A meaning representation language is a formal language that unambiguously represents NL semantics (e.g., first order logic).

Semantic parsing is the task of translating sentences into meaning representations (MRs) automatically.

Compositionality is crucial for semantic parsing.

In this lecture, we will discuss an approach to semantic parsing that is based on a syntactic parser (Ge and Mooney 2009).

All figures are taken from Ge and Mooney (2009).
RoboCup: virtual football players receive coaching instructions in the formal language CLang.

Example: *If our player 2 has the ball, then position our player 5 in the midfield.*

\[\text{((bowner (player our 2)) (do (player our 5) (pos (midfield)))})\]

Geoquery: formal language for querying a database about US geography.

Example: *Which river is the longest?*

\[\text{answer(x1,longest(x1,river(x1)))}\]
**Meaning Representation Languages**

**ROBOCUP**: virtual football players receive coaching instructions in the formal language CLang.

Example: *If our player 2 has the ball, then position our player 5 in the midfield.*

```
((bowner (player our 2)) (do (player our 5) (pos (midfield))))
```

**GEOQUERY**: formal language for querying a database about US geography.

Example: *Which river is the longest?*

```
answer(x1,longest(x1,river(x1)))
```

Both MR languages are unambiguously defined by a context-free grammar, which also defines the *predicate-argument structure.*
Semantic Parsing

Steps in the semantic parsing process:

1. Generate syntactic parse for input sentence;
2. Use semantic lexicon to assign semantic predicates to words;
3. Construct all possible MRs by recursively applying composition rules to the parse tree;
4. Use disambiguation model to score MRs.

Step 1 and 2 generate a semantically-augmented parse tree (SAPT): semantic labels added to non-leaf nodes in the syntactic tree to specify MR predicate and unfilled arguments.

Step 3 uses the SAPT to guide the composition of the MR, resulting in a semantic derivation for the sentence.

Meaning Composition

Semantic derivation:

\[(\text{owner} \ (\text{player our \ } \{2\}))\]

Macro-predicates

In some cases, the structure of the parse tree mismatches the predicate-argument structure of the MR:
Macro-predicates

Introduce *macro-predicates* to adjust the meaning representation in these cases:

$$\lambda a_1 \lambda a_2 \lambda \mathbf{p} \mathbf{DO} \lambda p_1 \mathbf{P_O S} = \lambda \mathbf{p} \mathbf{DO}$$

Then:

$$\lambda \mathbf{p} \mathbf{P_O S} = \lambda p_1 \mathbf{P_O S}$$

Sentence: "Our player 2 has the ball"

(bowner (player our {2}))

Syntactic parser
Semantic Lexicon
Composition rules
Disambiguation component

Overall Architecture for Semantic Parsing

Inducing the Semantic Lexicon

Learn a semantic lexicon and composition rules:

- training set contains NL sentences with gold-standard meaning representations;
- parse the MRs (unambiguously defined by MR grammar);
- use techniques from statistical machine translation to align the words in the NL sentence with the predicates in the MR;
- parse the NL sentence using the Collins parser;
- using word alignments and NL parse, generate a SAPT;
- extract semantic lexicon and composition rules from SAPT;
- some cases require special treatment: predicate not aligned to any word; predicate aligned to multiple words.

Inducing the semantic lexicon:

- the ML parse is linearized (broken down into individual rules);
- then word alignments between the words in the NL sentence and the MR rules are induced.

If
our
player
4
has
the
ball

RULE → (CONDITION DIRECTIVE)
CONDITION → (bowner Team [UNUM])
TEAM → our
UNUM → 4

Figure from Wong and Mooney (2006).
Inducing Composition Rules

Formally, composition rules are of the form:

\[ \Lambda_1.P_1 + \Lambda_2.P_2 \Rightarrow \{ \Lambda_p.P_p, R \} \]

where \( P_1, P_2 \) and \( P_p \) are predicates for the left child, right child, and parent node.

\( \Lambda \) is lambda term of the form \( \langle \lambda p_1, \ldots, \lambda p_m, \lambda a_1, \ldots, \lambda a_k \rangle \).

\( R \) specifies how the arguments of the parent predicates are composed: \( \{ a_{k_i} = R_1, \ldots, a_{k_i} = R_l \} \), where \( R_i \) is a child \( (c_i) \), or a child's argument \( (c_i, a_j) \).

Example: the rule extracted for player 2:

\[ \langle \lambda a_1 \lambda a_2 \rangle . P_{\text{PLAYER}} + P_{\text{UNUM}} \Rightarrow \{ \lambda a_1 . P_{\text{PLAYER}}, a_2 = c_2 \} \]

for position our player 5:

\[ \lambda a_1 . P_{\text{POS}} + P_{\text{PLAYER}} \Rightarrow \{ \langle \lambda p_1 \lambda a_2 \rangle . P_{\text{DO_POS}}, a_1 = c_2 \} \]

for position our player 5 in the midfield:

\[ \langle \lambda p_1 \lambda a_2 \rangle . P_{\text{DO_POS}} + P_{\text{MIDFIELD}} \Rightarrow \{ \lambda p_1 . P_{\text{DO_POS}}, \{ a_1 = (c_1, a_1), a_2 = c_2 \} \} \]

Disambiguation Model

Semantic knowledge is applied to each example to obtain all possible derivations.

A maximum entropy model is used to pick the best one. It defines a distribution over semantic derivations \( (D) \) given an NL sentence \( S \) and its syntactic parse \( T \):

\[ Pr(D|S, T; \theta) = \frac{\exp \sum \theta_i f_i(D)}{Z_\theta(S, T)} \]

where \( (f_1, \ldots, f_n) \) is a feature vector parametrized by \( \theta \), and \( Z_\theta(S, T) \) is a normalizing factor.
Disambiguation Model

Features:
- **lexical features**: number of times a word is assigned a particular predicate;
- **bilexical features**: number of times a word is assigned a predicate and a particular word precedes or follows it;
- **rule features**: number of times a particular composition rule is applied in the derivation.

Training:
- find parameters $\tilde{\theta}^*$ that maximize the likelihood of the MRs in the training set;
- correct semantic derivation is not provided: MR likelihood is defined as the sum of the probability of all derivations;
- parameters estimated using limited-memory BFGS.

Evaluation

Performance on CLANG:

<table>
<thead>
<tr>
<th></th>
<th>CLANG</th>
<th></th>
<th>GEOQUERY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F-measure</td>
</tr>
<tr>
<td>GoldSyn</td>
<td>84.73</td>
<td>74.00</td>
<td>79.00</td>
</tr>
<tr>
<td>Syn20</td>
<td>85.37</td>
<td>70.00</td>
<td>76.92</td>
</tr>
<tr>
<td>Syn0</td>
<td>87.01</td>
<td>67.00</td>
<td>75.71</td>
</tr>
</tbody>
</table>

Performance on GEOQUERY:

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoldSyn</td>
<td>91.94</td>
<td>88.18</td>
<td>90.02</td>
<td></td>
</tr>
<tr>
<td>Syn40</td>
<td>90.21</td>
<td>86.93</td>
<td>88.54</td>
<td></td>
</tr>
<tr>
<td>Syn0</td>
<td>81.76</td>
<td>78.98</td>
<td>80.35</td>
<td></td>
</tr>
</tbody>
</table>

Summary

- Semantic parsing is the task of translating sentences into meaning representations;
- this requires a syntactic parser plus a semantic lexicon, and composition rules;
- the semantic lexicon and composition rules can be learned by aligning words and semantic predicates;
- a statistical model can be used for disambiguating meaning representations;
- the accuracy of the syntactic parser is crucial for performance.