Natural Language Processing

Compositional Semantics
Dan Klein – UC Berkeley

Truth-Conditional Semantics

Linguistic expressions:
- "Bob sings"

Logical translations:
- \( \text{sing}(\text{bob}) \)
- Could be \( p_{1218}(e_{397}) \)

Denotation:
- \([\text{bob}]\) = some specific person (in some context)
- \([\text{sing}(\text{bob})]\) = ???

Types on translations:
- \( \text{bob} : e \) (for entity)
- \( \text{sing}(\text{bob}) : t \) (for truth-value)

Sentences:
- Are either true or false (given how the world actually is)
- \( \text{Bob sings} : \text{sing}(\text{bob}) \)

So what about verbs (and verb phrases)?
- \( \text{sings} \) must combine with \( \text{bob} \) to produce \( \text{sings}(\text{bob}) \)
- The \( \lambda \)-calculus is a notation for functions whose arguments are not yet filled.
- \( \text{sings} : \lambda x.\text{sings}(x) \)
- This is predicate – a function which takes an entity (type e) and produces a truth value (type t). We can write its type as \( e \rightarrow t \).
- Adjectives?

Denotation

What do we do with logical translations?
- Translation language (logical form) has fewer ambiguities
- Can check truth value against a database
- Denotation ("evaluation") calculated using the database
- More usefully: assert truth and modify a database
- Questions: check whether a statement in a corpus entails the (question, answer) pair:
  - "Bob sings and dances" \( \rightarrow \) "Who sings?" + "Bob"
- Chain together facts and use them for comprehension
Other Cases

- Transitive verbs:
  - \( \lambda x. \lambda y. \text{likes}(x,y) \)
  - Two-place predicates of type \( e \rightarrow (e \rightarrow t) \).
  - \( \text{likes} \) \( \lambda y. \text{likes}(y,\text{Amy}) \) is just like a one-place predicate.

- Quantifiers:
  - What does “Everyone” mean here?
  - \( \forall x. \text{likes}(x,\text{Amy}) \)
  - Mostly works, but some problems:
    - Have to change our NP/VP rule.
    - Won’t work for “Amy likes everyone.”
    - “Everyone likes someone.”
    - This gets tricky quickly!

Indefinites

- First try
  - “Bob ate a waffle” : \( \text{ate}(\text{bob}, \text{waffle}) \)
  - “Amy ate a waffle” : \( \text{ate}(\text{amy}, \text{waffle}) \)

- Can’t be right!
  - \( \exists x. \text{waffle}(x) \land \text{ate}(\text{bob}, x) \)
  - What does the translation of “\( a \)” have to be?
  - What about “\( \text{any} \)”?
  - What about “\( \text{every} \)”?

Grounding

- Grounding
  - So why does the translation \( \lambda x. \lambda y. \text{likes}(x,y) \) have anything to do with actual liking?
  - It doesn’t (unless the denotation model says so)
  - Sometimes that’s enough: wire up \( \text{bought} \) to the appropriate entry in a database

- Meaning postulates
  - Insist, e.g. \( \forall x. \forall y. \text{likes}(x,y) \Rightarrow \text{knows}(x,y) \)
  - This gets into lexical semantics issues

- Statistical version?

Tense and Events

- In general, you don’t get far with verbs as predicates
  - Better to have event variables \( e \)
    - “Alice danced” : \( \text{dance}(\text{alice}) \)
    - \( \exists e : \text{dance}(e) \land \text{agent}(e, \text{alice}) \land \text{time}(e) \leq \text{now} \)
  - Event variables let you talk about non-trivial tense / aspect structures
    - “Alice had been dancing when Bob sneezed”
    - \( \exists e, e' : \text{dance}(e) \land \text{agent}(e, \text{alice}) \land \text{sneeze}(e') \land \text{agent}(e', \text{bob}) \land \text{time}(e) < \text{start}(e') \land \text{end}(e') = \text{end}(e') \land \text{time}(e') < \text{now} \)

Adverbs

- What about adverbs?
  - “Bob sings terribly”
  - \( \text{terribly} \) \( \langle \text{sings} \rangle \) \( \langle \text{bob} \rangle \)?
  - \( \exists e \text{ present}(e) \land \text{type}(e, \text{singing}) \land \text{agent}(e, \text{bob}) \land \text{manner}(e, \text{terrible}) ? \)
  - It’s really not this simple...

Propositional Attitudes

- “Bob thinks that I am a gummy bear”
  - \( \text{thinks} \) \( \langle \text{bob}, \text{gummy} \langle \text{me} \rangle \rangle ? \)
  - \( \text{thinks} \) \( \langle \text{bob}, \text{“I am a gummy bear”} \rangle ? \)
  - \( \text{thinks} \) \( \langle \text{bob}, \text{“gummy} \langle \text{me} \rangle \rangle ? \)

- Usual solution involves intensions \( (\langle X \rangle) \) which are, roughly, the set of possible worlds (or conditions) in which \( X \) is true

- Hard to deal with computationally
  - Modeling other agents models, etc
  - Can come up in simple dialog scenarios, e.g., if you want to talk about what your bill claims you bought vs. what you actually bought
Trickier Stuff

- Non-Intersective Adjectives
  - green ball: $\lambda x. [\text{green}(x) \land \text{ball}(x)]$
  - fake diamond: $\lambda x. [\text{fake}(x) \land \text{diamond}(x)]$

- Generalized Quantifiers
  - the: $\mu f. [\text{unique-member}(f)]$
  - all: $\forall x. [\text{fake}(x) \rightarrow \text{diamond}(x)]$
  - could do with more general second order predicates, too (why worse?)

- Generalics
  - “Cats like naps”
  - “The players scored a goal”

- Pronouns (and bound anaphora)
  - “If you have a dime, put it in the meter.”

- ... the list goes on and on!

Multiple Quantifiers

- Quantifier scope
  - Groucho Marx celebrates quantifier order ambiguity:
    - “In this country a woman gives birth every 15 min. Our job is to find that woman and stop her.”

- Deciding between readings
  - “Bob bought a pumpkin every Halloween”
  - “Bob uses a phone as an alarm each morning”

- Multiple ways to work this out
  - Make it syntactic (movement)
  - Make it lexical (type-shifting)

Modeling Uncertainty

- Big difference between statistical disambiguation and statistical reasoning.

  The scout saw the enemy soldiers with night goggles.

- With probabilistic parsers, can say things like “72% belief that the PP attaches to the NP.”
- That means that probably the enemy has night vision goggles.
- However, you can’t throw a logical assertion into a theorem prover with 72% confidence.
- Use this to decide the expected utility of calling reinforcements?
- In short, we need probabilistic reasoning, not just probabilistic disambiguation followed by symbolic reasoning

Logical Form Translation

CCG Parsing

- Combinatory Categorial Grammar
  - Fully (mono-) lexicalized grammar
  - Categories encode argument sequences
  - Very closely related to the lambda calculus
  - Can have spurious ambiguities (why?)

Mapping to LF: Zettlemoyer & Collins 05/07

The task:

Input: List one way flights to Prague.
Output: $\lambda x. \text{flight}(x) \land \text{one-way}(x) \land \text{to}(x, \text{PRG})$

Challenging learning problem:

- Derivations (or parses) are not annotated
- Approach: [Zettlemoyer & Collins 2005]
- Learn a lexicon and parameters for a weighted Combinatory Categorial Grammar (CCG)
**Background**

- Combinatory Categorial Grammar (CCG)
- Weighted CCGs
- Learning lexical entries: GENLEX

**CCG Lexicon**

<table>
<thead>
<tr>
<th>Words</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>flights</td>
<td>N : ( \lambda x. \text{flight}(x) )</td>
</tr>
<tr>
<td>to</td>
<td>(N\N)/NP : ( \lambda x. \text{Af}. \lambda y. f(x) \wedge \text{to}(y,x) )</td>
</tr>
<tr>
<td>Prague</td>
<td>NP : PRG</td>
</tr>
<tr>
<td>New York city</td>
<td>NP : NYC</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Parsing Rules (Combinators)**

Application
- \( X/Y : f \quad Y : a \Rightarrow X : f(a) \)
- \( Y : a \quad X \backslash Y : f \Rightarrow X : f(a) \)

Composition
- \( X/Y : f \quad Y/Z : g \Rightarrow X/Z : \lambda x.f(g(x)) \)
- \( Y \backslash Z : f \quad X \backslash Y : g \Rightarrow X \backslash Z : \lambda x.f(g(x)) \)

Additional rules:
- Type Raising
- Crossed Composition

**Weighted CCG**

Given a log-linear model with a CCG lexicon \( \Lambda \), a feature vector \( f \), and weights \( w \):

- The best parse is:
  \[
  y^* = \arg\max_y \ w \cdot f(x,y)
  \]

Where we consider all possible parses \( y \) for the sentence \( x \) given the lexicon \( \Lambda \).

**CCG Parsing**

**Lexical Generation**

**Input Training Example**

Sentence: Show me flights to Prague.
Logic Form: \( \lambda x. \text{flight}(x) \wedge \text{to}(x, \text{PRG}) \)

**Output Lexicon**

<table>
<thead>
<tr>
<th>Words</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show me</td>
<td>S/N : ( \lambda f )</td>
</tr>
<tr>
<td>flights</td>
<td>N : ( \lambda x. \text{flight}(x) )</td>
</tr>
<tr>
<td>to</td>
<td>(N\N)/NP : ( \lambda x. \text{Af}. \lambda y. f(x) \wedge \text{to}(y,x) )</td>
</tr>
<tr>
<td>Prague</td>
<td>NP : PRG</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
GENLEX: Substrings X Categories

Input Training Example

| Sentence: | Show me flights to Prague. |
| Logic Form: | $\lambda x.\text{flight}(x) \land \text{to}(x, \text{PRG})$ |

Output Lexicon

All possible substrings:

| Show me flights to Prague. |

Categories created by rules that trigger on the logical form:

| $\lambda x.\text{flight}(x)$ |

Robustness

The lexical entries that work for:

Will not parse:

Relaxed Parsing Rules

Two changes

- Add application and composition rules that relax word order
- Add type shifting rules to recover missing words

These rules significantly relax the grammar

- Introduce features to count the number of times each new rule is used in a parse

Disharmonic Application

- Reverse the direction of the principal category:

| $\lambda x.\text{flight}(x)$ |

Missing content words

Insert missing semantic content

| $\lambda x.\text{flight}(x)$ |

Review: Application

| $\lambda x.\text{flight}(x)$ | $\lambda x.\text{flight}(x)$ |

[Zettlemoyer & Collins 2005]
**Missing content-free words**

**Bypass missing nouns**
- $N \triangleright N : f \rightarrow N : f(x, \text{true})$

<table>
<thead>
<tr>
<th>Northwest Air</th>
<th>to Prague</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda x. \text{airline}(x, \text{NWA}) \land \lambda x. \text{to}(x, \text{PRG})$</td>
<td>$\lambda x. \text{airline}(x, \text{NWA}) \land \lambda x. \text{to}(x, \text{PRG})$</td>
</tr>
</tbody>
</table>

**Related Work for Evaluation**

**Hidden Vector State Model: He and Young 2006**
- Learns a probabilistic push-down automaton with EM
- Is integrated with speech recognition

**WASP: Wong & Mooney 2007**
- Builds a synchronous CCG with statistical machine translation techniques
- Easily applied to different languages

**Zettlemoyer and Collins 2005**
- Uses GENLEX with maximum likelihood batch training and stricter grammar

**Two Natural Language Interfaces**

**ATIS (travel planning)**
- Manually-transcribed speech queries
- 4500 training examples
- 500 example development set
- 500 test examples

**Geo880 (geography)**
- Edited sentences
- 600 training examples
- 280 test examples

**Evaluation Metrics**

- Precision, Recall, and F-measure for:
  - Completely correct logical forms
  - Attribute / value partial credit

- $\lambda x. \text{flight}(x) \land \text{from}(x, \text{BOS}) \land \text{to}(x, \text{PRG})$

  is represented as:

  $\{ \text{from} = \text{BOS}, \text{to} = \text{PRG} \}$

**Two-Pass Parsing**

- Simple method to improve recall:
  - For each test sentence that can not be parsed:
    - Reparse with word skipping
    - Every skipped word adds a constant penalty
    - Output the highest scoring new parse

**Inputs:** Training set $\{(x_i, z_i) | i=1...n\}$ of sentences and logical forms. Initial lexicon $\Lambda$. Initial parameters $w$. Number of iterations $T$.

**Training:** For $t = 1...T$, $i = 1...n$:

1. **Step 1: Check Correctness**
   - Let $y^* = \arg \max_w f(x_i, y)$
   - If $L(y^*) = z_i$, go to the next example

2. **Step 2: Lexical Generation**
   - Let $\lambda_i$ be the lexical entries in $y^*$
   - Set lexicon to $\Lambda = \Lambda \cup \lambda_i$

3. **Step 3: Update Parameters**
   - Let $\lambda' = \arg \max_w f(x_i, y)$
   - Define $\lambda$ to be the lexical entries in $y'$
   - Set lexicon to $\Lambda = \Lambda \cup \lambda$
   - Set $w' = w + f(x_i, y') - f(x_i, y)$

**Output:** Lexicon $\Lambda$ and parameters $w$. 
### ATIS Test Set [Z+C 2007]

#### Exact Match Accuracy:

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Pass</td>
<td>90.61</td>
<td>81.92</td>
<td><strong>86.05</strong></td>
</tr>
<tr>
<td>Two-Pass</td>
<td>85.75</td>
<td>84.80</td>
<td>85.16</td>
</tr>
</tbody>
</table>

### Geo880 Test Set

#### Exact Match Accuracy:

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-Pass</td>
<td>95.49</td>
<td>83.20</td>
<td><strong>88.93</strong></td>
</tr>
<tr>
<td>Two-Pass</td>
<td>91.63</td>
<td>86.07</td>
<td>88.76</td>
</tr>
<tr>
<td>Zettlemoyer &amp; Collins 2005</td>
<td>96.25</td>
<td>79.29</td>
<td>86.95</td>
</tr>
<tr>
<td>Wong &amp; Mooney 2007</td>
<td>93.72</td>
<td>80.00</td>
<td><strong>86.31</strong></td>
</tr>
</tbody>
</table>