Word Sense Disambiguation

(Following slides are modified from Prof. Claire Cardie's slides.)

Quick Preliminaries

- Part-of-speech (POS)
- Function words / Content words / Stop words

Part of Speech (POS)

- Noun (person, place or thing)
  - Singular (NN): dog, fork
  - Plural (NNS): dogs, forks
  - Proper (NNP, NNPS): John, Springfields
  - Personal pronoun (PRP): I, you, he, she, it
  - Wh-pronoun (WP): who, what
- Verb (actions and processes)
  - Base, infinitive (VB): eat
  - Past tense (VBD): ate
  - Gerund (VBG): eating
  - Past participle (VBN): eaten
  - Non 3rd person singular present tense (VBP): eat
  - 3rd person singular present tense: (VBZ): eats
  - Modal (MD): should, can
  - **To (TO): to (to eat)**

Part of Speech (POS)

- Adjective (modify nouns)
  - Basic (JJ): red, tall
  - Comparative (JJR): redder, taller
  - Superlative (JJS): reddest, tallest
- Adverb (modify verbs)
  - Basic (RB): quickly
  - Comparative (RBR): quicker
  - Superlative (RBS): quickest
- Preposition (IN): on, in, by, to, with
- Determiner:
  - Basic (DT) a, an, the
  - WH-determiner (WDT): which, that
- Coordinating Conjunction (CC): and, but, or,
- Particle (RP): off (took off), up (put up)

Penn Tree Tagset

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	“to”	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential ‘there’	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	$
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	“	left quote	‘ or “
POS	possessive ending	*’s*	”	right quote	’ or ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one’s*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Function Words / Content Words

- **Function words (closed class words)**
  - words that have little lexical meaning
  - express grammatical relationships with other words
  - Prepositions (in, of, etc), pronouns (she, we, etc), auxiliary verbs (would, could, etc), articles (a, the, an), conjunctions (and, or, etc)
- **Content words (open class words)**
  - Nouns, verbs, adjectives, adverbs etc
  - Easy to invent a new word (e.g. “google” as a noun or a verb)
- **Stop words**
  - Similar to function words, but may include some content words that carry little meaning with respect to a specific NLP application

(Machine Learning) Approaches for WSD

- Dictionary-based approaches
  - Simplified Lesk
  - Corpus Lesk
- Supervised-learning approaches
  - Naïve Bayes
  - Decision List
  - K-nearest neighbor (KNN)
- Semi-supervised-learning approaches
  - Yarowsky's Bootstrapping approach
- Unsupervised-learning approaches
  - Clustering

Dictionary-based approaches

- Rely on machine readable dictionaries
- Initial implementation of this kind of approach is due to Michael Lesk (1986)
- "Lesk algorithm"
  - Given a word *W* to be disambiguated in context C
    - Retrieve all of the sense definitions, *S*, for *W* from the MRD
    - Compare each *s* in *S* to the dictionary definitions D of all the remaining words c in the context C
    - Select the sense *s* with the most overlap with D (the definitions of the context words C)

Example

- Word: *cone*
- Context: *pine cone*
- Sense definitions

  pine 1 kind of evergreen tree with needle-shaped leaves
  2 waste away through sorrow or illness

  cone 1 solid body which narrows to a point
  2 something of this shape whether solid or hollow
  3 fruit of certain evergreen trees

- Accuracy of 50-70% on short samples of text from *Pride and Prejudice* and an AP newswire article.

Simplified Lesk Algorithm

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)
```

Pros & Cons?

- Pros
  - Simple
  - Does not require (human-annotated) training data

- Cons
  - Very sensitive to the definition of words
  - Words used in definition might not overlap with the context.
  - Even if there is a human annotated training data, it does not learn from the data.

Variations of Lesk

- **Original Lesk (Lesk 1986):**
  - **signature**(sense) = signature of content words in context/gloss/example
  - Problem with Lesk: overlap is often zero.
- **Corpus Lesk (With a labeled training corpus)**
  - Use sentences in corpus to compute signature of senses
  - Compute weighted overlap:
  - Weigh each word by **its inverse document frequency (IDF)** score:
  - IDF(word) = log(#AllDocs / #DocsContainingWord)
  - Here, document = context/gloss/example sentences

(Machine Learning) Approaches for WSD

- Dictionary-based approaches
  - Simplified Lesk
  - Corpus Lesk
- Supervised-learning approaches
  - Naïve Bayes
  - Decision List
  - K-nearest neighbor (KNN)
- Semi-supervised-learning approaches
  - Yarowsky’s Bootstrapping approach
- Unsupervised-learning approaches
  - Clustering

Machine Learning framework

Examples of task

(features + class)

description of context

correct word sense

ML Algorithm

Novel example

(features)

Classifier

(program)

class

learn one such classifier
for each lexeme to be
disambiguated

Running example

*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.*

1 Fish sense

2 Musical sense

3 …

Feature vector representation

- **target:** the word to be disambiguated
- **context** : portion of the surrounding text
  - Select a “window” size
  - Tagged with part-of-speech information
  - Stemming or morphological processing
  - Possibly some partial parsing
- Convert the context (and target) into a set of features
  - Attribute-value pairs
    - Numeric, boolean, categorical, …

Collocational features

- Encode information about the lexical inhabitants of *specific* positions located to the left or right of the target word.
  - E.g. the word, its root form, its part-of-speech
  - *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.*

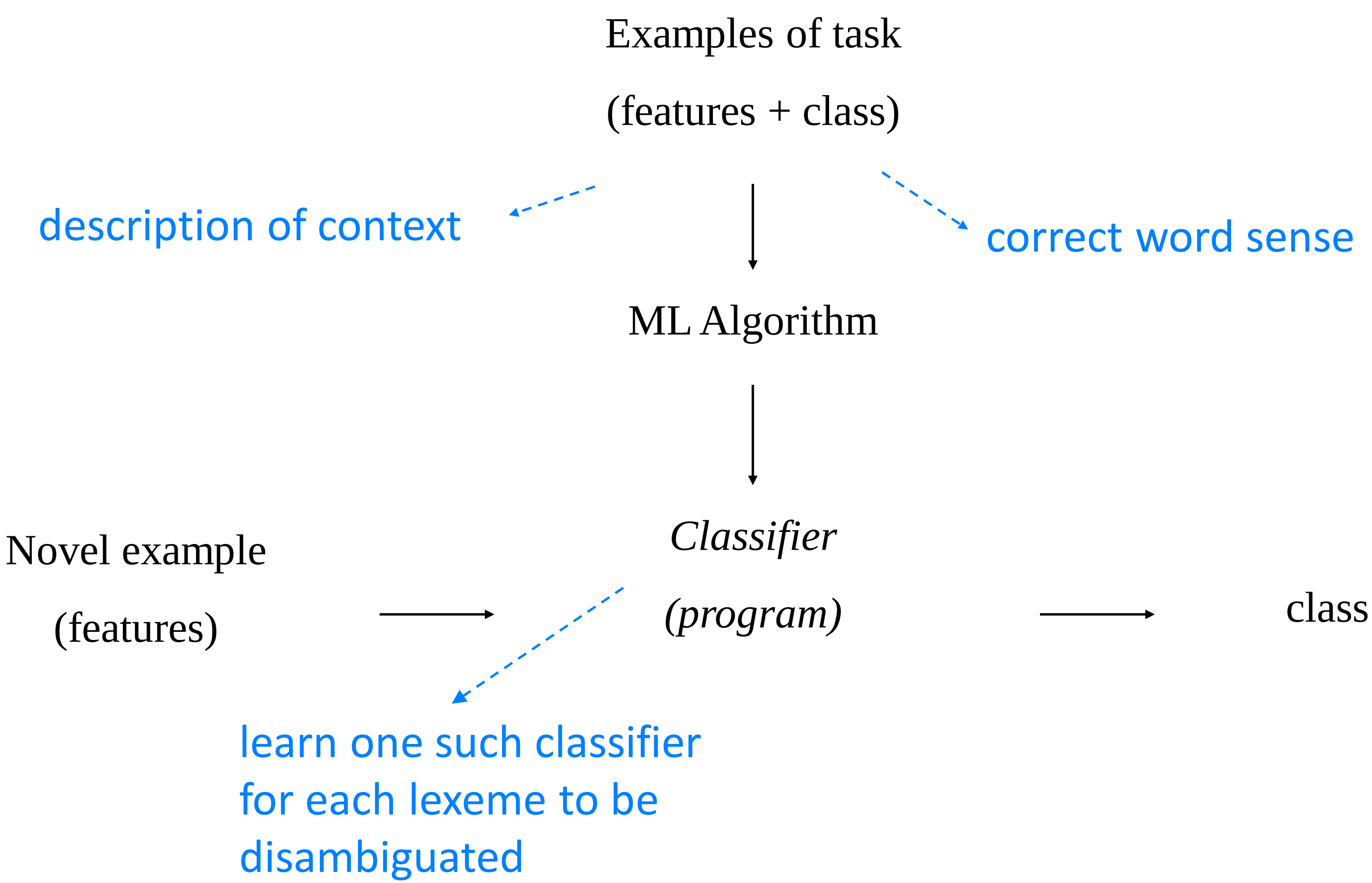
pre2-word	pre2-pos	pre1-word	pre1-pos	fol1-word	fol1-pos	fol2-word	fol2-pos
guitar	NN	and	CJC	player	NN	stand	VVB

Co-occurrence features

- Encodes information about neighboring words, ignoring exact positions.
  - Select a small number of frequently used content words for use as features
    - 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*
    - Co-occurrence vector (window of size 10)
  - **Attributes**: the words themselves (or their roots)
  - **Values**: number of times the word occurs in a region surrounding the target word

fishing?	big?	sound?	player?	fly?	rod?	pound?	double?	…	guitar?	band?
0	0	0	1	0	0	0	0		1	0

Inductive ML framework



Naïve Bayes classifiers for WSD

- Assumption: choosing the best sense for an input vector amounts to choosing the most probable sense for that vector

$$\hat{s} = \underset{s \in S}{\arg\max}\ P(s \mid V)$$

  - S denotes the set of senses
  - V is the context vector
- Apply Bayes rule:

$$\hat{s} = \underset{s \in S}{\arg\max}\ \frac{P(V \mid s)P(s)}{P(V)}$$

Naïve Bayes classifiers for WSD

- Estimate P(V|s):

$$P(V \mid s) \approx \prod_{j=1}^{\#\,feature-value\,pairs} P(v_j \mid s)$$

- P(*s*): proportion of each sense in the sense-tagged corpus

$$\hat{s} = \arg\max_{s \in S} \; P(s) \prod_{j=1}^{\#\,feature-value\,pairs} P(v_j \mid s)$$

(Machine Learning) Approaches for WSD

- Dictionary-based approaches
  - Simplified Lesk
  - Corpus Lesk
- Supervised-learning approaches
  - Naïve Bayes
  - Decision List
  - K-nearest neighbor (KNN)
- Semi-supervised-learning approaches
  - Yarowsky's Bootstrapping approach
- Unsupervised-learning approaches
  - Clustering

Decision list classifiers

- Decision lists: equivalent to simple case statements.
  - Classifier consists of a sequence of tests to be applied to each input example/vector; returns a word sense.
- Continue only until the first applicable test.
- Default test returns the majority sense.

Decision list example

- Binary decision: fish *bass* vs. musical *bass*

Rule		Sense
fish within window	⇒	**bass**^{1}
striped bass	⇒	**bass**^{1}
guitar within window	⇒	**bass**^{2}
bass player	⇒	**bass**^{2}
piano within window	⇒	**bass**^{2}
tenor within window	⇒	**bass**^{2}
sea bass	⇒	**bass**^{1}
play/V bass	⇒	**bass**^{2}
river within window	⇒	**bass**^{1}
violin within window	⇒	**bass**^{2}
salmon within window	⇒	**bass**^{1}
on bass	⇒	**bass**^{2}
bass are	⇒	**bass**^{1}

Learning decision lists

- Consists of *generating* and *ordering* individual tests based on the characteristics of the training data
- Generation: every feature-value pair constitutes a test
- Ordering: based on accuracy on the training set

$$abs\left(\log \frac{P(Sense_1 \mid f_i = v_j)}{P(Sense_2 \mid f_i = v_j)} \right)$$

- Associate the appropriate sense with each test

(Machine Learning) Approaches for WSD

- Dictionary-based approaches
  - Simplified Lesk
  - Corpus Lesk
- Supervised-learning approaches
  - Naïve Bayes
  - Decision List
  - K-nearest neighbor (KNN)
- Semi-supervised-learning approaches
  - Yarowsky's Bootstrapping approach
- Unsupervised-learning approaches
  - Clustering

Nearest-Neighbor Learning Algorithm

- Learning is just storing the representations of the training examples in *D*.
- Testing instance *x*:
  - Compute similarity between *x* and all examples in *D*.
  - Assign *x* the category of the most similar example in *D*.
- Does not explicitly compute a generalization or category prototypes.
- Also called:
  - Case-based
  - Memory-based
  - Lazy learning

K Nearest-Neighbor

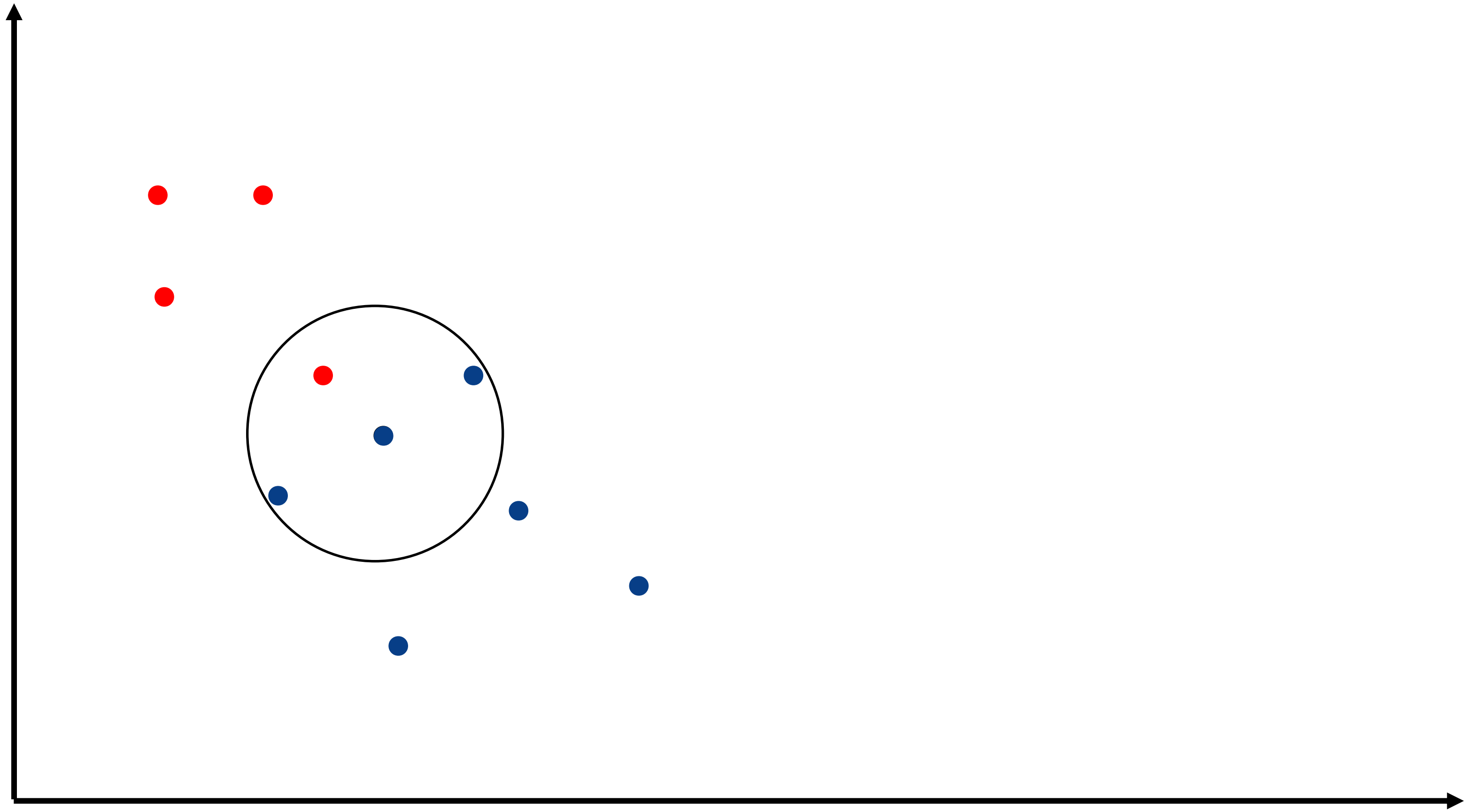
- Using only the closest example to determine categorization is subject to errors due to:
  - A single atypical example.
  - Noise (i.e. error) in the category label of a single training example.
- More robust alternative is to find the *k* most-similar examples and return the majority category of these *k* examples.
- Value of *k* is typically odd to avoid ties, 3 and 5 are most common.

Similarity Metrics

Nearest neighbor method depends on a similarity (or distance) metric.

1. Simplest for continuous *m*-dimensional instance space is *Euclidian distance*.
2. Simplest for *m*-dimensional binary instance space is *Hamming distance* (number of feature values that differ).
3. For text, *cosine similarity* of TF-IDF weighted vectors is typically most effective.

3 Nearest Neighbor Illustration
(Euclidian Distance)



(Machine Learning) Approaches for WSD

- Dictionary-based approaches
  - Simplified Lesk
  - Corpus Lesk

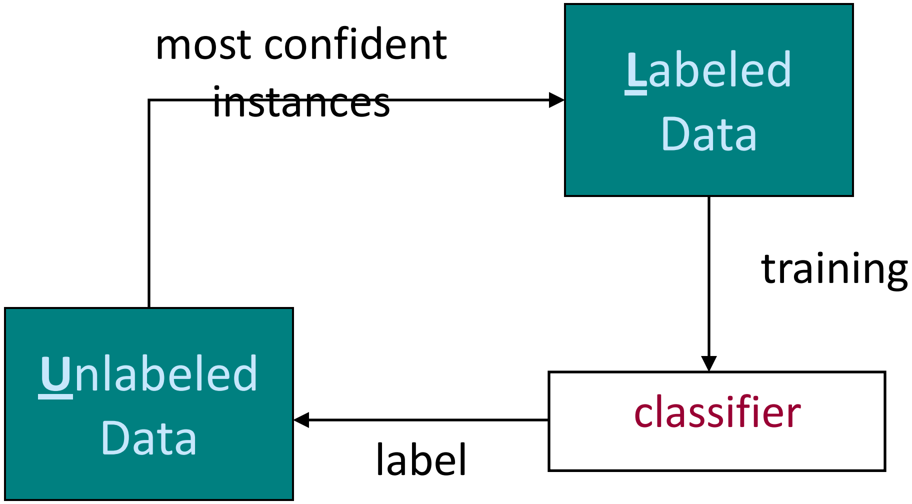
- Supervised-learning approaches
  - Naïve Bayes
  - Decision List
  - K-nearest neighbor (KNN)

- Semi-supervised-learning approaches
  - Yarowsky’s Bootstrapping approach

- Unsupervised-learning approaches
  - Clustering

Weakly supervised approaches

- Problem: Supervised methods require a large sense-tagged training set
- Bootstrapping approaches: Rely on a small number of labeled **seed** instances



Repeat:

1. train *classifier* on *L*
2. label *U* using *classifier*
3. add *g* of *classifier*'s best *x* to *L*

Generating initial seeds

- Hand label a small set of examples
  - Reasonable certainty that the seeds will be correct
  - Can choose prototypical examples
  - Reasonably easy to do
- **One sense per collocation** constraint (Yarowsky 1995)
  - Search for sentences containing words or phrases that are strongly associated with the target senses
    - Select *fish* as a reliable indicator of $bass_1$
    - Select *play* as a reliable indicator of $bass_2$
  - Or derive the collocations automatically from machine readable dictionary entries
  - Or select seeds automatically using collocational statistics (see Ch 6 of J&M)

One sense per collocation

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.

one sense per discourse constraint

Word	Senses	Accuracy	Applicability
plant	living/factory	99.8%	72.8%
tank	vehicle/container	99.6%	50.5%
poach	steal/boil	100.0%	44.4%
palm	tree/hand	99.8%	38.5%
axes	grid/tools	100.0%	35.5%
sake	benefit/drink	100.0%	33.7%
bass	fish/music	100.0%	58.8%
space	volume/outer	99.2%	67.7%
motion	legal/physical	99.9%	49.8%
crane	bird/machine	100.0%	49.1%
Average		99.8%	50.1%

How well does this constraint work on ~37,000 examples?

- Accuracy column shows --- when a word occurs more than once in a discourse, how often does it take on the majority sense of that discourse
- Applicability column shows --- how often does the word occur more than once in a particular discourse

Yarowsky’s bootstrapping approach

To learn disambiguation rules for a polysemous word:

1. [Find all instances of the word in the training corpus and save the contexts around each instance.]

2. [For each word sense, identify a small set of training examples representative of that sense. Now we have a few labeled examples for each sense.]

3. Build a classifier (e.g. decision list) by training a supervised learning algorithm with the labeled examples.

4. Apply the classifier to all the unlabeled examples. Find instances that are classified with probability > a threshold and add them to the set of labeled examples.

5. *Optional:* Use the one-sense-per-discourse constraint to augment the new examples.

6. Go to Step 3. Repeat until the unlabelled data is stable.

(Machine Learning) Approaches for WSD

- Dictionary-based approaches
  - Simplified Lesk
  - Corpus Lesk

- Supervised-learning approaches
  - Naïve Bayes
  - Decision List
  - K-nearest neighbor (KNN)

- Semi-supervised-learning approaches
  - Yarowsky’s Bootstrapping approach

- Unsupervised-learning approaches
  - Clustering

Unsupervised WSD

- Rely on **agglomerative clustering** to cluster feature-vector representations (without class/word-sense labels) according to a similarity metric
- Represent each cluster as the average of its constituent feature-vectors
- Label the cluster by hand with known word senses
- Unseen feature-encoded instances are classified by assigning the word sense of the most similar cluster
- Schuetze (1992, 1998) uses a (complex) clustering method for WSD
  - For coarse binary decisions, unsupervised techniques can achieve results approaching those of supervised and bootstrapping methods
  - In most cases approaching the 90% range
  - Tested on a small sample of words

Issues for evaluating clustering

- The correct senses of the instances used in the training data may not be known.
- The clusters are almost certainly heterogeneous w.r.t. the sense of the training instances contained within them.
- The number of clusters is almost always different from the number of senses of the target word being disambiguated.

Which is better???

- Dictionary-based approaches
  - Simplified Lesk
  - Corpus Lesk

- Supervised-learning approaches
  - Naïve Bayes
  - Decision List
  - K-nearest neighbor (KNN)

- Semi-supervised-learning approaches
  - Yarowsky’s Bootstrapping approach

- Unsupervised-learning approaches
  - Clustering

Word Sense Disambiguation
Evaluation

WSD Evaluation

- Corpora:
  - *line* corpus (Leacock et al. 1993)
  - Yarowsky's 1995 corpus
    - 12 words (plant, space, bass, …)
    - ~4000 instances of each
  - Ng and Lee (1996)
    - 121 nouns, 70 verbs (most frequently occurring/ambiguous); WordNet senses
    - 192,800 occurrences
  - SEMCOR (Landes et al. 1998)
    - Portion of the Brown corpus tagged with WordNet senses
  - SENSEVAL (Kilgarriff and Rosenzweig, 2000)
    - Annual performance evaluation conference
    - Provides an evaluation framework (Kilgarriff and Palmer, 2000)
- **Baseline**: most frequent sense

WSD Evaluation

- Metrics
  - Accuracy (% of correct prediction)
    - Nature of the senses used has a huge effect on the results
    - E.g. results using coarse distinctions cannot easily be compared to results based on finer-grained word senses
  - Partial credit
    - Worse to confuse musical sense of *bass* with a fish sense than with another musical sense
    - Exact-sense match → full credit
    - Select the correct broad sense → partial credit
    - Scheme depends on the organization of senses being used

Evaluation of WSD

- ***"In vitro" or "intrinsic":***
  - Corpus developed in which one or more ambiguous words are labeled with explicit sense tags according to some sense inventory.
  - Corpus used for training and testing WSD and evaluated using accuracy (percentage of labeled words correctly disambiguated).
    - Use most common sense selection as a baseline.
- ***"In vivo" or "extrinsic":***
  - Incorporate WSD system into some larger application system, such as machine translation, information retrieval, or question answering.
  - Evaluate relative contribution of different WSD methods by measuring performance impact on the overall system on final task (accuracy of MT, IR, or QA results).

N-Fold Cross-Validation

- Ideally, test and training sets are independent on each trial.
  - But this would require too much labeled data.
- Partition data into *N* equal-sized disjoint segments.
- Run *N* trials, each time using a different segment of the data for testing, and training on the remaining *N*–1 segments.
- This way, at least test-sets are independent.
- Report average classification accuracy over the *N* trials.
- Typically, *N* = 10.

Baselines

- You must compare the performance of your system against reasonable "**baselines**".
- Baselines are simple methods that give rough idea on the lower bound of performance.
- Sometimes it is surprisingly hard to beat baselines! More complex methods do not necessarily perform better than simple baselines.

- Possible baselines for WSD?
  - Random prediction
  - Most frequent sense (a must) -- not that trivial to beat
  - Lesk algorithm (optional)
  - Naïve Bayes (optional)

SENSEVAL-2 2001

- Three tasks
  - Lexical sample
  - All-words
  - Translation
- 12 languages
- Lexicon
  - SENSEVAL-1: from HECTOR corpus
  - SENSEVAL-2: from WordNet 1.7
- 93 systems from 34 teams

Lexical sample task

- Select a sample of words from the lexicon
- Systems must then tag instances of the sample words in short extracts of text
- SENSEVAL-1: 35 words
  - 700001 John Dos Passos wrote a poem that talked of `the <tag>bitter</> beat look, the scorn on the lip."
  - 700002 The beans almost double in size during roasting. Black beans are over roasted and will have a <tag>bitter</> flavour and insufficiently roasted beans are pale and give a colourless, tasteless drink.

Lexical sample task: SENSEVAL-1

Nouns		Verbs		Adjectives		Indeterminates	
-n	N	**-v**	N	**-a**	N	**-p**	N
accident	267	amaze	70	brilliant	229	band	302
behaviour	279	bet	177	deaf	122	bitter	373
bet	274	bother	209	floating	47	hurdle	323
disability	160	bury	201	generous	227	sanction	431
excess	186	calculate	217	giant	97	shake	356
float	75	consume	186	modest	270		
giant	118	derive	216	slight	218		
…	…	…	…	…	…		
TOTAL	2756	TOTAL	2501	TOTAL	1406	TOTAL	1785

All-words task

- Systems must tag almost all of the content words in a sample of running text
  - sense-tag all predicates, nouns that are heads of noun-phrase arguments to those predicates, and adjectives modifying those nouns
  - ~5,000 running words of text
  - ~2,000 sense-tagged words

Translation task

- SENSEVAL-2 task
- Only for Japanese
- word sense is defined according to translation distinction
  - if the head word is translated differently in the given expressional context, then it is treated as constituting a different sense
- word sense disambiguation involves selecting the appropriate English word/phrase/sentence equivalent for a Japanese word

SENSEVAL-2 results

Language	Task	No. of submissions	No. of teams	IAA	Baseline	Best system
Czech	AW	1	1	-	-	.94
Basque	LS	3	2	.75	.65	.76
Estonian	AW	2	2	.72	.85	.67
Italian	LS	2	2	-	-	.39
Korean	LS	2	2	-	.71	.74
Spanish	LS	12	5	.64	.48	.65
Swedish	LS	8	5	.95	-	.70
Japanese	LS	7	3	.86	.72	.78
Japanese	TL	9	8	.81	.37	.79
English	AW	21	12	.75	.57	.69
English	LS	26	15	.86	.51/.16	.64/.40

SENSEVAL-2 de-briefing

- Where next?
  - Supervised ML approaches worked best
    - Looking at the role of feature selection algorithms
  - Need a well-motivated sense inventory
    - Inter-annotator agreement went down when moving to WordNet senses
  - Need to tie WSD to real applications
    - The translation task was a good initial attempt

SENSEVAL-3 2004

- 14 core WSD tasks including
  - All words (Eng, Italian): 5000 word sample
  - Lexical sample (7 languages)
- Tasks for identifying semantic roles, for multilingual annotations, logical form, subcategorization frame acquisition

English lexcial sample task

- **Data collected from the Web from Web users**
- Guarantee at least two word senses per word
- 60 ambiguous nouns, adjectives, and verbs
- test data
  - ½ created by lexicographers
  - ½ from the web-based corpus
- Senses from WordNet 1.7.1 and **Wordsmyth** (verbs)
- Sense maps provided for fine-to-coarse sense mapping
- **Filter out multi-word expressions from data sets**

English lexical sample task

Class	Nr of words	Avg senses (fine)	Avg senses (coarse)
Nouns	20	5.8	4.35
Verbs	32	6.31	4.59
Adjectives	5	10.2	9.8
Total	57	6.47	4.96

Table 1: Summary of the sense inventory

Results

- 27 teams, 47 systems
- Most frequent sense baseline
  - 55.2% (fine-grained)
  - 64.5% (coarse)
- Most systems significantly above baseline
  - Including some unsupervised systems
- Best system
  - 72.9% (fine-grained)
  - 79.3% (coarse)

SENSEVAL-3 lexical sample results

System/Team	Description	Fine P	Fine R	Coarse P	Coarse R
htsa3 U.Bucharest (Grozea)	A Naive Bayes system, with correction of the a-priori frequencies, by dividing the output confidence of the senses by $frequency^{\alpha}$ ($\alpha = 0.2$)	72.9	72.9	79.3	79.3
IRST-Kernels ITC-IRST (Strapparava)	Kernel methods for pattern abstraction, paradigmatic and syntagmatic info. and unsupervised term proximity (LSA) on BNC, in an SVM classifier.	72.6	72.6	79.5	79.5
nusels Nat.U. Singapore (Lee)	A combination of knowledge sources (part-of-speech of neighbouring words, words in context, local collocations, syntactic relations), in an SVM classifier.	72.4	72.4	78.8	78.8
htsa4	Similar to htsa3, with different correction function of a-priori frequencies.	72.4	72.4	78.8	78.8
BCU_comb Basque Country U. (Agirre & Martinez)	An ensemble of decision lists, SVM, and vectorial similarity, improved with a variety of smoothing techniques. The features consist of local collocations, syntactic dependencies, bag-of-words, domain features.	72.3	72.3	78.9	78.9
htsa1	Similar to htsa3, but with smaller number of features.	72.2	72.2	78.7	78.7
rlsc-comb U.Bucharest (Popescu)	A regularized least-square classification (RLSC), using local and topical features, with a term weighting scheme.	72.2	72.2	78.4	78.4
htsa2	Similar to htsa4, but with smaller number of features.	72.1	72.1	78.6	78.6
BCU_english	Similar to BCU_comb, but with a vectorial space model learning.	72.0	72.0	79.1	79.1

SENSEVAL-3 results (unsupervised)

System/Team	Description	Fine		Coarse	
		P	R	P	R
wsdiit IIT Bombay (Ramakrishnan et al.)	An unsupervised system using a Lesk-like similarity between context of ambiguous words, and dictionary definitions. Experiments are performed for various window sizes, various similarity measures	66.1	65.7	73.9	74.1
Cymfony (Niu)	A Maximum Entropy model for unsupervised clustering, using neighboring words and syntactic structures as features. A few annotated instances are used to map context clusters to WordNet/Worsmyth senses.	56.3	56.3	66.4	66.4
Prob0 Cambridge U. (Preiss)	A combination of two unsupervised modules, using basic part of speech and frequency information.	54.7	54.7	63.6	63.6
clr04-ls CL Research (Litkowski)	An unsupervised system relying on definition properties (syntactic, semantic, subcategorization patterns, other lexical information), as given in a dictionary. Performance is generally a function of how well senses are distinguished.	45.0	45.0	55.5	55.5
CIAOSENSO U. Genova (Buscaldi)	An unsupervised system that combines the conceptual density idea with the frequency of words to disambiguate; information about domains is also taken into account.	50.1	41.7	59.1	49.3

Pseudowords

- Artificial words created by concatenation of two randomly chosen words
- E.g. “banana” + “door” => “banana-door”

- Pseudowords can generate training and test data for WSD automatically. How?

- Issues with pseudowords?