CS674 Natural Language Processing

- Last class
  - Metaphor
  - Synonymy, hyponymy
  - Lexical semantic resources
- Today
  - Word sense disambiguation
    - » Supervised
    - » Weakly supervised

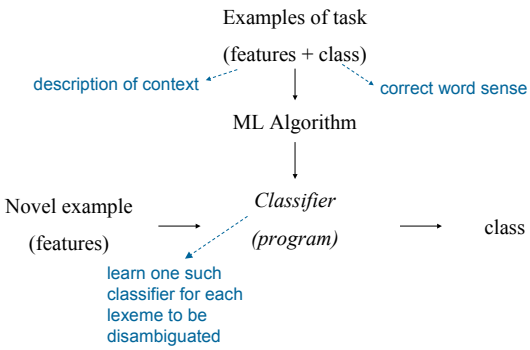
Word sense disambiguation

- Given a *fixed* set of senses is associated with a lexical item, determine which of them applies to a particular instance of the lexical item
- Two fundamental approaches
  - WSD occurs during semantic analysis as a side-effect of the elimination of ill-formed semantic representations
  - Stand-alone approach
    - » WSD is performed independent of, and prior to, compositional semantic analysis
    - » Makes minimal assumptions about what information will be available from other NLP processes
    - » Applicable in large-scale practical applications

Machine learning approaches

- Inductive machine learning methods
  - Supervised
  - Bootstrapping
  - Unsupervised
- Emphasis is on acquiring the knowledge needed for the task from data, rather than from human analysts.

Inductive ML framework



Feature vector input

- **target:** the word to be disambiguated
- **context** : portion of the surrounding text
  - Tagged with part-of-speech information
  - Select a "window" size
  - Stemming or morphological processing
  - Possibly some partial parsing
- Convert the context into a set of features
  - Attribute-value pairs
    - » Numeric or nominal values

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.*
  - [guitar, NN1, and, CJC, player, NN1, stand, VVB]

Co-occurrence features

- Encodes information about neighboring words, ignoring exact positions.
  - **Features**: the words themselves (or their roots)
  - **Values**: number of times the word occurs in a region surrounding the target word
  - 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) for the previous example:
      [0,0,0,1,0,0,0,0,0,0,1,0]

Naïve Bayes classifiers for WSD

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

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

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

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

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

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

- Mooney (1996) reports on *line* corpus that naïve-Bayes and an ANN worked best, achieving 73% correct.

WSD Evaluation

- Baseline: most frequent sense
- Corpora:
  - *line* corpus
  - Yarowsky's 1995 corpus
    - » 12 words (plant, space, bass, …)
    - » ~4000 instances of each
  - SEMCOR (Landes et al. 1998)
    - » Portion of the Brown corpus tagged with WordNet senses
  - SENSEVAL (Kilgarriff and Rosenzweig, 2000)
    - » Also provides an evaluation framework (Kilgarriff and Palmer, 2000) a la MUC and TREC WSD Evaluation

WSD Evaluation

- Metrics
  - Precision
    - » 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

Decision list classifiers

- Equivalent to simple case statements.
- Classifier consists of a sequence of tests to be applied to each input 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

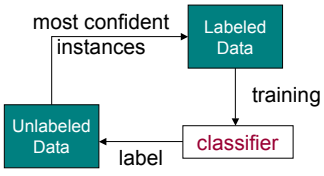
- Consists of *generating* and *ordering* individual tests based on the characteristics of the training data
- Generation: every feature-value pair constitutes a test
- Ordering: based on accuracy on the training set

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

- Associate the appropriate sense with each test

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

Klucevsek **play**s Giulietti or Titano piano accordions with the more flexible, more difficult free **bass** rather than the traditional Stradella **bass** with its preset chords designed mainly for accompaniment.

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.

Associates describe Mr. Whitacre as a quiet, disciplined and assertive manager whose favorite form of escape is **bass fish**ing.

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.

Saturday morning I arise at 8:30 and click on "America's best-known **fish**erman," giving advice on catching **bass** in cold weather from the seat of a bass boat in Louisiana.

Yarowsky's bootstrapping approach

- Relies on a **one sense per discourse** constraint: The sense of a target word is highly consistent within any given document
  - Evaluation on ~37,000 examples

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%

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. The unlabeled examples are called the *residual*.
3. Build a classifier (decision list) by training a supervised learning algorithm with the labeled examples.
4. Apply the classifier to all the examples. Find members of the residual 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 filter and/or augment the new examples.
6. Go to Step 3. Repeat until the residual set is stable.