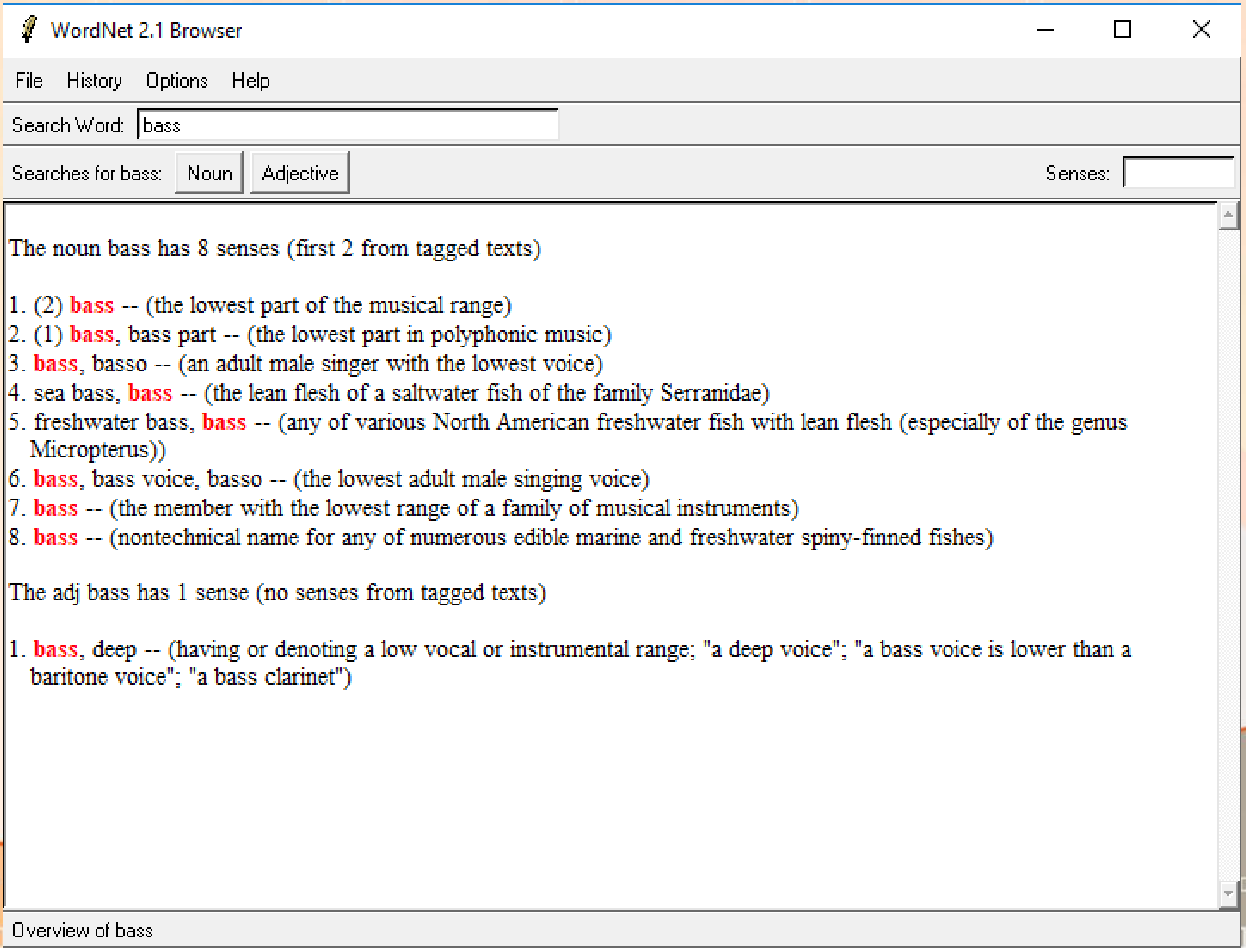
- *Summary of what we need:*
  - *the **tag set** ("sense inventory")*
  - *the **training corpus***
  - *A set of **features** extracted from the training corpus*
  - *A **classifier***

# 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).*



# 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 lexsn=2:31:00::>**recognized**</wf>*

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

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

*<punc>.</punc>*

# Supervised WSD

*How to 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*
  - *These vectors represent, e.g., the window of words around the target*

❑ *Two kinds of features in the vectors*

1. ***Collocational features***
   - *Features about words at **specific** positions near target word*
     - ➢ *Often limited to just word identity and POS*
2. ***Bag-of-words features***
   - *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*

# Collocational features

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

*guitar and bass player stand*

$$\left[w_{i-2}, \mathrm{POS}_{i-2}, w_{i-1}, \mathrm{POS}_{i-1}, w_{i+1}, \mathrm{POS}_{i+1}, w_{i+2}, \mathrm{POS}_{i+2}, w_{i-2}^{i-1}, w_{i}^{i+1}\right]$$

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

- ➢ *word 1,2,3 grams in window of ±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 for WSD: definition

- *Input:*
  - *a word $w$ and some features $f$*
  - *a fixed set of classes* $C = \{c_1, c_2, \ldots, c_J\}$

- *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 $C = \{c_1, c_2, \ldots, c_J\}$*
  - *A training set of $m$ hand-labeled text windows again called "documents" $(d_1,c_1),\ldots,(d_m,c_m)$*
- *Output:*
  - *a learned classifier $\gamma: d \rightarrow c$*

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

# 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) = count(w,c)/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*

# Applying Naive Bayes to WSD: Details

V = {fish, smoked, line, haul, guitar, jazz}

| | 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 | ? |

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

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

**Priors:**

*P*(*f*)= 3/4

*P*(*g*)= 1/4

**Conditional Probabilities:**

P(line|*f*) = (1+1) / (8+6) = 2/14

P(guitar|*f*) = (0+1) / (8+6) = 1/14

P(jazz|*f*) = (0+1) / (8+6) = 1/14

P(line|*g*) = (1+1) / (3+6) = 2/9

P(guitar|*g*) = (1+1) / (3+6) = 2/9

P(jazz|*g*) = (1+1) / (3+6) = 2/9

**Choosing a class:**

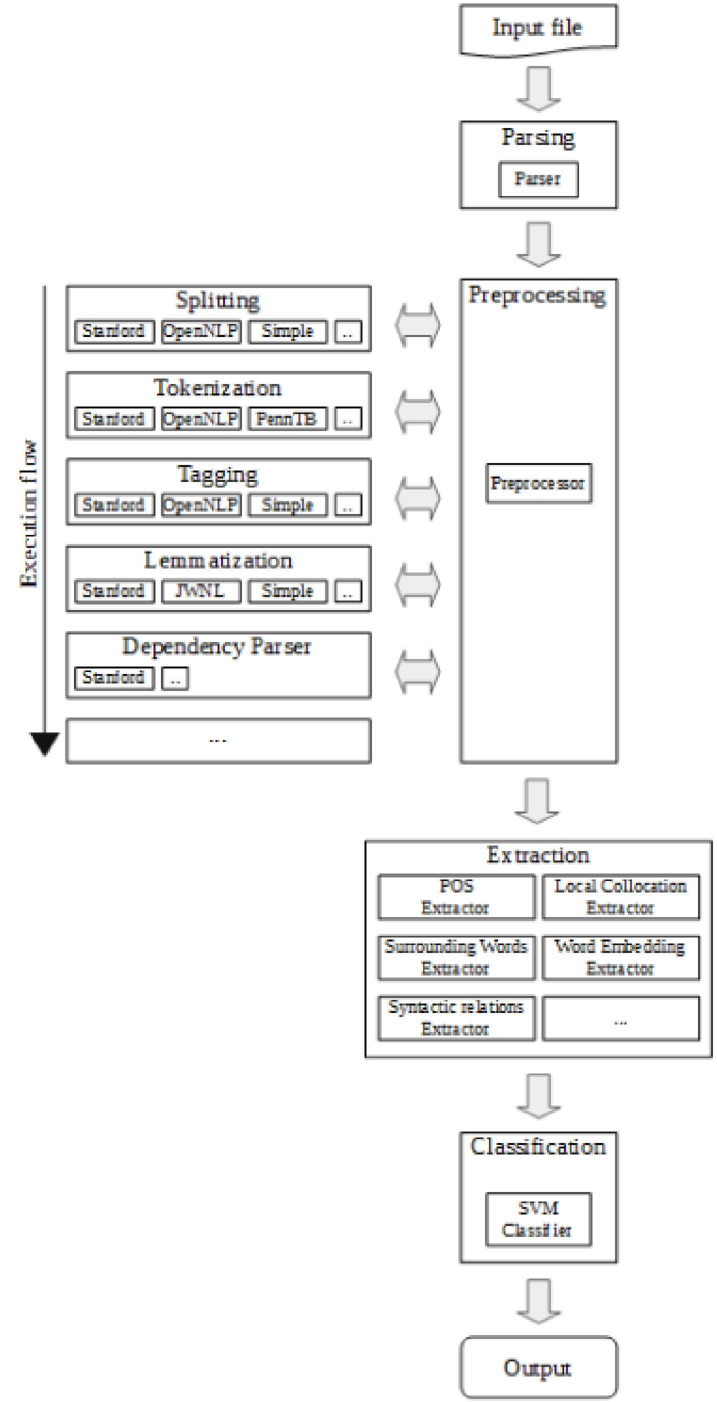
P(f|d5) $\propto$ 3/4 * 2/14 * (1/14)$^2$ * 1/14 $\approx$ 0.00003

P(g|d5) $\propto$ 1/4 * 2/9 * (2/9)$^2$ * 2/9 $\approx$ 0.0006

## ➢ *SUPWSD: A Flexible Toolkit for Supervised Word Sense Disambiguation*

The implementation of a state-of-the-art supervised WSD system, together with a Natural Language Processing pipeline for preprocessing and feature extraction.

*http://github.com/SI3P/SupWSD*



# Neural WSD method

## ➢ *Neural Architecture*

The architecture relies on 3 layers:

1. The input layer, which takes directly the words in a vector form, from a pre-trained word embeddings model.
2. The hidden layer, composed of bidirectional LSTM units (Hochreiter and Schmidhuber, 1997).
3. The output layer, which represents foreach word in the input, a probability distribution over all senses in the output vocabulary used, thanks to a classical *softmax function*.

Code available at:

https://github.com/getalp/disambiguate

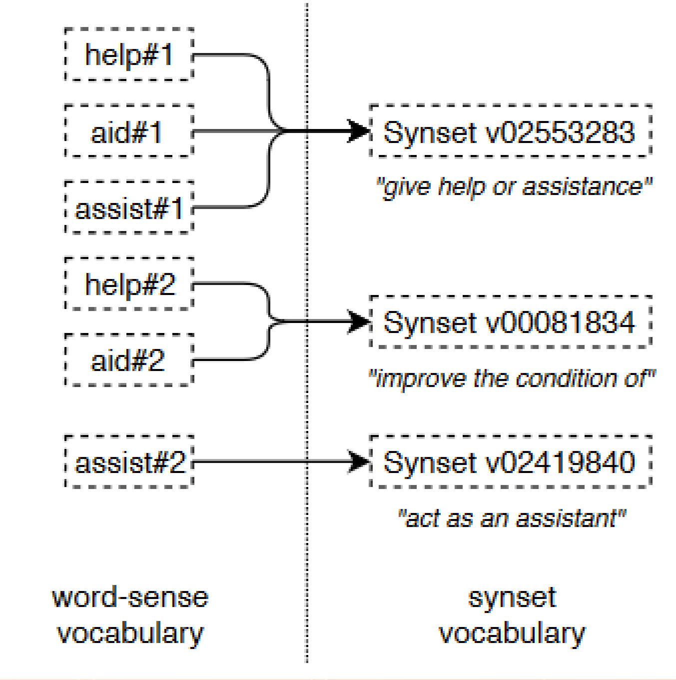
Paper available at:

https://arxiv.org/pdf/1811.00960v1.pdf

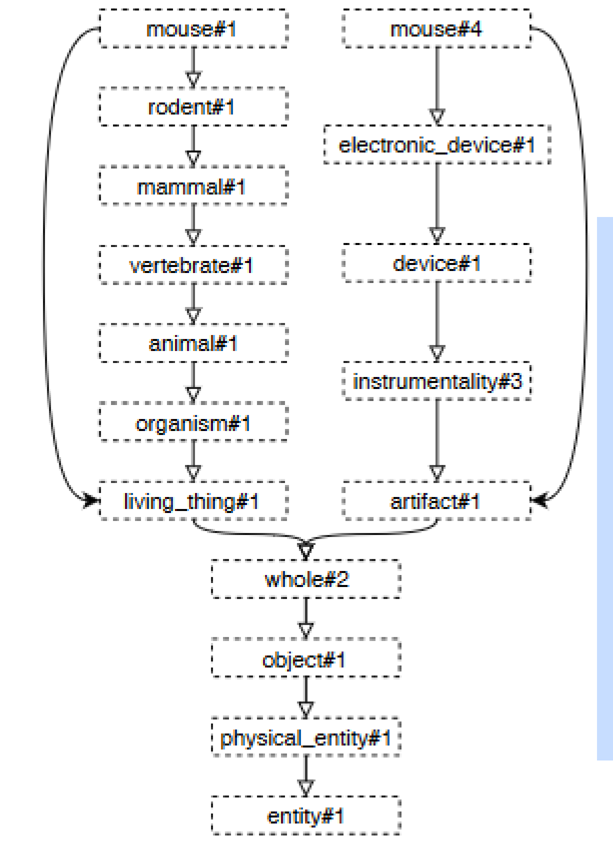
# Several innovations

1. *Word-sense to synset vocabulary reduction*

https://github.com/getalp/disambiguate




2. *Sense Vocabulary Reduction through Hypernymy and Hyponymy Relationships*




When applied on WordNet, the number of synsets in the vocabulary drops from 117,659 to 39,147 (approx. 66% of reduction), and When applied on the SemCor, It contains 12,779 different synsets, which counts for 32% of coverage.

# 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$^1$, works, industrial plant | buildings for carrying on industrial labor |
| 207 | plant$^2$, flora, plant life | a living organism lacking the power of locomotion |
| 2 | plant$^3$ | something planted secretly for discovery by another |
| 0 | plant$^4$ | 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%*

# Dictionary and Thesaurus Methods

## 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:*

| | | |
|---|---|---|
| $bank^1$ | 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" |
| $bank^2$ | 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$^1$ | 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$^2$ | 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 in the 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**[1], 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*

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

$$score(sense_i,\ context_j) = \sum_{w \in overlap(signature_i,\ context_j)} \mathrm{idf}_w$$

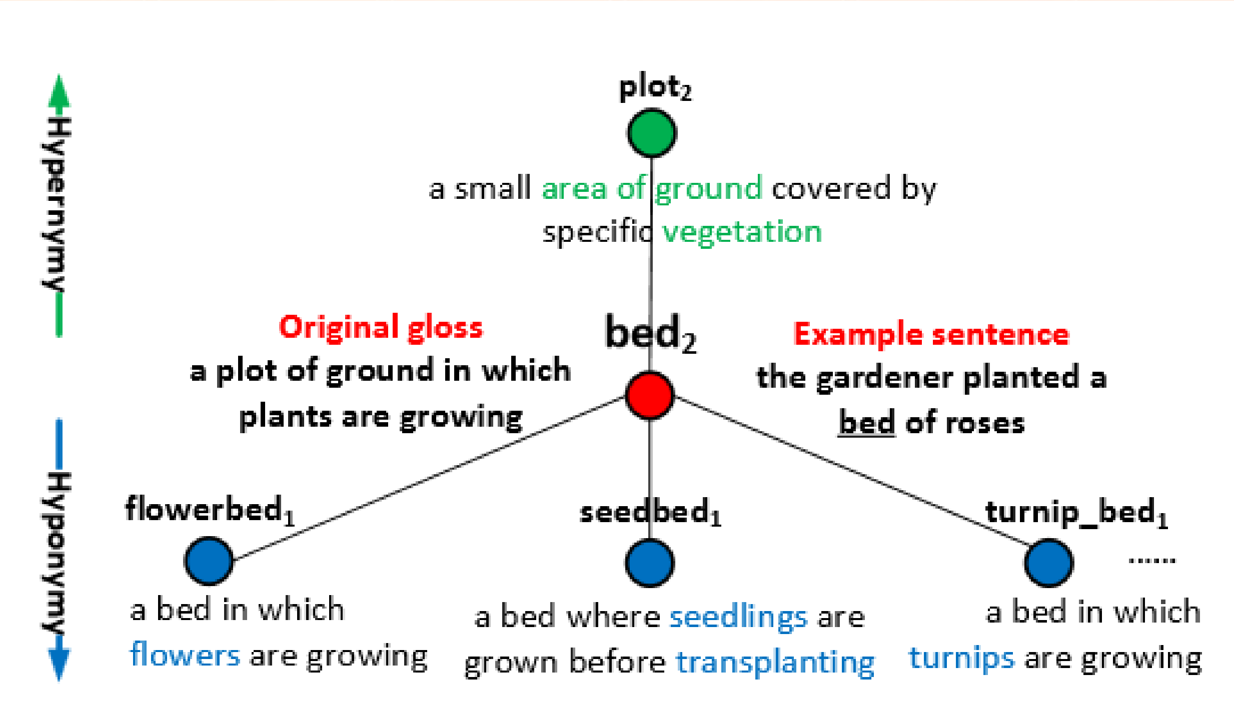
# Incorporating Glosses into Neural WSD

- *Integrate the context and glosses of the target word into a unified framework in order to make full use of both labeled data and lexical knowledge:*

The paper: [*https://arxiv.org/pdf/1805.08028v2.pdf*](https://arxiv.org/pdf/1805.08028v2.pdf)

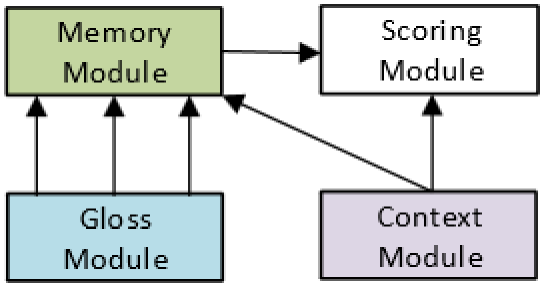
*The Code: https://github.com/luofuli/word-sense-disambiguation*

The glosses of hypernyms and hyponyms can enrich the original gloss information as well as help to build a better sense representation.



# A model for gloss-augmented WSD neural network

- The gloss-augmented (GAS) WSD model uses a neural network which integrates the context and the glosses of the target word into a unified framework.




- It consists of four modules:

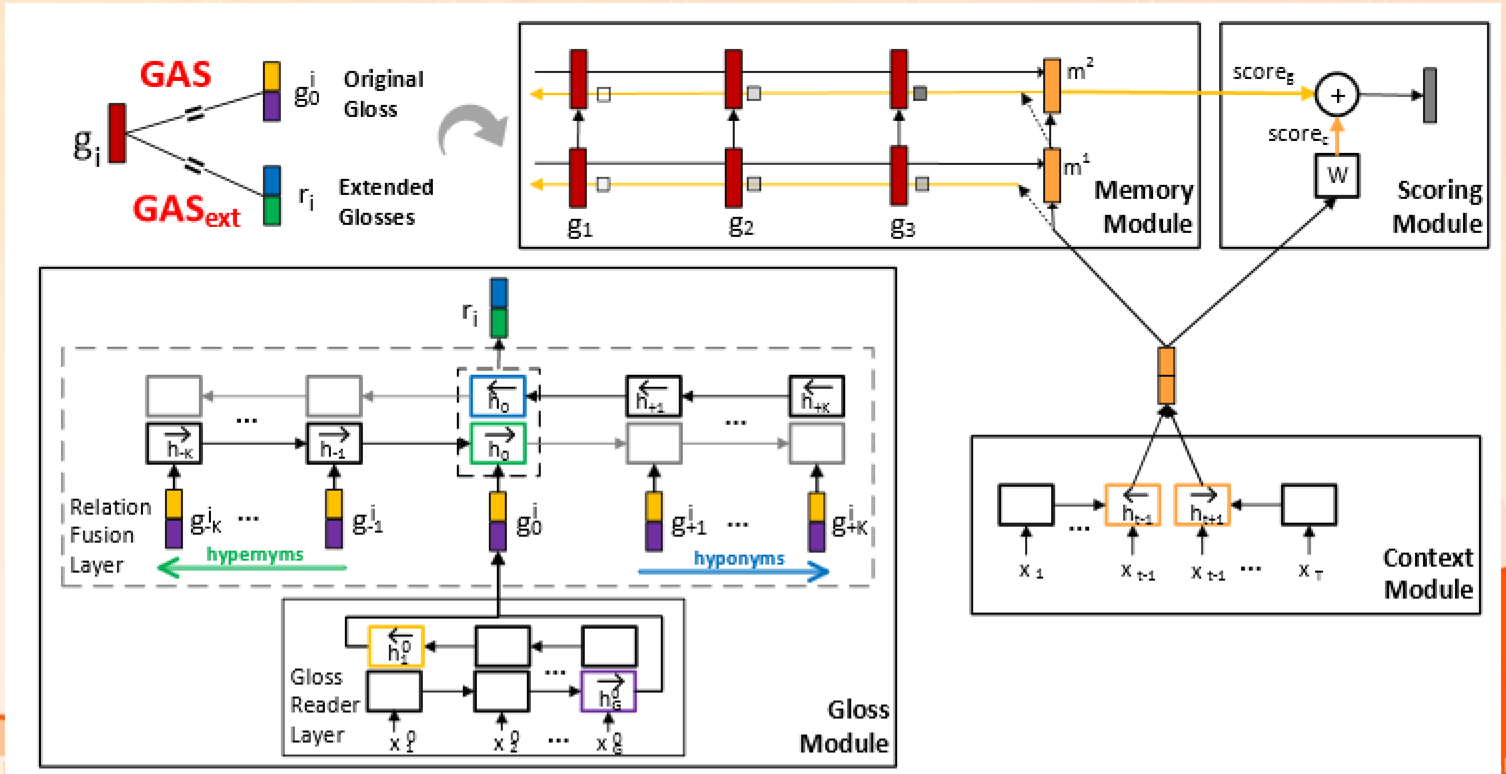
1. *The Context Module: encodes the local context (the sequence of surrounding words) of the target word into a distributed vector representation;*
2. *The Gloss Module: encodes all the glosses of the target word into separate vector representations: we can get $|s_t|$ word sense gloss representations according, if the word t has $s_t$ different senses.*
3. *The Memory Module: The memory module is employed to model <u>the semantic relationship between the context embedding and gloss embedding</u> produced by the context module and the gloss module respectively.*
4. *The Scoring Module: generates a probability distribution over all the possible senses of the target word by considering both the labeled contexts and the gloss knowledge.*

# Detailed Neural Architecture

The **context module** encodes the adjacent words surrounding the target word into a vector *c*. The **gloss module** encodes the original gloss or extended glosses into a vector $g_i$.

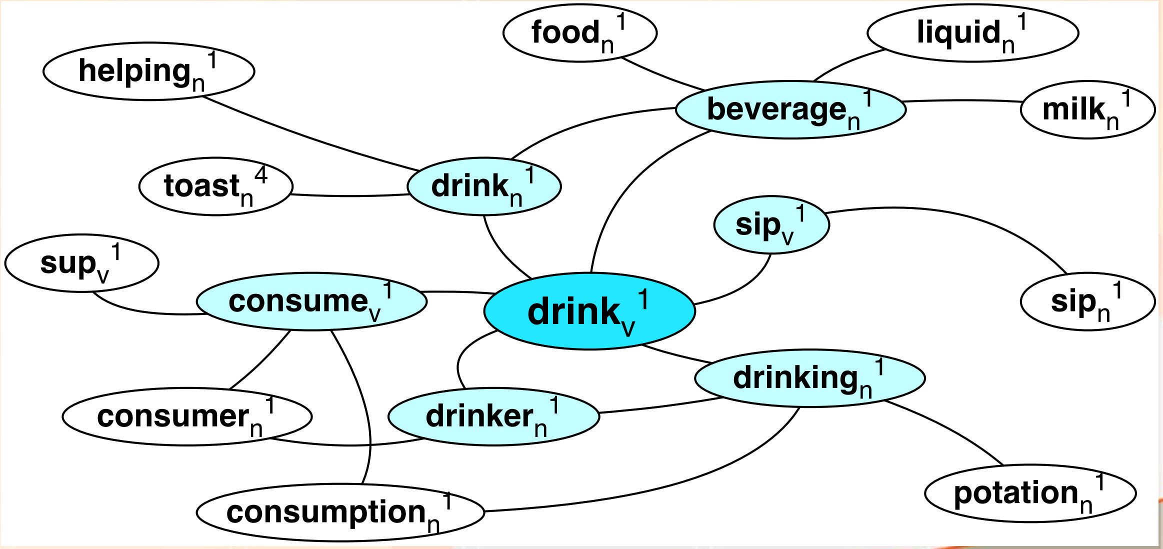
In the **memory module**, we calculate the inner relationship (as attention) between context and each gloss $g_i$. and then update the memory as $m_i$ at pass *i*.

In the **scoring module**, we make final predictions based on the last pass's attention of memory module and the context vector *c*. Note that GAS only uses the original gloss, while $\text{GAS}_{\text{ext}}$ uses the extended glosses through hypernymy and hyponymy relations. In other words, the relation fusion layer (grey dotted box) only belongs to $\text{GAS}_{\text{ext}}$.



# Graph-based methods

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



# How to use the graph for WSD ???

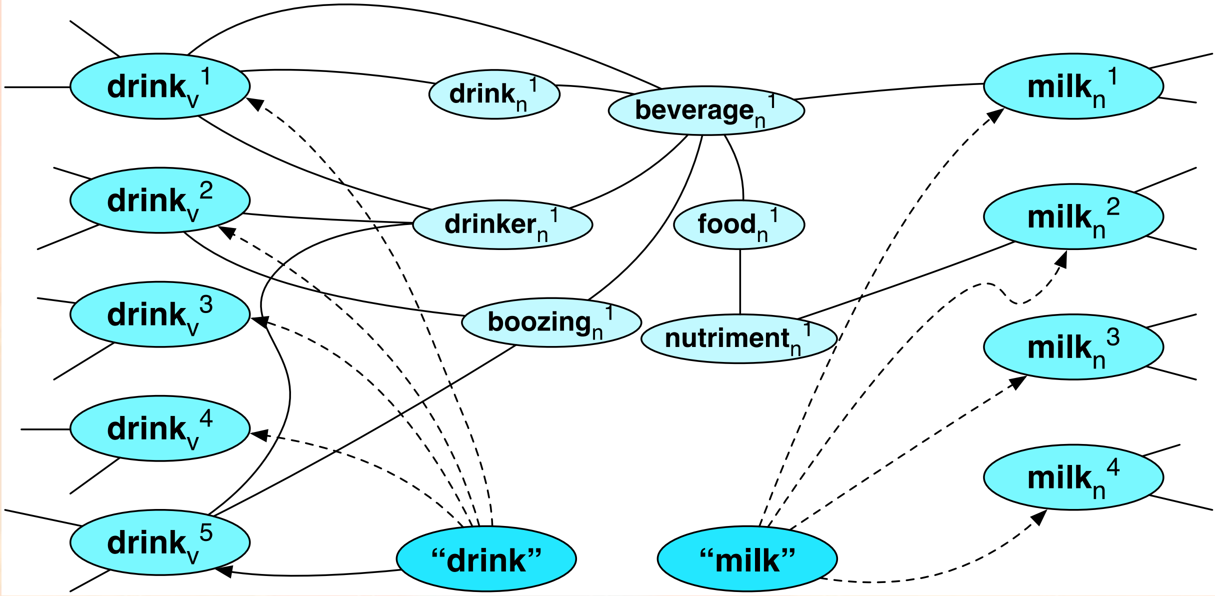
- Insert target word and words in its sentential context into the graph, with directed edges to their senses

*"She drank some milk"*

- *Choose the most central sense*

*Add some probability to "drink" and "milk" and compute node with highest "pagerank":*

*(Agirre and Soroa, 2009)*




Personalizing PageRank for Word Sense Disambiguation

https://www.aclweb.org/anthology/E09-1005.pdf

# 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* `play` *occurs with the music sense of bass*
  - *the word* `fish` *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.<br><br>An electric guitar and **bass play**er 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.<br><br>And it all started when **fish**ermen 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)*

# A detailed example: STEP 1

- Step 1: for a polysemous word *w*, identify all its examples in a given corpus and store their contexts as lines in an initially untagged training set.

| 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 |
| ? | ... ... |

# Step 2

- For each sense of the word, identify a relatively small number of training examples representative of that sense.

⇒ **Solution:** hand-tag a subset of the training sentences

- Yarowsky had a better solution:
  - identify a small number of seed collocations representative of each sense and tag all training examples containing the seed collocates with the sense label.
- **Example:** word: plant
  - sense A: collocation: plant life
  - sense B: collocation: manufacturing plant

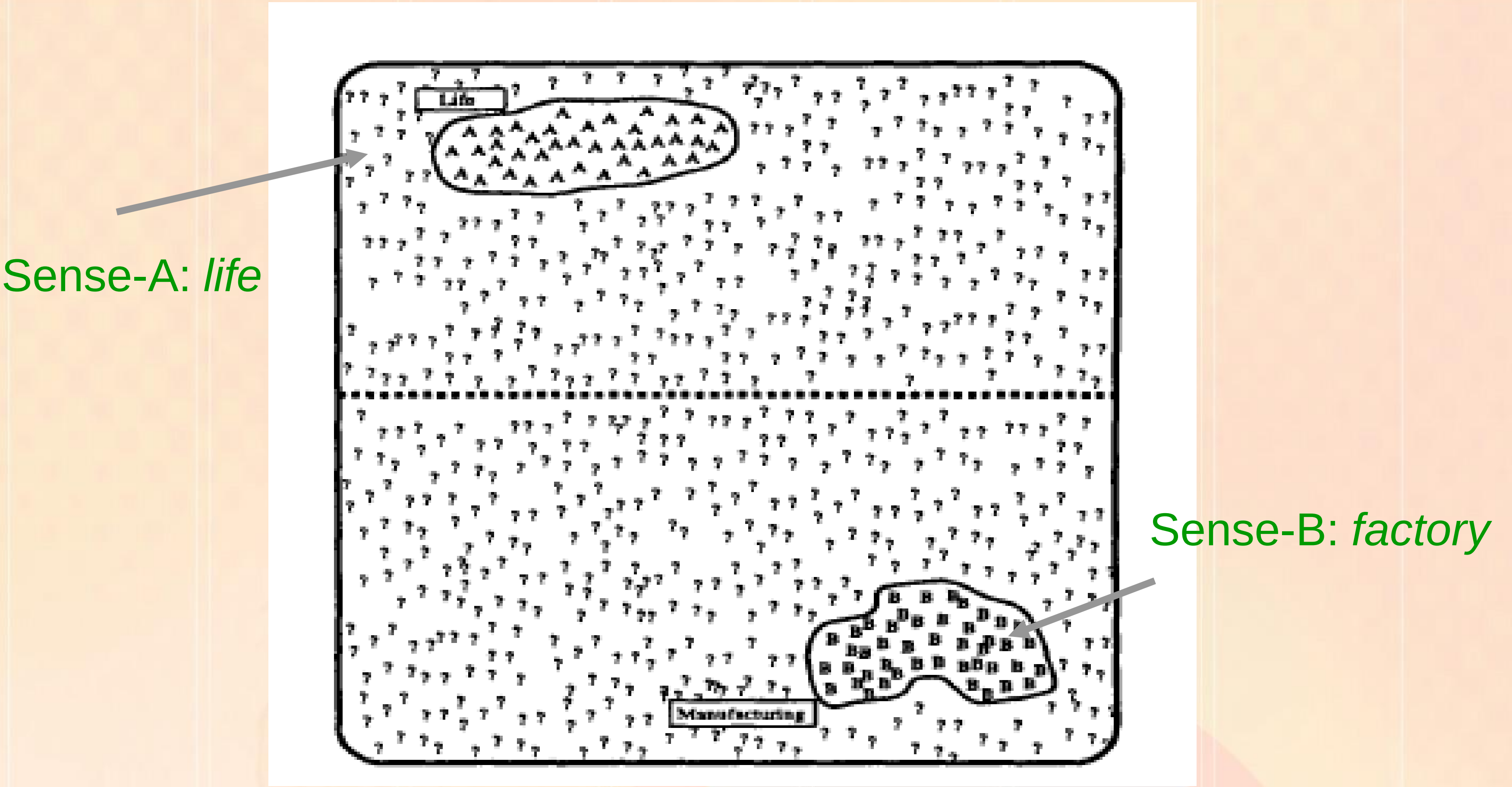
# Training Examples

| Sense | Training Examples (Keyword in Context) |
|---|---|
| A | used to strain microscopic *plant* **life** from the ... |
| A | ... zonal distribution of *plant* **life** . ... |
| A | close-up studies of *plant* **life** and natural ... |
| A | too rapid growth of aquatic *plant* **life** in water ... |
| A | ... the proliferation of *plant* and animal **life** ... |
| A | establishment phase of the *plant* virus **life** cycle ... |
| A | ... that divide **life** into *plant* and animal kingdom |
| A | ... many dangers to *plant* and animal **life** ... |
| A | mammals . Animal and *plant* **life** are delicately |
| A | beds too salty to support *plant* **life** . River ... |
| A | heavy seas, damage , and *plant* **life** growing on ... |
| A | ... ... |
| ? | ... vinyl chloride monomer *plant* , which is ... |
| ? | ... molecules found in *plant* and animal tissue |
| ? | ... Nissan car and truck *plant* in Japan is ... |
| ? | ... and Golgi apparatus of *plant* and animal cells ... |
| ? | ... union responses to *plant* closures . ... |
| ? | ... ... |

# More Training Examples

| | |
|---|---|
| ? | … … |
| ? | … cell types found in the *plant* kingdom are … |
| ? | … company said the *plant* is still operating … |
| ? | … Although thousands of *plant* and animal species |
| ? | … animal rather than *plant* tissues can be … |
| ? | … computer disk drive *plant* located in … |
| B | … … |
| B | automated **manufacturing** *plant* in Fremont … |
| B | … vast **manufacturing** *plant* and distribution … |
| B | chemical **manufacturing** *plant* , producing viscose |
| B | … keep a **manufacturing** *plant* profitable without |
| B | computer **manufacturing** *plant* and adjacent … |
| B | discovered at a St. Louis *plant* **manufacturing** |
| B | … copper **manufacturing** *plant* found that they |
| B | copper wire **manufacturing** *plant* , for example … |
| B | 's cement **manufacturing** *plant* in Alpena … |
| B | polystyrene **manufacturing** *plant* at its Dow … |
| B | company **manufacturing** *plant* is in Orlando … |

# Sample Initial State




- All occurrences of the target word are identified
- A small training set of seed data is tagged with word sense

# Step 3a

- Train the supervised classification algorithm on the SENSE-A/SENSE-B seed sets.

**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 |
| | ... | |

# Step 3b

- Apply the decision-list classifier to the entire sample set.
- Take those members in the residual that are tagged as SENSE-A or SENSE-B with probability above a certain threshold and add those examples to the growing seed sets.

- What happens? $\Rightarrow$ The new additions contain newly-learned collocations that are reliably indicative of the previously-trained seed sets.

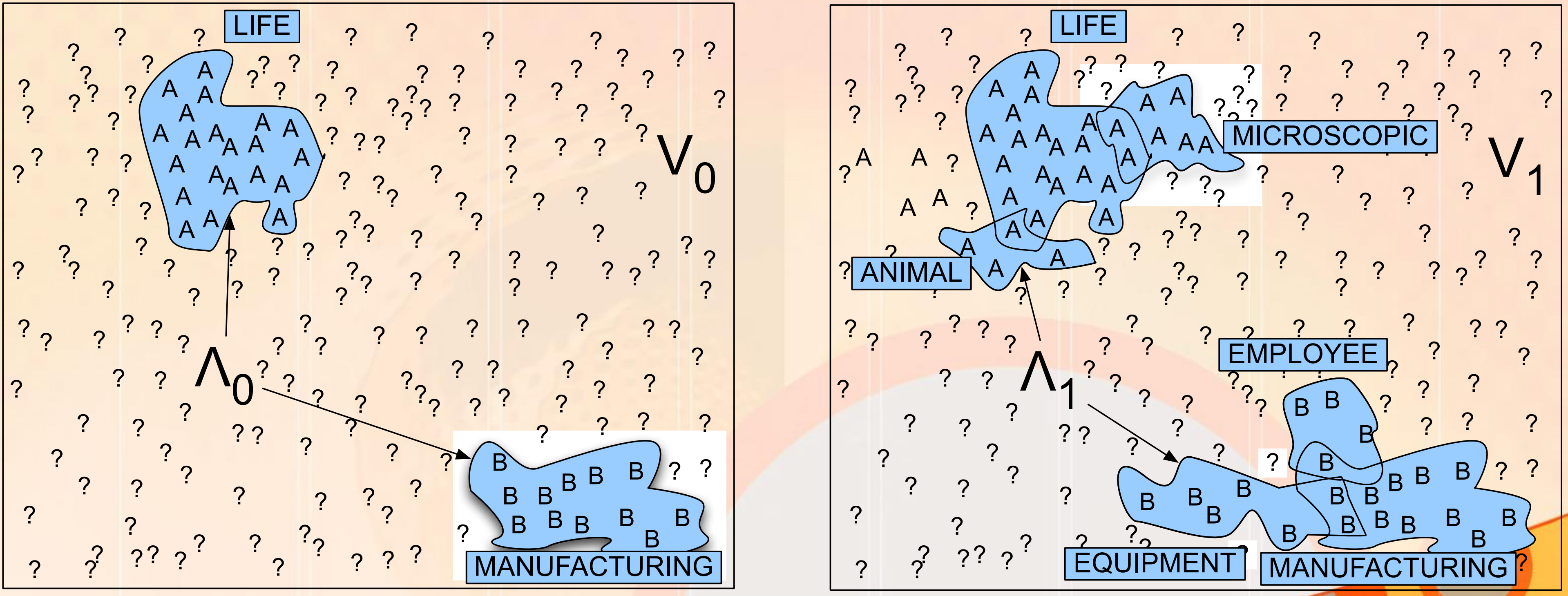
# Step 3c

- Optionally, use the one-sense-per-discourse heuristic to both filter and augment the addition of collocations.

- If several instances of a polysemous word in a discourse have already been assigned SENSE-A

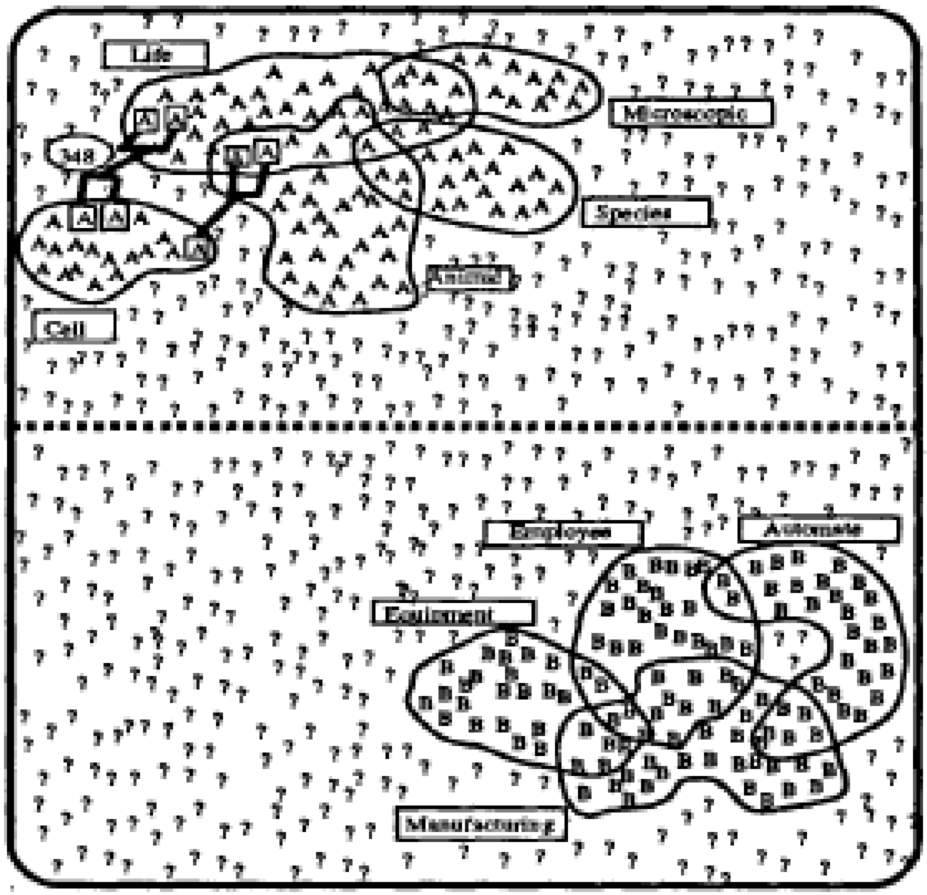
→ extend this tag to all examples in the discourse, conditional on the relative numbers and the probabilities associated with the tagged examples.

# Stages in the Yarowsky bootstrapping algorithm for the word "plant"




(a) (b)

# Sample Intermediate State




*Seed set grows and residual set shrinks ….*

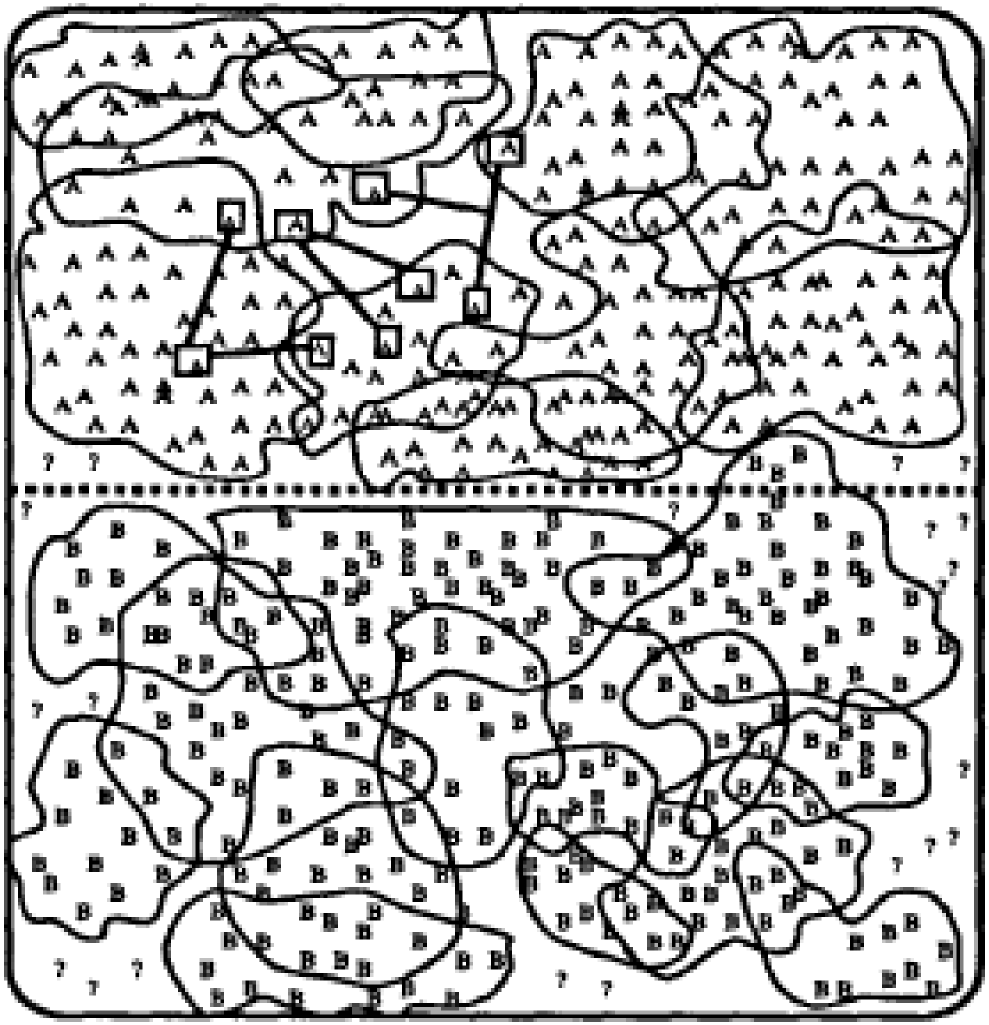
# Step 3d

- Repeat Step 3 iteratively. The training set (seeds + newly added examples) will tend to grow. The residual will tend to shrink.

- **Step 4:** STOP when the training parameters are held constant, the algorithm will converge on a stable residual set.

- **Step 5:** The classification procedure from the final supervised trained step can be applied to new data.

# Later



*Convergence: Stop when residual set stabilizes*

# Summary

- *Word Sense Disambiguation: choosing correct sense in context*
- *Applications: MT, QA, etc.*
- *Three classes of Methods*
  - *Supervised Machine Learning: Naive Bayes/NN classifier*
  - *Thesaurus/Dictionary Methods + Augmented Gloss Neural Architecture (has code!!!)*
  - *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*