

CMPT-413

Computational Linguistics

Anoop Sarkar
`http://www.cs.sfu.ca/~anoop`

April 7, 2008

Lexical Semantics

- ▶ So far, we have listed **words** in our **lexicon** or **vocabulary** assuming a single meaning per word:
Consider n -grams $P(w_i | w_{i-2}, w_{i-1}) = P(\text{Bank} | \text{on}, \text{Commerce})$ or
prepositional phrase attachment *if $p=\text{on}$ and $n2=\text{bank}$ then change N to V*
- ▶ Consider ... *withdraw twenty dollars on the bank* (correct = V)
vs.
... *withdraw the troops on the bank* (correct = N)
- ▶ The same word *bank* means two different things but we cannot distinguish between them using the traditional definition of word.

Lexical Semantics

- ▶ To deal with this issue, we combine the *spelling* or *pronunciation* of a word and the *meaning*.
In the *lexicon* we now store **lexemes** instead of words. A lexeme pairs a particular spelling or pronunciation with a particular meaning.
- ▶ The meaning part of a lexeme is called a **sense**. For CL, our interest is in relations between lexemes or disambiguating different senses of a word.
word: bank → lexeme: **bank**¹ OR word: bank → lexeme: **bank**²
- ▶ Note that meanings are often not definitions, but often are simple listings of compatible lexemes.
cf. dictionary defns: *red*, *n.* the color of blood or ruby; *blood*, *n.* red liquid circulating in animals

Homonyms

- ▶ Homonyms: words that have the same form but different meanings
 1. *Instead, the chemical plant was found in violation of several environmental laws*
 2. *Stanley formed an expedition to find a rare plant found along the Amazon river*
- ▶ Same orthographic form: *plant* but two senses: **plant**¹ and **plant**²

Homonyms

- ▶ Text vs. speech: fly-casting for *bass* vs. rhythmic *bass* chords
These cases are homonyms in text, but not in speech.
Referred to as **homographs**
- ▶ Speech vs. text: *would* vs. *wood*
These cases are not homonyms in text, but easily confused in speech. Referred to as **homophones**
- ▶ Note that this problem in some cases can be solved using *part of speech tagging*
Can you think of a case which cannot be solved using POS tagging?

Applications

- ▶ Spelling correction: homophones: *weather* vs. *whether*
- ▶ Speech recognition: homophones: *to*, *two*, *too*. Also homonyms (see *n*-gram e.g.)
- ▶ Text to speech: homographs: *bass* vs. *bass*
- ▶ Information retrieval: homonyms: *latex*

Polysemy

- ▶ Consider the homonym: *bank* → commercial **bank**¹ vs. river **bank**²
- ▶ Now consider
 1. *A PCFG can be trained using derivation trees from a tree bank annotated by human experts*
- ▶ Is this a new sense of *bank*?

Polysemy

- ▶ Senses can be derived from a particular lexeme. This process is known as **polysemy**
In previous case we would say that the use of *bank* is a sense derived from commercial **bank**¹
- ▶ In some cases, splitting into different lexemes has other supporting evidence: **bank**¹ has Italian origin vs. **bank**² has Scandinavian origin
 1. *A PCFG can be trained using a bank of derivation trees called a tree-bank annotated by human experts*
- ▶ How can we tell between homonyms and polysemous uses of a word?

Word sense and conjunction: zeugma

- ▶ Consider the case for a verb like *serve*
 1. *Does United serve breakfast?*
 2. *Does United serve Philadelphia?*
 3. *Does United serve breakfast and dinner?*
 4. *#Does United serve breakfast and Philadelphia?*

Word Sense Disambiguation

- ▶ Consider a noun like *bank*
 1. *How many senses does it have?*
 2. *How are these senses related?*
 3. *How can they be reliably distinguished?*
- ▶ For NLP software, among these three questions, typically at runtime we need to automatically find the answer to the last question: given a word in context, map it to the correct lexeme: **word-sense disambiguation**

Word Sense Disambiguation: data

new_JJ Ukrainian_JJ	plant _NN operators_NNS to_TO re
are_VBP leaving_VBG the_DT	plant s_NNS in_IN Ukraine_NNP an
safety_NN procedures_NNS at_IN	plant s_NNS in_IN both_DT countr
the_DT Orange_NNP County_NNP	plant _NN ..
three_CD missile_NN	plant s_NNS in_IN southern_JJ Ca
the_DT whole_JJ Chernobyl_NNP	plant _NN in_IN 1991_CD ,_, five
a_DT hill_NN ,_, gardeners_NNS	plant _NN begonias_NNS ,_, makin
200_CD million_CD printing_NN	plant _NN in_IN Brooklyn_NNP ,_,
incompletely_JJ oxidated_JJ	plant _NN and_CC animal_NN sedim
you_PRP eat_VBP a_DT	plant _NN ..
return_NN for_IN a_DT new_JJ	plant _NN near_IN Tuscaloosa_NNP
could_MD finance_VB	plant _NN construction_NN with_I
return_NN for_IN a_DT new_JJ	plant _NN near_IN Tuscaloosa_NNP

- ▶ Keyword in context listing for *plant* as a noun.
- ▶ Two senses of *plant*: *living* or *factory*.
- ▶ Part of speech tagging is essential: ignore *plant* as a verb.

Word Sense Disambiguation: features

- ▶ Consider the input:

that_WDT also_RB is_VBZ a_DT preserve_VB for_IN
plant_NN ,_, animal_NN and_CC bird_NN life_NN

- ▶ Features that can help us determine the word sense:

'W+1=,_,',
'W-1=for_IN',
'W-2,W-1=preserve_VB,for_IN',
'W+1,W+2=,_,,animal_NN',
'W-1,W+1=for_IN,,_',
'W+-K=that_WDT',
'W+-K=also_RB',
'W+-K=is_VBZ',
'W+-K=a_DT',
'W+-K=preserve_VB',
'W+-K=animal_NN',
'W+-K=and_CC',
'W+-K=bird_NN',
'W+-K=life_NN'

Word Sense Disambiguation: methods

- ▶ Several options for creating a system that does word-sense disambiguation
- ▶ Supervised learning:
 - ▶ Label training data.
 - ▶ Learn a classifier $\text{Pr}(\textit{sense} \mid \textit{features})$
- ▶ Unsupervised learning
 - ▶ Cluster sentences into two (or more) classes.
 - ▶ Label each class manually with the sense information.
- ▶ Bootstrapping
 - ▶ Use *seed rules* to identify some examples of almost sure cases of each sense.
 - ▶ Train a classifier on this data.
 - ▶ Use classifier to identify the sense for new examples, and iterate.

Word Sense Disambiguation: Decision Lists

- ▶ A *Decision List* is a simple classifier that is effective for word-sense disambiguation
- ▶ For each feature, we get an estimate for the probability of the word sense
- ▶ For example, consider factory sense (TECH) or living sense (BIO) for the word plant:
 - ▶ Consider the feature 'W+1=life'
 - ▶ We might get the following counts from training data:

$$\text{Count}(\text{TECH}, \text{'W+1=life'}) = 1$$

$$\text{Count}(\text{BIO}, \text{'W+1=life'}) = 100$$

- ▶ Using these counts we derive an estimate for:

$$P(\text{BIO} \mid \text{'W+1=life'}) = \frac{100 + \alpha}{101 + 2\alpha}$$

- ▶ Interpret this probability as a rule: if *feature* is observed, label as *sense* with confidence $P(\text{sense} \mid \text{feature})$
- ▶ Set $\alpha = 0.1$ (smoothing is essential in the next step)

Word Sense Disambiguation: Decision Lists

- ▶ A *Decision List* is a list of such rules sorted by strength
- ▶ The strength of a rule is derived using the log odds of picking one sense over another:

$$\text{strength}(\text{feature}) = \text{abs} \left(\log \left(\frac{P(\text{sense 1} \mid \text{feature})}{P(\text{sense 2} \mid \text{feature})} \right) \right)$$

- ▶ For example,

strength	feature f	sense s	$P(s \mid f)$
5.6	'W-1=manufacturing_NN'	'TECH'	0.99
4.7	'W-1,W+1=manufacturing_NN,in_IN'	'TECH'	0.99
4.5	'W+-K=animal_NN'	'BIO'	0.99
4.5	'W+1=life_NN'	'BIO'	0.99
	⋮		

- ▶ To apply the decision list, use the strongest (first) rule that can be applied (the feature appears in the input).

Word Sense Disambiguation: Decision Lists

- ▶ Decision lists can be trained on labeled data
- ▶ Yarowsky (1994) applies decision lists to accent restoration in French and Spanish:

De-accented form	Accented form	Percent
cesse	cesse	53%
	cessé	47%
coute	coûte	53%
	coûté	47%
cote	côté	69%
	côte	28%
	cote	3%
	coté	< 1%

- ▶ Task is to convert the de-accented form to the appropriate accented form.
- ▶ Very similar to word-sense disambiguation. (labeled data is easily constructed)
- ▶ Useful for automatic generation of accents while typing.

Word Sense Disambiguation: Decision Lists

- ▶ Yarowsky (1995) describes a bootstrapping approach for WSD for the following words:

Word	Senses
plant	living/factory
tank	vehicle/container
poach	steal/boil
palm	tree/hand
axes	grind/tools
sake	benefit/drink
bass	fish/music
space	volume/outer
motion	legal/physical
crane	bird/machine

Word Sense Disambiguation: Bootstrapping

- ▶ Expert picks a few seed rules (they should be *strong* rules)

manufacturing plant \Rightarrow TECH

plant life \Rightarrow BIO

- ▶ Apply seed rules on the unlabeled data.
- ▶ Bootstrapping Algorithm (Yarowsky 1995)
 - ▶ Train a decision list using the (partially labeled) data.
 - ▶ Use the original unlabeled data, and apply the decision list classifier only if the probability of prediction is greater than some threshold, say 0.97
 - ▶ Re-train a new decision list, and repeat this procedure until the labels for the data do not change.

Word Sense Disambiguation: Bootstrapping

- ▶ Another useful property: “One Sense Per Discourse”.
- ▶ Yarowsky (1995) observes that if the same word occurs multiple times in a document, then it is very likely to have the same word sense.
- ▶ After the decision list is applied, this “one sense per discourse” property is applied to label all the target words in a document.
- ▶ With just two seed rules, Yarowsky (1995) obtains 90.6% accuracy (average across all the words in previous slide).
- ▶ With better seed rules, accuracy goes up to 95.5% accuracy.

Word Sense Disambiguation: Bootstrapping

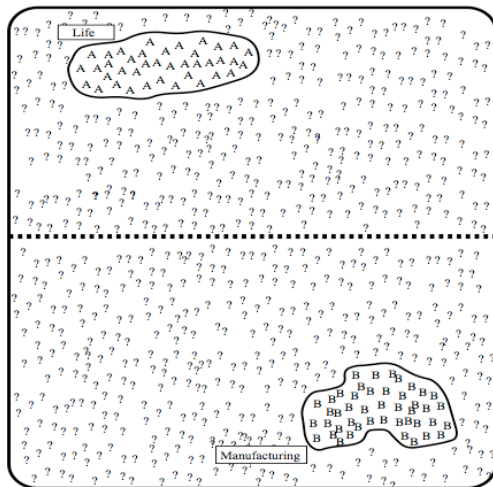


Figure 1: Sample Initial State

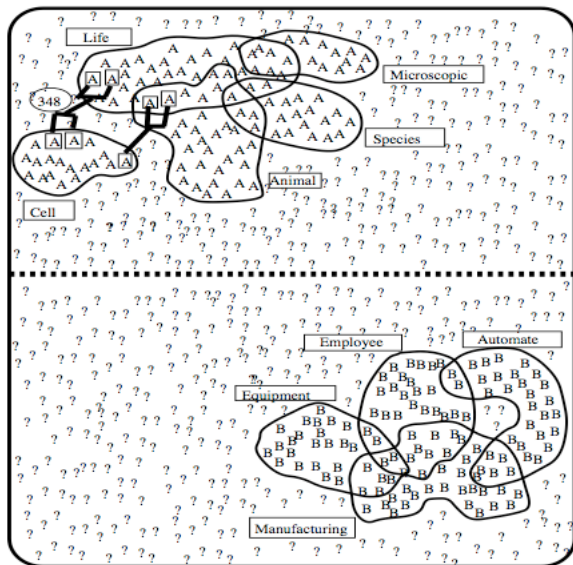
A = SENSE-A training example

B = SENSE-B training example

? = currently unclassified training example

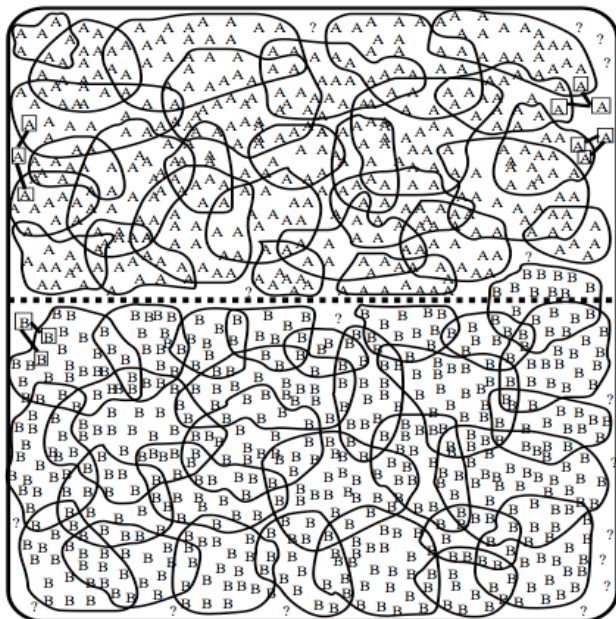
Life = Set of training examples containing the collocation “life”.

Word Sense Disambiguation: Bootstrapping



“One Sense Per Discourse” applied to Document 348

Word Sense Disambiguation: Bootstrapping



Synonyms

- ▶ Synonyms: Different lexemes with the same meaning
 1. *How big/large is that plane?*
 2. *Would I be flying on a big/large or small plane?*
- ▶ Synonyms clash with polysemous meanings
 1. *Seema is my big sister*
 2. *#Seema is my large sister*

WordNet

- ▶ WordNet is an electronic database of word relationships, handcrafted from scratch by researchers at Princeton University (George Miller, Christine Fellbaum, et al.)
- ▶ WordNet contains 3 databases: for verbs, nouns and one for adjectives and adverbs

Category	Unique Forms	Number of Senses
Noun	94474	116317
Verb	10319	22066
Adjective	20170	29881
Adverb	4546	5677

WordNet

- ▶ Ask the question: how many senses per noun or verb? The distribution of senses follows Zipf's (2nd) Law.
- ▶ WordNet provides multiple lexeme entries for each word and for each part of speech,
e.g. *plant* as noun has 3 senses; *plant* as verb has 2 senses
- ▶ WordNet also provides *domain-independent* lexical relations such as IS-A, HasMember, MemberOf, ...

WordNet: noun relations

Relation	Definition	Example
Hypernym	this is a kind of	<i>breakfast</i> → <i>meal</i>
Hyponym	this has a specific instance	<i>meal</i> → <i>lunch</i>
Has-Member	this has a member	<i>faculty</i> → <i>professor</i>
Member-Of	this is member of a group	<i>copilot</i> → <i>crew</i>
Has-Part	this has a part	<i>table</i> → <i>leg</i>
Part-Of	this is part of	<i>course</i> → <i>meal</i>
Antonym	this is an opposite of	<i>leader</i> → <i>follower</i>

WordNet: verb relations

Relation	Definition	Example
Hypernym	this event is a kind of	<i>fly</i> → <i>travel</i>
Tropynym	this event has a subtype	<i>walk</i> → <i>stroll</i>
Entails	this event entails	<i>snore</i> → <i>sleep</i>
Antonym	this event is opposite of	<i>increase</i> → <i>decrease</i>

WordNet: example from ver1.7.1

Sense1: Canada

⇒North American country,North American nation

⇒country, state, land

⇒administrative district,administrative division,territorial division

⇒district, territory

⇒region

⇒location

⇒entity, physical thing

WordNet: example from ver1.7.1

Sense 3: Vancouver

- ⇒city, metropolis, urban center
 - ⇒municipality
 - ⇒urban area
 - ⇒geographical area
 - ⇒region
 - ⇒location
 - ⇒entity, physical thing
 - ⇒administrative district, territorial division
 - ⇒district, territory
 - ⇒region
 - ⇒location
 - ⇒entity, physical thing
- ⇒port
 - ⇒geographic point
 - ⇒point
 - ⇒location
 - ⇒entity, physical thing

WordNet

- ▶ A **synset** in WordNet is a list of synonyms (interchangeable words)
- ▶ { chump, fish, fool, gull, mark, patsy, fall guy, sucker, schlemiel, shlemiel, soft touch, mug }
- ▶ How can we use this information like synsets, hypernyms, etc. from WordNet to benefit NLP applications?
- ▶ Consider one example: PP attachment, words plus word classes extracted from the hypernym hierarchy increase accuracy from 84% to 88% (Stetina and Nagao, 1998)

- ▶ Another example of WordNet used in NLP applications:
selectional restrictions
- ▶ We have considered subcategorization:
VP-with-NP-complement → *V(eat) NP* “eat six bowls of rice ”
But not selectional restrictions of the verb itself: “ *eat tomorrow* ”
Consider *what do you want to eat tomorrow*
- ▶ We can use the **synset** { *food*, *nutrient* } to describe the NP argument of *eat* – then the 60K lexemes under these nodes in the WordNet hierarchy will be acceptable.
(however, what about “ *eat my shorts* ”)
→ several other applications have been explored