CS 181: Natural Language Processing

Lecture 19: Computational Lexical Semantics

Kim Bruce
Pomona College
Spring 2008

Disclaimer: Slide contents borrowed from many sources on web!

Finding Word Senses

- Compositional Semantics
- Based on meanings of words
- Many words have several meanings:
  - bank, dish, bass, ...
- How can we find meanings?
  - So far: lexical relations between senses, WordNet, thematic roles & PropBank, selectional restrictions

Primitive Decomposition

- Define word wrt set of primitive semantic features
- Example
  - hen: +female, +chicken, +adult
  - rooster -female, +chicken, +adult
  - chick +chicken, -adult

Express Propositionally

- John opened the door
  - (CAUSE(John(BECOME(OPEN(door)))))
- The door opened
  - (BECOME(OPEN(door)))
- The door is open
  - (OPEN(door))
- Ties together meanings of all sentences with DOOR, BECOME, etc.

More Radical

Primitive	Definition
ATRANS	The abstract transfer of possession or control from one entity to another.
PTRANS	The physical transfer of an object from one location to another
MTRANS	The transfer of mental concepts between entities or within an entity.
MBUILD	The creation of new information within an entity.
PROPEL	The application of physical force to move an object.
MOVE	The integral movement of a body part by an animal.
INGEST	The taking in of a substance by an animal.
EXPEL	The expulsion of something from an animal.
SPEAK	The action of producing a sound.
ATTEND	The action of focusing a sense organ.

Conceptual Dependency

- The waiter brought Mary the check.
  - $\exists x, y\ Atrans(x) \wedge Actor(x, Waiter) \wedge Object(x, Check) \wedge To(x, Mary) \wedge Ptrans(y) \wedge Actor(y, Waiter) \wedge Object(y, Check) \wedge To(y, Mary)$
  - Physically conveyed + transfered control (abstract)
- Hard to come up with primitives, so not used much.

Zipf's Law

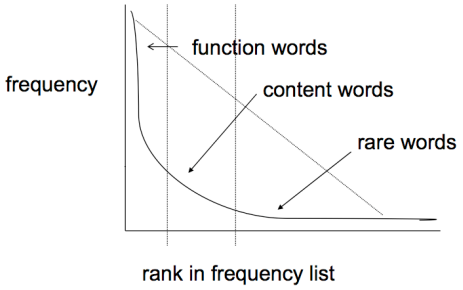
Zipf's law

- given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table.
- Exists constant k s.t. freq*rank = k
- In Brown corpus,
  1. "the", 7%
  2. "of", 3.5%
  3. "and", 2.8%
  4. ...

Tom Sawyer

Word	Freq. (f)	Rank (r)	$f \cdot r$	Word	Freq. (f)	Rank (r)	$f \cdot r$
the	3332	1	3332	turned	51	200	10200
and	2972	2	5944	you'll	30	300	9000
a	1775	3	5235	name	21	400	8400
he	877	10	8770	comes	16	500	8000
but	410	20	8400	group	13	600	7800
be	294	30	8820	lead	11	700	7700
there	222	40	8880	friends	10	800	8000
one	172	50	8600	begin	9	900	8100
about	158	60	9480	family	8	1000	8000
more	138	70	9660	brushed	4	2000	8000
never	124	80	9920	sins	2	3000	6000
Oh	116	90	10440	Could	2	4000	8000
two	104	100	10400	Applausive	1	8000	8000

English



Zipf's Law

- Useful as a rough description of frequency distribution of words in corpora.
- Also shows up in lots of other places.
  - word senses
  - references to scientific papers
  - royalties to pop-music composers
  - ...

Computational Lexical Semantics

Word Sense Disambiguation

- Use labeled corpora and context to determine word sense.
- Easy if only a few words, harder if need to disambiguate all.

Targeted Word Sense

- Suppose have hand-labeled data.
- Use supervised learning with feature vectors.
- Look at limited radius around word to determine neighboring words.
- May just look at as set or by exact location (collocation features).

Feature Vectors

- Window of n words around target word
- Encode info about these words
  - E.g., words, root forms, POS tags, ...
    - 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.
  - Surrounding context (local features)
    - [(guitar, NN1), (and, CJC), (player, NN1, (stand,VVB)]

More Feature Vectors

- More info about words
  - Frequent co-occurring words (*bag-of-words*)
    - [fishing, big, sound, player, fly, rod, pound, double, runs, playing, guitar, band]
    - [0,0,0,1,0,0,0,0,0,0,1,0]
  - Other features:
    - [followed by "player", contains "show" in sentence,...]
    - [yes, no, ...]

Naive Bayes

Naive Bayesian Classifier

- Assume choosing best sense ŝ given a feature vector is same as choosing most probable sense given the vector.

$$\hat{s} = argmax_{s \in S} P(s|\vec{f})$$

- Data too sparse, so use Bayes:

$$\hat{s} = argmax_{s \in S} \frac{P(\vec{f}|s)P(s)}{P(\vec{f})}$$

Naive Bayesian Classifier

- Still too hard, so assume features are conditionally independent:

$$P(\vec{f}|s) \equiv \prod_{j=1}^{n} P(f_j|s)$$

- Hence:

$$\hat{s} = argmax_{s \in S} P(s) \prod_{j=1}^{n} P(f_j|s)$$

Training

$$P(s_i) = \frac{count(s_i, w)}{count(w)}$$

- Use estimates:

$$P(f_j|s) = \frac{count(f_j, s)}{count(s)}$$

- Error warning: Numbers are small so safer to take logs and add.

Example

- Suppose 2000 instance of "bank", 1,500 for bank/1 (savings) & 500 for bank/2 (river)
  - P(S = 1) = 1,500/2,000 = .75
  - P(S = 2) = 500/2,000 = .25
- Suppose "credit" occurs 200 w/ bank/1 and 4 times w/ bank/2
  - P(F1 = "credit") = 204/2000 = .102
  - P(F1 = "credit" | S=1) = 200/1,500 = .133
  - P(F1 = "credit" | S=2) = 4/500 = .008
- Suppose have "bank" instance w/ feature "credit"
  - P(S=1 | F1 = "credit") = .133*.75/.102 = .978
  - P(S=2 | F1 = "credit") = .008*.25/.102 = .020

Comparing Bayes with ...

- Studies comparing naive Bayes with
  - Neural network & context vector (1993)
  - Neural network, decision tree, disjunctive/ conjunctive normal form learners, perceptron (1996)
  - Decision tree, rule-based learner, probabilistic model, etc. (1998)
- All found naive Bayes classifier performed as well as any of the others
- Disadvantage - hard to interpret answers

Decision Lists & Trees

Decision Lists

- Widely used in machine learning
- Represent problem as series of questions (about presence of features)
- Easier to interpret solutions than naive Bayes

Decision List Classifiers

- Equivalent to simple case statements.
- Sequence of tests applied to feature vector.
- If test succeeds then return corresponding sense.
- If fails, continue to next test.

How To Form Questions

- Identify collocational features from tagged data.
- Yarowsky: Each individual feature-value pair is a test
- Order tests by how likely they are to distinguish via conditional probabilities:

$$\left| log \left(\frac{P(s_1|f_i)}{P(s_2|f_i} \right) \right|$$

- Associate appropriate sense w/each test

Recall Example

- Suppose 2000 instance of "bank", 1,500 for bank/1 (savings) & 500 for bank/2 (river)
  - P(S = 1) = 1,500/2,000 = .75
  - P(S = 2) = 500/2,000 = .25
- Suppose "credit" occurs 200 w/ bank/1 and 4 times w/ bank/2
  - P(F1 = "credit") = 204/2000 = .102
  - P(F1 = "credit" | S=1) = 200/1,500 = .133
  - P(F1 = "credit" | S=2) = 4/500 = .008
- Suppose have "bank" instance w/ feature "credit"
  - P(S=1 | F1 = "credit") = .133*.75/.102 = .978
  - P(S=2 | F1 = "credit") = .008*.25/.102 = .020
- DL score = | log(.978/.020)| = 3.89

Using Decision List

- Go through DL (in order) looking for match. First match gives sense. If all fail, return majority sense

DL-score	Feature	Sense
3.89	*credit* w/in bank	bank/1 financial
2.20	bank *is muddy*	bank/2 river
1.09	*pole* w/in bank	bank/2 river
0.00	*of the* bank	*useless!*

Decision Tree

- Look for features that divide senses
- Well-known decision tree learning algorithms like C4.5.

Evaluating WSD

- Metrics:
  - Precision: % right out of those attempted
  - Recall: % right overall
- Difficulty in judging:
  - Results depend on coarse vs. fine-grain senses
  - Distinguish chair as seat vs professor easy
  - Distinguish bank institution vs. building hard
  - Clustering senses may help

Evaluating

- Upper bounds:
  - Human:
    - 95% and up if clear and distinct (coarse)
    - 65% - 70% if polysemous (related senses)
- Lower bounds:
  - Choose most frequent sense
    - WordNet orders by frequency
  - How good is 90%?
    - Excellent if 2 senses equally likely
    - Trivial if most frequent sense occurs 9 out of 10

Testing Your Technique

- SemCor subset of Brown w/ 234K words
- Concatenate randomly chosen words:
  - E.g., cat + brick = catbrick
- Replace all occurrences of each by combo
- Now run algorithm to see if can find original
- In general gives overly optimistic since generally easier to disambiguate.

Dictionary and Thesaurus Methods

Simplified Lesk Algorithm

- Assume have machine-readable-dictionary
  - E.g., WordNet, OED, Collins, Roget Thesaurus
- Identify sense using definition overlap
  - Get all sense definitions of word to be disambiguated
  - For each sense definition determine overlap of definition with context of word use.
  - Choose one w/most overlap

Example

- Disambiguate "pine" in
  "Pine cones hanging in a tree".
- Sense definitions:
  1. Kinds of evergreen tree with needle-shaped leaves
  2. Waste away through sorrow or illness
- Pick max:
  - size(Pine/1 ∩ sentence) = 1
  - size(Pine/2 ∩ sentence) = 0

Original Lesk

- More complicated -- disambiguate all words in sentence.
- Look for overlap of definitions of senses of words.
- Combinatorial explosion if lots of words!

Evaluating Lesk

- Competition: Senseval-2 all-words data, with back-off to most frequent sense
  - Original Lesk: 42%
  - Simplified Lesk: 58%
- Can improve (Corpus Lesk) if weight words on basis of inverse document frequency
  - Rare words count more.

Selectional Preferences

- Selectional restrictions on thematic roles used to rule out senses.
  - "Wash a dish" vs. "Cook a dish"
  - Dish as WASH-OBJECT vs COOK-FOOD
- Unsupervised.
  - How to learn these?
  - PropBank helps

Problems

- Violations often occur
  - The students threw dishes at each other.
  - You can't eat money.
- Complicated ways of measuring how much info predicate tells about semantic class of arguments.
- Doesn't perform as well as Lesk or other supervised methods.

Word Similarity

- Words in discourse must be related in meaning for discourse to be coherent.
- Use to aid WSD
- Find semantic similarity between
  - pairs of concepts
  - word & surrounding context

Semantic Similarity Metrics

- Similarity(C_1, C_2) = -log (path(C_1, C_2))
- To check words, may have many senses
  - wordsim(w_1,w_2) = max (sim(c_1, c_2)) over c_1 in senses(w_1), c_2 in senses(w_2)
- Relies on assumption that each link represents a uniform distance -- *Not true!!*
- Try to refine to match depth, but ...

Information Content Word Similarity

- (Resnik) P(c) = probability of seeing concept in large corpus

$$P(c) = \frac{\sum_{w \in words(c)} count(w)}{N}$$

- Information content: IC(c) = -log (P(c))
  - low probability = high info content
- LCS(c_1,c_2) = lowest common subsumer = lowest node subsuming c_1 & c_2

Resnik Similarity Measure

- $sim_{resnik}(c_1,c_2)$ = $IC(LCS(c_1,c_2))$
  - higher similarity, lower LCS, higher sim value
- Alternative (Jiang & Conrath)
  - $sim(c_1,c_2)$ = $2*IC(LCS(c_1,c_2)) - (IC(c_1) + IC(c_2))$

Extended Lesk

- Based on extended gloss overlap:
  - Look not just at words, but all related words!
  - similarity(c_1,c_2) = overlap(gloss(c_1),gloss(c_2))
    + overlap(gloss(hypo(c_1)),gloss(c_2))
    + overlap(gloss(c_1),gloss(hypo(c_2)))
    + overlap(gloss(hypo(c_1)),gloss (hypo(c_2)))
- If *rels* is set of relations whose glosses are compared then

$$sim_{eLesk}(c_1, c_2) = \sum_{r,q \in rels} overlap(gloss(r(c_1)), gloss(q(c_2)))$$

Evaluation

- Jiang-Conrath works better in practice than Resnik, though still only 39% on Senseval-2
- Extended Lesk similarity also works among best in practice.

Minimally Supervised WSD

Bootstrapping

- Getting fully annotated data is hard.
- Want to automatically build training set.
- Yarowsky:
  - Start with small hand-labeled training set
  - Use to classify some occurrences in unlabeled set.
  - Add those to the training set and repeat

Yarrow

- Used one sense per collocation:
  - Pick one word to associate w/each sense
    - fish for bass/1, play for bass/2
  - Label occurrences collocated w/words
  - train on sense labeled, looking for new collocations (e.g. for decision list)
  - Continue as long as helps
- 97% precision on words w/two senses
- 70% on SemCor words

Yarrow

- One sense per discourse
  - All occurrences in discourse have same sense
  - 98% accuracy if have two-way ambiguity
  - 70% if more fine grained (SemCor)
  - Fine for polysemy, not homonymy

Any Questions?