Lexical Semantics &
Word Sense Disambiguation

L545
Spring 2010

Lexical Semantics

A (word) sense is a representation of one meaning of a word

- bank\(^1\): financial institution
- bank\(^2\): sloped ground near water

Various relations:
- homonymy: 2 words/senses happen to sound the same (e.g., \textit{bank} \(^1\) & \textit{bank} \(^2\))
- polysemy: 2 senses have some semantic relation between them
  - \textit{bank} \(^1\) & \textit{bank} \(^3\) = repository for biological entities
- metonymy: subtype of polysemy where use of one aspect refers to other aspects
  - \textit{bank} \(^1\) & \textit{bank} \(^4\) = building belonging to a financial institution

Relations between senses

Synonymy and Antonymy

Synonymy: senses of 2 different words have (nearly) identical meanings
- One can be substituted for the other in a sentence without changing the propositional content of the sentence
- We define synonymy as a relation between senses, not words

Antonymy: words with opposite meanings
- binary opposition or at opposite ends of some scale

Relations between senses

Hyponymy

A hyponym is a more specific sense of another, i.e., a subclass
- e.g., car is a hyponym of vehicle

A hypernym (superordinate) is a more general sense of another, i.e., a superclass
- e.g., vehicle is a hypernym of car
- Class of vehicles includes as members all cars (i.e., every car is a vehicle)

A taxonomy is an arrangement of the elements of an ontology (set of distinct objects) into tree-like class inclusion structure

WordNet

WordNet (http://wordnet.princeton.edu/) is a database of lexical relations:
- Database of nouns (117,097); database of verbs (11,488); database of adjectives (22,141) & adverbs (4,601)

WordNet contains different senses of a word, defined by 
\textit{synsets} (synonym sets)
- \{chump\(^1\), fool\(^2\), gull\(^1\), mark\(^9\), patsy\(^1\), fall guy\(^1\), sucker\(^1\), soft touch\(^1\), mug\(^2\)\}
- \textit{synset gloss}: \textit{a person who is gullible and easy to take advantage of} 
  - These words, in these senses, can be substituted for each other

For nouns, WordNet encodes antonyms, hypo/hypernyms, mero/holonyms, as well as some more specific versions (e.g., substance meronym)
For verbs, WordNet encodes antonyms, hypernyms, troponyms, & entails relations

Semantic Fields

Part-whole relations: \textit{meronymy}
- \textit{wheel} is a meronym of \textit{car}
- \textit{car} is a holonym of \textit{wheel}

Semantic fields: more integrated/holistic approach to capturing relationships between entire sets of words from a single domain
- Background knowledge: captured by a frame, model, or script
- Air travel: reservation, flight, travel, buy, price, cost, fare, rates, meal, plane

WordNet

WordNet (http://wordnet.princeton.edu/) is a database of lexical relations:
- Database of nouns (117,097); database of verbs (11,488); database of adjectives (22,141) & adverbs (4,601)

WordNet contains different senses of a word, defined by \textit{synsets} (synonym sets)
- \{chump\(^1\), fool\(^2\), gull\(^1\), mark\(^9\), patsy\(^1\), fall guy\(^1\), sucker\(^1\), soft touch\(^1\), mug\(^2\)\}
- \textit{synset gloss}: \textit{a person who is gullible and easy to take advantage of} 
  - These words, in these senses, can be substituted for each other

For nouns, WordNet encodes antonyms, hypo/hypernyms, mero/holonyms, as well as some more specific versions (e.g., substance meronym)
For verbs, WordNet encodes antonyms, hypernyms, troponyms, & entails relations
Thematic roles

Event participants can be described in terms of thematic roles
• Capture the relationships between verbal arguments, e.g., commonality between Breakers and Openers

Common thematic roles (Fig. 19.5):

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT</td>
<td>volitional causer of an event</td>
</tr>
<tr>
<td>FORCE</td>
<td>non-volitional causer of an event</td>
</tr>
<tr>
<td>RESULT</td>
<td>end product of an event</td>
</tr>
<tr>
<td>INSTRUMENT</td>
<td>instrument used in an event</td>
</tr>
<tr>
<td>SOURCE</td>
<td>origin of the object of a transfer event</td>
</tr>
<tr>
<td>EXPERIENER</td>
<td>experiencer of an event</td>
</tr>
<tr>
<td>THEME</td>
<td>participant most directly affected by an event</td>
</tr>
<tr>
<td>BENEFICIARY</td>
<td>beneficiary of an event</td>
</tr>
<tr>
<td>CONTENT</td>
<td>proposition or content of a propositional event</td>
</tr>
<tr>
<td>GOAL</td>
<td>destination of an object of a transfer event</td>
</tr>
</tbody>
</table>

Selectional restrictions

Selectional restrictions are constraints a verb imposes on the types of concepts that may fill its argument roles

Consider: I want to eat someplace that’s close

1. Intransitive verb with locational adjunct?
2. Transitive verb?

Selectional restrictions help sort the meaning out

Representing selectional restrictions

Can extend event semantics to represent restrictions:

\[ \exists e, x, y \text{Eating}(e) \land \text{Agent}(e, x) \land \text{Theme}(e, y) \land \text{EdibleThing}(y) \]

Then, we ate a hamburger is encountered, the predicates have to be consistent:

\[ \exists e, x, y \text{Eating}(e) \land \text{Agent}(e, x) \land \text{Theme}(e, y) \land \text{EdibleThing}(y) \land \text{Hamburger}(y) \]

More easily, one can just state selectional restrictions in terms of WordNet synsets (rather than logical concepts)

• The THEME role of eat is restricted to \{food, nutrient\}

Diathesis alternations

Thematic roles generalize over different surface realizations of arguments

Many verbs allow their thematic roles to be realized in different syntactic positions

• John_{agent} broke the window_{theme}.
• John_{agent} broke the window_{theme} with a rock_{instrument}.
• The rock_{instrument} broke the window_{theme}.
• The window_{theme} broke.
• The window_{theme} was broken by John_{agent}

Diathesis alternations = multiple argument structure realizations

• Verbs with the same set of alternations tend to have the same meanings

Selectional restrictions (2)

Selectional restrictions are specific to a given sense (not to a word/lexeme)

• . . . they served green-lipped mussels from New Zealand [cooking]
• Which airlines serve Denver? [providing commercial service to]

Selection restrictions can be quite specific

• imagine: many restrictions on AGENT, not many on THEME
• diagonalize: THEME has to be a matrix
• odorless arguments have to be concepts which can have an odor

Primitive decomposition

One can also try to define a word by breaking it down into semantic features

• hen: +female, +chicken, +adult
• Jim killed his philodendron
  
  \[
  \begin{align*}
  & \text{KILL}(x, y) \iff \text{CAUSE}(x, \text{BECOME}(\text{NOT}(\text{ALIVE}(y))))
  
  \end{align*}
  \]

Need a large variety of predicates (e.g., OPEN or CAUSE) & an inventory of semantic primitives
Word Sense Disambiguation (WSD)

Word Sense Disambiguation (WSD) is the task of determining the proper sense of an ambiguous word in a given context. For example, given the word *bank*, is it:

- the rising ground bordering a body of water
- an establishment for exchanging funds
  - Or maybe, more generally, a repository (e.g., *blood bank*)

WSD comes in two variants:

- **lexical sample task**: small pre-selected set of target words (along with sense inventory)
- **all-words task**: entire texts . . . more data sparseness problems

Supervised WSD:

Supervised WSD: extract features which are helpful for particular senses & train a classifier to assign correct sense.

- **lexical sample** task: labeled corpora for individual words
- **all-word** disambiguation task: use a semantic concordance (e.g., SemCor)

Feature extraction

1. POS tag, lemmatize/stem, & perhaps parse the sentence in question
2. Extract context features within a certain window of a target word
   - Feature vector: numeric or nominal values encoding linguistic information

Collocational features

Collocational features encode information about specific positions to the left or right of a target word.

- capture local lexical & grammatical information

*Consider:* An electric guitar and **bass** player stand off to one side, not really part of the scene ...

\[
\begin{align*}
[w_{i-2}, \text{POS}_{i-2}, w_{i-1}, \text{POS}_{i-1}, w_{i+1}, \text{POS}_{i+1}, w_{i+2}, \text{POS}_{i+2}] \\
\text{[guitar, NN, and, CC, player, NN, stand, VB]}
\end{align*}
\]

Bag-of-words features

Bag-of-words features encode unordered sets of surrounding words, ignoring exact position.

- captures more semantic properties & general topic of discourse
- vocabulary for surrounding words usually pre-defined

*E.g.,* 12 most frequent content words from *bass* sentences in the WSJ:

\[
\text{[fishing, big, sound, player, fly, rod, pound, double, runs, playing, guitar, band]}
\]

leading to this feature vector:

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

Bayesian WSD

- Look at a context of surrounding words, call it *c*, within a window of a particular size
- Select the best sense *s* from among the different senses

\[
\begin{align*}
\hat{s} &= \arg \max_s P(s_k|c) \\
\hat{s} &= \arg \max_s \frac{P(c|s_k)P(s_k)}{P(c)} \\
\hat{s} &= \arg \max_s P(c|s_k)P(s_k)
\end{align*}
\]

It is actually computationally simpler to calculate logarithms, giving us:

\[
\begin{align*}
\hat{s} &= \arg \max_s \left[ \log P(c|s_k) + \log P(s_k) \right]
\end{align*}
\]
**Context features**

We treat the context ($c$) as a bag of words

- Each word in the context is a feature, $v_j$

Thus, to get the probability $P(c|s_k)$, we have to look at each $P(v_j|s_k)$

**Naive Bayes assumption**

We make the Naive Bayes assumption that every surrounding word $v_j$ is independent of the other ones:

(3) $P(c|s_k) = \prod_{v_j \in c} P(v_j|s_k)$

This means that we have:

(4) $s = \arg \max_{s_k} \left\{ \sum_{v_j \in c} \log P(v_j|s_k) + \log P(s_k) \right\}$

We get maximum likelihood estimates from the corpus to obtain $P(s_k)$ and $P(v_j|s_k)$

---

**WSD Evaluation**

- Extrinsic (in vivo) evaluation: evaluate WSD in the context of another task, e.g., question answering
- Intrinsic (in vitro) evaluation: evaluate WSD as a stand-alone system
  - Exact-match sense accuracy
  - Precision/recall measures, if systems pass on some labelings

**Baselines:**

- Most frequent sense: for WordNet, generally take first sense
- Lesk algorithm

Ceiling: inter-annotator agreement, generally 75-80%

---

**Dictionary-based WSD (Lesk algorithm)**

Use general characterizations of the senses to aid in disambiguation

**Intuition:** words found in a particular sense definition can provide contextual cues, e.g., for ash:

<table>
<thead>
<tr>
<th>Sense</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$: tree</td>
<td>a tree of the olive family</td>
</tr>
<tr>
<td>$s_2$: burned stuff</td>
<td>the solid residue left when combustible material is burned</td>
</tr>
</tbody>
</table>

If we find tree in the context of ash, the sense is more likely $s_1$

---

**The algorithm**

Look at words within the sense definition and the words within the definitions of context words, too (unioning over different senses)

1. Take all senses $s_k$ of a word $w$ and gather the set of words for each definition
   - Treat it as a bag of words
2. Gather all the words in the definitions of the surrounding words, within some context window
3. Calculate the overlap
4. Choose the sense with the higher overlap
Example

(5) This cigar **burns** slowly and creates a stiff ash.

(6) The ash is one of the last **trees** to come into leaf.

So, sense \( s_2 \) goes with the first sentence and \( s_1 \) with the second

- Note that, depending on the dictionary, **leaf** might also be a contextual cue for sense \( s_1 \) of **ash**

Problems with dictionary-based WSD

- Not very accurate: 50%-70%
- Highly dependent upon the choice of dictionary
- Not always clear whether the dictionary definitions align with what we think of as different senses

Selectional Preferences

**Idea:** Use selectional preferences to restrict the set of senses

**Background:** Kullback-Leibler divergence measures the difference between 2 probability distributions

\[
D(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}
\]

Following Resnik, first measure how strongly a verb constrains its direct object, its so-called **selectional preference strength** \( S(v) \)

- For the direct object (DO), focus only on the head word
- Group objects together into noun classes \( C \)

\[
S(v) = D(P(C|v)||P(C)) = \sum_c P(c|v) \log \frac{P(c|v)}{P(c)}
\]

For a given noun within a class, we use:

\[
A(v, n) = \max_{c \in \text{classes}(n)} A(v, c)
\]

Using selectional associations

So, if we have the following:

(11) Susan interrupted the chair.

we know that **chair** has two senses, falling under the categories people and furniture

(12) \( A(\text{interrupt, chair}) = A(\text{interrupt, people}) \)

Selectional Association

But this does not tell us how strongly a word is associated with one particular noun class

To get this, we find the proportion of times that this noun class is associated with the verb

\[
A(v, c) = \frac{P(c|v) \log \frac{P(c|v)}{P(c)}}{S(v)}
\]

(9)

For a given noun within a class, we use:

(10) \( A(v, n) = \max_{c \in \text{classes}(n)} A(v, c) \)

Bootstrapping

**Bootstrapping,** or **semi-supervised learning,** involves using a small set of labeled data to help label more data.

e.g., the Yarowsky algorithm

1. Train decision-list classifier on small labeled data
2. Apply classifier to large unlabeled data
3. Select examples with most confidence and add to labeled set
4. Repeat

Need good confidence metrics and a good way to generate initial seeds to form the labeled data set
Heuristic-based WSD

Can use a heuristic to automatically select seeds

- **One sense per discourse**: the sense of a word is highly consistent within a given document
- **One sense per collocation**: collocations rarely have multiple senses associated with them

Using collocational knowledge

Rank senses based on what collocations the word appears in, e.g., *show interest* might be strongly correlated with the ‘attention, concern’ usage of *interest*

- The collocational feature could actually be a surrounding POS tag, or a word in the object position
- For a given context, select which collocational feature will be used to disambiguate, based on which feature is strongest indicator
  - Avoid having to combine different pieces of information this way

Rankings are based on the following, where $f$ is a collocational feature:

\[
\frac{P(s_k_1|f)}{P(s_k_2|f)}
\]

Calculating collocations

1. Initially, calculate the collocations for $s_k$
2. Calculate the contexts in which an ambiguous word is assigned to $s_k$, based on those collocations
3. Calculate the set of collocations that are most characteristic of the contexts for $s_k$, using the formula:

\[
\frac{P(s_k_1|f)}{P(s_k_2|f)}
\]
4. Repeat steps 2 & 3 until a threshold is reached.

Using discourse knowledge

We also want to integrate discourse knowledge—in a given discourse, we are unlikely to see senses flip around

... the existence of *plant* and animal life ... classified as either *plant* or animal ... Although bacterial and *plant* cells are enclosed ...

So, we can take the output of a WSD algorithm (such as “one sense per collocation”) and change it so that every ambiguous word in a particular discourse is given the majority sense

- Or every word beneath some threshold of likelihood ...

Word similarity

Based on the notion of synonymy, we expect synonyms to behave similarly

We define this in two ways:

- Knowledge-based: thesaurus-based WSD
- Knowledge-free: distributional methods

Word similarity computations are useful for IR, QA, summarization, language modeling, automatic essay grading, etc.

Thesaurus-based WSD

Use essentially the same set-up as dictionary-based WSD, but now:

- instead of requiring context words to have overlapping dictionary definitions
- we require surrounding context words to list the focus word $w$ (or the subject code of $w$) as one of their topics

E.g., if an *animal* or *insect* appears in the context of *bass*, we choose the *fish* sense instead of the musical one

Alternative: use path lengths in an ontology like WordNet to calculate word similarity
Translation-based WSD

Idea: when disambiguating a word $w$, look for a combination of $w$ and some contextual word which translates to a particular pair, indicating a particular sense

- interest can be ‘legal share’ (Beteiligung in German) or ‘concern’ (Interesse)
- In the phrase show concern, we are more likely to translate to Interesse zeigen than Beteiligung zeigen
- So, in this English context, the German context tells us to go with the sense that corresponds to Interesse

Information-theoretic WSD

Instead of using all contextual features—which we assume are independent—an information-theoretic approach tries to find one disambiguating feature

- Take a set of possible indicators and determine which is the best, i.e., which gives the highest mutual information in the training data

Possible indicators:
- object of the verb
- the verb tense
- word to the left
- word to the right
- etc.

And then when sense tagging, find the value of that indicator and tag appropriately

Partitioning

More specifically, you have to determine what the values ($x_i$) of the indicator indicate, i.e. what sense ($s_i$) they point to.

- Assume two senses ($P_1$ and $P_2$), which can be captured in subsets $Q_1 = \{x_i | x_i$ indicates sense 1$\}$ and $Q_2 = \{x_i | x_i$ indicates sense 2$\}$
- We will have a set of indicator values $Q$; our goal is to partition $Q$ into these two sets

The partition we choose is the one which maximizes the mutual information scores $I(P_1,Q_1)$ and $I(P_2,Q_2)$

- The Flip-Flop algorithm is used when you have to automatically determine your senses (e.g., if using parallel text)

Disambiguation

After determining the best indicator and partitioning the values, disambiguating is easy:

1. Determine the value $x_i$ of the indicator for the ambiguous word.
2. If $x_i$ is in $Q_1$, assign it sense 1; otherwise, sense 2.

This method is also applicable for determining which indicators are best for a set of translation words

Unsupervised WSD

Without any information, you really cannot sense tag something ... but you can perform sense discrimination, or clustering

- In other words, you can group comparable senses together (even if you cannot give it the correct label)

We will look briefly at the EM (Expectation-Maximization) algorithm for this task, based on a Bayesian model
EM algorithm: Bayesian review

Remember how we did Bayesian WSD for supervised learning:

- Look at a context of surrounding words, call it \( c \) (\( v_j = \text{word in context} \)), within a window of a particular size
- Select the best sense \( s \) from among the different senses

\[
\begin{align*}
\text{s} &= \text{arg max } P(s_k|c) \\
&= \text{arg max } \frac{P(c|s_k)P(s_k)}{P(c)} \\
&= \text{arg max } \sum_{v_j \in c} \log P(v_j|s_k) + \log P(s_k) \\
&= \text{arg max } \sum_{v_j \in c} \log P(v_j|s_k) + \log P(s_k)
\end{align*}
\]

We need some other way to get estimates of \( P(s_k) \) and \( P(c|s_k) \)

EM algorithm (cont.)

(b) Maximization: Use the expected probabilities to re-estimate the parameters:

\[
\begin{align*}
\text{P}(v_j|s_k) &= \frac{\sum_{c_i \in c} P(c_i|s_k)}{\sum_{c_i \in c} \sum_{v_j \in c_i} P(c_i|s_k)} \\
\text{P}(s_k) &= \frac{\sum_{c_i \in c} P(c_i|s_k)}{\sum_{c_i \in c} P(c_i|s_k)}
\end{align*}
\]

→ Of all the times that \( v_j \) occurs in a context of any of this word’s senses, how often does \( v_j \) indicate \( s_k \)?

→ Of all the times that any sense generates \( c_i \), how often does \( s_k \) generate it?