Word Sense Disambiguation – WSD

B. Senthil Kumar

Asst. Professor

Sense & Disambiguation

Supervised ML – Naive Bayes

Dictionary based – Lesk Algorithm

Bootstrapping – Yarowsky Algorithm

Disambiguation

- Many words have several meanings or senses.
- Thus there is an ambiguity about how they are to be interpreted.
- The task of *disambiguation* is to determine which of the senses of an ambiguous word is invoked in a particular use of the word.
- This is done by looking at the context of the word's use.
- A word is assumed to have a finite number of discrete senses, often given by a *dictionary, thesaurus, or other reference source*.
- The task of the program is to make a forced choice between these senses of an ambiguous word, based on the context of use.

Supervised WSD

- Extract features from the text and then train a classifier to assign the correct sense given these features.
- A feature vector consisting of *numeric values* is used to encode this linguistic information as an input to ML algorithms.
- Two classes of features:
  - collocational features
  - bag-of-words features

Collocational features

- A collocation is a word or phrase in a position-specific relationship to a target word (i.e., exactly one word to the right, or exactly 4 words to the left, and so on). Thus
- Collocational features encode information about specific *positions* located to the left or right of the target word.
- Example:

  exactly 1 word to the right, or exactly 4 words to the left, and so on
- Typical features for these context words include the word itself, the root form of the word, and the word's pos.

Collocational features

- 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.
- Disambiguate the sense of ***bass ?***
- A collocational feature-vector, extracted from a window of two words along with its pos.

[w_{i-2} , POS_{i-2} , w_{i-1} , POS_{i-1} , w_{i+1} , POS_{i+1} , w_{i+2} , POS_{i+2}]

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

Bag-of-words features

- A bag-of-words means an unordered set of words, ignoring their exact position.
- bag-of-words approach:
  - The context of a target word by a vector of features.
  - Each binary feature indicating whether a vocabulary word w does or doesn't occur in the context.
- A bag-of-words vector consisting of the 12 most frequent content words from a collection of *bass* sentences drawn from the WSJ: [*fishing, big, sound, player, fly, rod, pound, double, runs, playing, guitar, band*]

Bag-of-words features

- 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.

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

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

forms the basis for *vector space model*

Naïve Bayes Approach

Choosing the best sense *s* out of the set of possible senses S for a feature vector f amounts to choosing the most probable sense given that vector.

$$\hat{s} = \text{argmax } P(s|f)$$

$$= \text{argmax } P(f|s)\,P(s)\,/\,P(f)$$

$$= \text{argmax } P(f|s)\,P(s)$$

P (f)-const. for all senses

$$= \text{argmax } P(s) \prod P(f|s)$$

where

P(s) – prior probability of each sense

P(f | s) – individual feature probability

Naïve Bayes Approach

Courtesy:
Dan Jurafsky

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

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

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

Choosing a class:

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

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

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

Dictionary-based

- Lesk algorithm – choose the sense whose dictionary gloss or definition shares the most words with the target word's neighborhood.

function SIMPLIFIED LESK(*word, sentence*) **returns** best sense of *word*

best-sense ← most frequent sense for *word*
max-overlap ← 0
context ← set of words in *sentence*
for each *sense* **in** senses of *word* **do**
  signature ← set of words in the gloss and examples of *sense*
  overlap ← COMPUTEOVERLAP(*signature*, *context*)
  if *overlap* > *max-overlap* **then**
    max-overlap ← *overlap*
    best-sense ← *sense*
end
return(*best-sense*)

Lesk Algorithm

- 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"

- bank[1] has two words overlapping: *deposits* and *mortgage*

  bank[2] has zero, so sense bank[1] is chosen.

Lesk Algorithm

- Limitations:
- The dictionary entries for the target words are short, and may not provide enough chance of overlap with the context.
- Solution: apply a weight to each overlapping word. The weight is the inverse document frequency or IDF – **Corpus Lesk**.
- Used as baseline in SENSEVAL competitions.

WSD: Bootstrapping

- **Yarowsky's Algorithm:**
- Given a small seed-set Λ0 of labeled instances of each sense, and a much larger unlabeled corpus V0.
- The algorithm first trains an initial decision-list classifier on the seed-set Λ0.
- Use this classifier to label the unlabeled corpus V0.
- Select the examples in V0 that it is most confident about, removes them, and adds them to the training set (call it now Λ1).
- Train a new decision list classifier on Λ1.
- Iterate by applying the classifier to the unlabeled set V1 extracting a new training set Λ2 and so on.

Yarowsky Algorithm

- With each iteration, the training corpus grows and the untagged corpus shrinks.
- The process is repeated until some sufficiently low error-rate on the training set, or until no further examples from the untagged corpus are above threshold.
- Key to bootstrapping – accurate initial set of seeds.
- One way to generate the initial seeds is to hand-label a small set of examples or use a heuristic to select accurate seeds.
- Yarowsky (1995) used the **One Sense per Collocation** heuristic, which relies on, certain words or phrases strongly associated with the target senses tend not to occur with the other sense.

Yarowsky Algorithm

Seed sentence:

bass[1] – collocate (fish, bass)

bass[2] – collocate (play, bass)

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 play**er stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.

When the New Jersey Jazz Society, in a fund-raiser for the American Jazz Hall of Fame, honors this historic night next Saturday, Harry Goodman, Mr. Goodman's brother and **bass play**er at the original concert, will be in the audience with other family members.

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 **fish**ermen decided the striped **bass** in Lake Mead were too skinny.

Though still a far cry from the lake's record 52-pound **bass** of a decade ago, "you could fillet these **fish** again, and that made people very, very happy," Mr. Paulson says.

Figure 20.5 Samples of *bass* sentences extracted from the WSJ using the simple correlates *play* and *fish*.

Yarowsky Algorithm

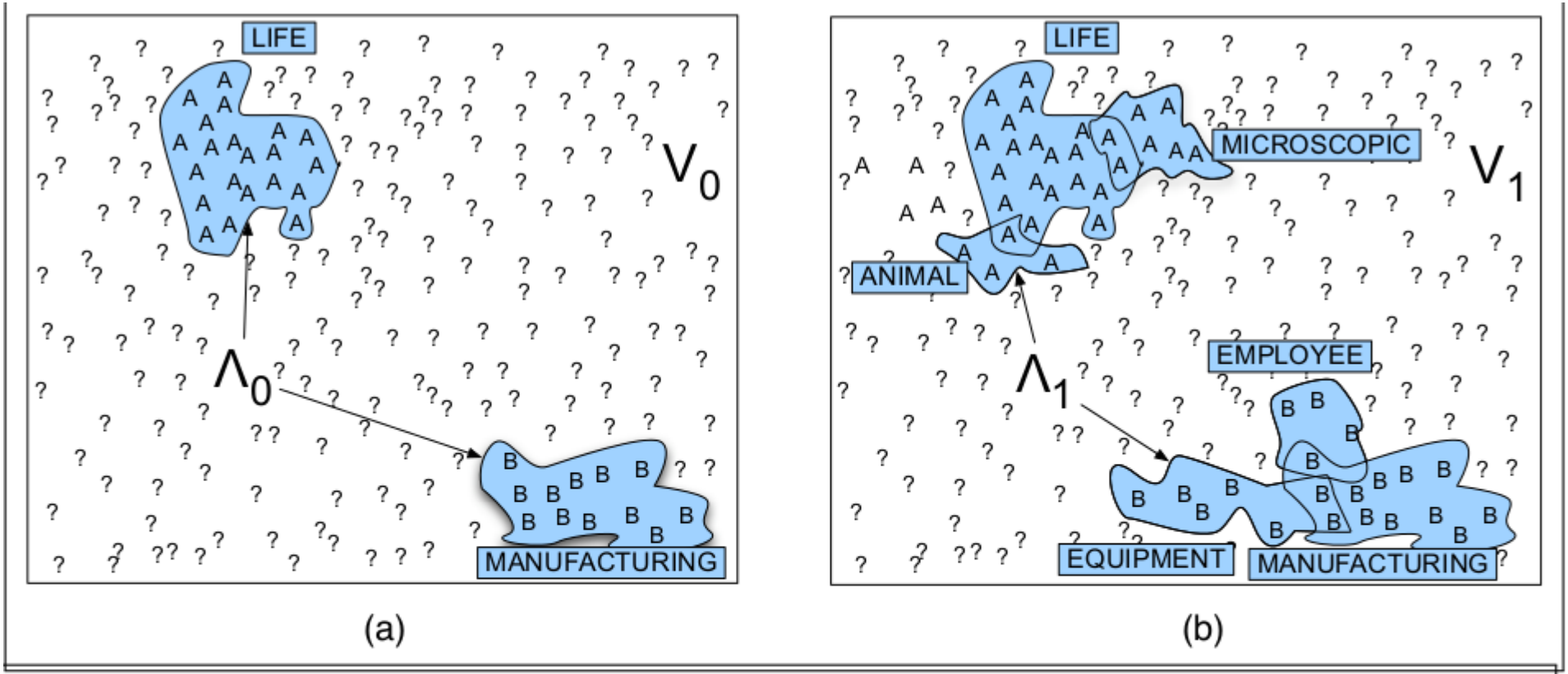

Figure 20.4 The Yarowsky algorithm disambiguating 'plant' at two stages; '?' indicates an unlabeled observation, A and B are observations labeled as SENSE-A or SENSE-B. 'LIFE' indicates observations occur with collocate "life". The initial stage (a) shows only seed sentences Λ_0 labeled by collocates ('life' and 'manufacturing'). An intermediate stage is shown in (b) where more collocates have been discovered ('equipment', 'microscopic', etc) and more instances in V_0 have been moved into Λ_1, leaving a smaller unlabeled set V_1. Figure adapted from Yarowsky (1995).

References

- *Speech and Language Processing*, Daniel Jurafsky, Martin, Pearson, 2006.
- *Natural Language Processing and Information Retrieval*, Tanveer Siddique, Tiwari, Oxford Press