Introduction to NLP

Semantic Parsing
Semantic Parsing

• Converting natural language to a logical form
  – e.g., executable code for a specific application

• Example:
  – Airline reservations
  – Geographical query systems
Stages of Semantic Parsing

- Input
  - Sentence
- Syntactic Analysis
  - Syntactic structure
- Semantic Analysis
  - Semantic representation
Compositional Semantics

- Add semantic attachments to CFG rules
- Compositional semantics
  - Parse the sentence syntactically
  - Associate some semantics to each word
  - Combine the semantics of words and non-terminals recursively
  - Until the root of the sentence
Example

• Input
  – Javier likes pizza

• Output
  – like(Javier, pizza)
Example

S  ->  NP  VP  {VP.Sem(NP.Sem)}  t
VP  ->  V  NP  {V.Sem(NP.Sem)}  <e,t>
NP  ->  N  {N.Sem}  e
V  ->  likes  {λ x,y likes(x,y)}  <e,<e,t>>
N  ->  Javier  {Javier}  e
N  ->  pizza  {pizza}  e
Semantic Parsing

- Associate a semantic expression with each node

```
S: likes(Javier, pizza)
```

```
N: Javier
```

```
V: \( \lambda x, y \) likes(x, y)
```

```
N: pizza
```

```
VP: \( \lambda x \) likes(x, pizza)
```

```
Javier
```

```
likes
```

```
pizza
```
# Grammar with Semantic Attachments

<table>
<thead>
<tr>
<th>Grammar Rule</th>
<th>Semantic Attachment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow NP \ VP$</td>
<td>${NP.sem(VP.sem)}$</td>
</tr>
<tr>
<td>$NP \rightarrow Det \ Nominal$</td>
<td>${Det.sem(Nominal.sem)}$</td>
</tr>
<tr>
<td>$NP \rightarrow ProperNoun$</td>
<td>${ProperNoun.sem}$</td>
</tr>
<tr>
<td>Nominal $\rightarrow Noun$</td>
<td>${Noun.sem}$</td>
</tr>
<tr>
<td>$VP \rightarrow Verb$</td>
<td>${Verb.sem}$</td>
</tr>
<tr>
<td>$VP \rightarrow Verb \ NP$</td>
<td>${Verb.sem(NP.sem)}$</td>
</tr>
<tr>
<td>$Det \rightarrow every$</td>
<td>${\lambda P.\lambda Q.\forall x P(x) \Rightarrow Q(x)}$</td>
</tr>
<tr>
<td>$Det \rightarrow a$</td>
<td>${\lambda P.\lambda Q.\exists x P(x) \land Q(x)}$</td>
</tr>
<tr>
<td>$Noun \rightarrow restaurant$</td>
<td>${\lambda r.Restaurant(r)}$</td>
</tr>
<tr>
<td>$ProperNoun \rightarrow Matthew$</td>
<td>${\lambda m.m(Matthew)}$</td>
</tr>
<tr>
<td>$ProperNoun \rightarrow Franco$</td>
<td>${\lambda f.f(Franco)}$</td>
</tr>
<tr>
<td>$ProperNoun \rightarrow Frasca$</td>
<td>${\lambda f.f(Frasca)}$</td>
</tr>
<tr>
<td>$Verb \rightarrow closed$</td>
<td>${\lambda x.\exists e Closed(e) \land ClosedThing(e,x)}$</td>
</tr>
<tr>
<td>$Verb \rightarrow opened$</td>
<td>${\lambda w.\lambda z.w(\lambda x.\exists e Opened(e) \land Opener(e,z) \land Opened(e,x))}$</td>
</tr>
</tbody>
</table>

Example from Jurafsky and Martin
Using CCG (Steedman 1996)

- CCG representations for semantics
  - $\text{ADJ}: \lambda x.\text{tall}(x)$
  - $(S\backslash \text{NP})/\text{ADJ} : \lambda f.\lambda x.f(x)$
  - $\text{NP}: \text{YaoMing}$

$$
\begin{array}{ccc}
\text{YaoMing} & \text{is} & \text{tall} \\
\text{NP} & (S\backslash \text{NP})/\text{ADJ} & \text{ADJ} \\
\text{YaoMing} & \lambda f.\lambda x.f(x) & \lambda x.\text{tall}(x) \\
\end{array}
$$

$S\backslash \text{NP}$

$\lambda x.\text{tall}(x)$

$S$

$\text{Tall (YaoMing)}$
CCG Parsing

• Example:
  – https://bitbucket.org/yoavartzi/spf

• Tutorial by Artzi, FitzGerald, Zettlemoyer
What is the capital of the state with the largest population?
answer(C, (capital(S,C), largest(P, (state(S), population(S,P))))).

What are the major cities in Kansas?
answer(C, (major(C), city(C), loc(C,S), equal(S,stateid(kansas)))).

<table>
<thead>
<tr>
<th>Type</th>
<th>Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>country</td>
<td>countryid(Name)</td>
<td>countryid(usa)</td>
</tr>
<tr>
<td>city</td>
<td>cityid(Name, State)</td>
<td>cityid(austin,tx)</td>
</tr>
<tr>
<td>state</td>
<td>stateid(Name)</td>
<td>stateid(texas)</td>
</tr>
<tr>
<td>river</td>
<td>riverid(Name)</td>
<td>riverid(colorado)</td>
</tr>
<tr>
<td>place</td>
<td>placeid(Name)</td>
<td>placeid(pacific)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Form</th>
<th>Predicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>capital(C)</td>
<td>C is a capital (city).</td>
</tr>
<tr>
<td>city(C)</td>
<td>C is a city.</td>
</tr>
<tr>
<td>major(X)</td>
<td>X is major.</td>
</tr>
<tr>
<td>place(P)</td>
<td>P is a place.</td>
</tr>
<tr>
<td>river(R)</td>
<td>R is a river.</td>
</tr>
<tr>
<td>state(S)</td>
<td>S is a state.</td>
</tr>
<tr>
<td>capital(S,C)</td>
<td>The capital of S is C.</td>
</tr>
<tr>
<td>area(S,A)</td>
<td>The area of S is A.</td>
</tr>
<tr>
<td>equal(V,C)</td>
<td>variable V is ground term C.</td>
</tr>
<tr>
<td>density(S,D)</td>
<td>The (population) density of S is P</td>
</tr>
<tr>
<td>elevation(P,E)</td>
<td>The elevation of P is E.</td>
</tr>
<tr>
<td>high_point(S,P)</td>
<td>The highest point of S is P.</td>
</tr>
<tr>
<td>higher(P1,P2)</td>
<td>P1’s elevation is greater than P2’s.</td>
</tr>
<tr>
<td>loc(X,Y)</td>
<td>X is located in Y.</td>
</tr>
<tr>
<td>low_point(S,P)</td>
<td>The lowest point of S is P.</td>
</tr>
<tr>
<td>len(R,L)</td>
<td>The length of R is L.</td>
</tr>
<tr>
<td>next_to(S1,S2)</td>
<td>S1 is next to S2.</td>
</tr>
<tr>
<td>size(X,Y)</td>
<td>The size of X is Y.</td>
</tr>
<tr>
<td>traverse(R,S)</td>
<td>R traverses S.</td>
</tr>
</tbody>
</table>
Zettlemoyer and Collins (2005)

a) What states border Texas
   \[ \lambda x. \text{state}(x) \land \text{borders}(x, \text{texas}) \]

b) What is the largest state
   \[ \text{arg max}(\lambda x. \text{state}(x), \lambda x. \text{size}(x)) \]

c) What states border the state that borders the most states
   \[ \lambda x. \text{state}(x) \land \text{borders}(x, \text{arg max}(\lambda y. \text{state}(y),
   \lambda y. \text{count}(\lambda z. \text{state}(z) \land \text{borders}(y, z)))) \]

---

a) \[
\begin{array}{ccc}
\text{Utah} & \text{borders} & \text{Idaho} \\
NP & (S\setminus NP)/NP & NP \\
\text{utah} & \lambda x. \lambda y. \text{borders}(y, x) & \text{idaho} \\
\rightarrow & (S\setminus NP) & \lambda y. \text{borders}(y, \text{idaho}) \\
\rightarrow & S & \text{borders(utah, idaho)}
\end{array}
\]

b) \[
\begin{array}{cccc}
\text{What} & \text{states} & \text{border} & \text{Texas} \\
(S/(S\setminus NP))/N & \lambda x. \text{state}(x) & (S\setminus NP)/NP & NP \\
\lambda f. \lambda g. \lambda x. f(x) \land g(x) & \lambda x. \text{state}(x) & \lambda y. \text{borders}(y, x) & \text{texas} \\
\rightarrow & S/(S\setminus NP) & \lambda g. \lambda x. \text{state}(x) \land g(x) & \rightarrow \\
& \lambda x. \text{state}(x) & \lambda y. \text{borders}(y, \text{texas}) & \rightarrow \\
& S & \text{borders}(x, \text{texas}) & \rightarrow
\end{array}
\]
Zettlemoyer and Collins (2005)

\[
\begin{align*}
\text{states} & : = \ N : \lambda x.\text{state}(x) \\
\text{major} & : = \ N/N : \lambda f.\lambda x.\text{major}(x) \land f(x) \\
\text{population} & : = \ N : \lambda x.\text{population}(x) \\
\text{cities} & : = \ N : \lambda x.\text{city}(x) \\
\text{rivers} & : = \ N : \lambda x.\text{river}(x) \\
\text{run through} & : = \ (S\setminus NP)/NP : \lambda x.\lambda y.\text{traverse}(y, x) \\
\text{the largest} & : = \ NP/N : \lambda f.\ \text{arg max}(f, \lambda x.\text{size}(x)) \\
\text{river} & : = \ N : \lambda x.\text{river}(x) \\
\text{the highest} & : = \ NP/N : \lambda f.\ \text{arg max}(f, \lambda x.\text{elev}(x)) \\
\text{the longest} & : = \ NP/N : \lambda f.\ \text{arg max}(f, \lambda x.\text{len}(x)) \\
\end{align*}
\]

**Figure 6:** Ten learned lexical items that had highest associated parameter values from a randomly chosen development run in the Geo880 domain.
Figure 1: Input utterances and their logical forms are encoded and decoded with neural networks. An attention layer is used to learn soft alignments.

Figure 3: Sequence-to-tree (SEQ2TREE) model with a hierarchical tree decoder.
Dong and Lapata (2016)

**JOBS** This benchmark dataset contains 640 queries to a database of job listings. Specifically, questions are paired with Prolog-style queries. We used the same training-test split as Zettlemoyer and Collins (2005) which contains 500 training and 140 test instances. Values for the variables company, degree, language, platform, location, job area, and number are identified.

**GEO** This is a standard semantic parsing benchmark which contains 880 queries to a database of U.S. geography. GEO has 880 instances split into a training set of 680 training examples and 200 test examples (Zettlemoyer and Collins, 2005). We used the same meaning representation based on lambda-calculus as Kwiatkowski et al. (2011). Values for the variables city, state, country, river, and number are identified.

**ATIS** This dataset has 5,410 queries to a flight booking system. The standard split has 4,480 training instances, 480 development instances, and 450 test instances. Sentences are paired with lambda-calculus expressions. Values for the variables date, time, city, aircraft code, airport, airline, and number are identified.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Length</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBS</td>
<td>9.80</td>
<td>what microsoft jobs do not require a bscs?</td>
</tr>
<tr>
<td></td>
<td>22.90</td>
<td>answer(company(J,’microsoft’),job(J),not((req_deg(J,’bscs’))))</td>
</tr>
<tr>
<td>GEO</td>
<td>7.60</td>
<td>what is the population of the state with the largest area?</td>
</tr>
<tr>
<td></td>
<td>19.10</td>
<td>(population:i (argmax S0 (state:t $0) (area:i S0)))</td>
</tr>
<tr>
<td>ATIS</td>
<td>11.10</td>
<td>dallas to san francisco leaving after 4 in the afternoon please</td>
</tr>
<tr>
<td></td>
<td>28.10</td>
<td>(lambda S0 e (and (&gt; (departure_time S0) 1600:ti) (from S0 dallas:ci) (to S0 san francisco:ci)))</td>
</tr>
<tr>
<td>IFTTT</td>
<td>6.95</td>
<td>Turn on heater when temperature drops below 58 degree</td>
</tr>
<tr>
<td></td>
<td>21.80</td>
<td>TRIGGER: Weather - Current_temperature_drops_below - ((Temperature (58)) (Degrees_in (f))) ACTION: WeMo_Insight_Switch - Turn_on - ((Which_switch? (&quot;&quot;)))</td>
</tr>
</tbody>
</table>

Table 1: Examples of natural language descriptions and their meaning representations from four datasets. The average length of input and output sequences is shown in the second column.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>COCKTAIL (Tang and Mooney, 2001)</td>
<td>79.4</td>
</tr>
<tr>
<td>PRECISE (Popescu et al., 2003)</td>
<td>88.0</td>
</tr>
<tr>
<td>ZC05 (Zettlemoyer and Collins, 2005)</td>
<td>79.3</td>
</tr>
<tr>
<td>DCS+L (Liang et al., 2013)</td>
<td>90.7</td>
</tr>
<tr>
<td>TISP (Zhao and Huang, 2015)</td>
<td>85.0</td>
</tr>
<tr>
<td>SEQ2SEQ</td>
<td>87.1</td>
</tr>
<tr>
<td></td>
<td>77.9</td>
</tr>
<tr>
<td></td>
<td>70.7</td>
</tr>
<tr>
<td>SEQ2TREE</td>
<td>90.0</td>
</tr>
<tr>
<td></td>
<td>83.6</td>
</tr>
</tbody>
</table>

Table 2: Evaluation results on JOBS.
Figure 6: Alignments (same color rectangles) produced by the attention mechanism (darker color represents higher attention score). Input sentences are reversed and stemmed. Model output is shown for SEQ2SEQ (a, b) and SEQ2TREE (c, d).
FrameNet

• Represents
  – Events, relations, states, entities

• 1,195 semantic frames

• Example: Absorb_heat
  – An Entity (generally food) is exposed to a Heat_source whose Temperature may also be specified. Generally, the Entity undergoes some sort of change as a result of this process.
    • Bacon was frying in the pan, and a great heap of eggs already lay steaming on a plate.
    • If it cooks at 400 for an hour, it 'll be nothing but a pile of ash!

• 1,774 frame-to-frame relations

• Links
  – https://framenet.icsi.berkeley.edu/fndrupal/
Assistance

Definition:
A **Helper** benefits a **Benefited_party** by enabling the culmination of a **Goal** that the **Benefited_party** has. A **Focal_entry** that is involved in reaching the **Goal** may stand in for it. This Frame has unique Frame-to-Frame Relations: it inherits from and uses the Intentionality_act frame. This is atypical of FrameNet's frame relations.

Will you HELP the Government find your brother?

Maybe Stephen should ASSIST him with the last manuscript.

The Helper HELPED me psychologically to overcome the physical loss I had suffered.

You have HELPED him tremendously by showing him how to stand up for himself and by being his friend.

By bringing assistance to his troops wherever they were in trouble, he AIDED them greatly.

**FES:**

**Core:**

**Benefited_party [Ben]**

The **Benefited_party** receives a benefit from the action of the **Helper**.

**Focal_entry [Focal_e]**

This FE identifies a **Focal_entry** involved in achieving the **Goal**.

Whoever didn't cook has to HELP with the dishes.

**Goal [Goal]**

The desirable state of affairs that the **Benefited_party** is involved in and which is enabled by the **Helper**.

Jack HELPED Jim climb Mt. Everest.

**Helper [Help]**

The **Helper** performs some action that benefits the **Benefited_party**.

**Non-Core:**

**Degree [deg]**

The measure to which the **Helper's** assistance brings the **Benefited_party** closer to accomplishing their **Goal**.

The Algebra Buster software HELPED my daughter very much.

**Semantic Type: Degree**

**Domain [dom]**

The aspect of accomplishing the **Goal** to which the **Helper's** assistance is relevant.
Abstract Meaning Representation (AMR)

- http://amr.isi.edu/
- Single structure that includes:
  - Predicate–Argument Structure
  - Named Entity Recognition
  - Coreference Resolution
  - Wikification

[slide from Jonathan Kummerfeld]
Example

“Lassie ate four bones that she found.”
About 14,000 people fled their homes at the weekend after a local tsunami warning was issued, the UN said on its Web site.
Status of AMR

- AMR currently lacks
  - Multilingual consideration
  - Quantifier scope
  - Co-references across sentences
  - Grammatical number, tense, aspect, quotation marks
  - Many noun-noun or noun-adjective relations
  - Many detailed frames, e.g. Earthquake (with roles for magnitude, epicenter, casualties, etc)

[slide from Jonathan Kummerfeld]
AMR Parsing (Wang et al. 2015,16)

(a) Dependency tree

(b) AMR graph

Figure 1: Dependency tree and AMR graph for the sentence, “The police want to arrest Micheal Karras in Singapore.”
### AMR Parsing (Wang et al. 2015, 2016)

<table>
<thead>
<tr>
<th>Action</th>
<th>Current state ⇒ Result state</th>
<th>Assign labels</th>
<th>Precondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEXT EDGE-(l_r)</td>
<td>((\sigma_0</td>
<td>\sigma'</td>
<td>, \beta_0</td>
</tr>
<tr>
<td>SWAP-(l_r)</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G') \Rightarrow (\sigma_0</td>
</tr>
<tr>
<td>REATTACH(_k)-(l_r)</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G') \Rightarrow (\sigma_0</td>
</tr>
<tr>
<td>REPLACE HEAD</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G') \Rightarrow (\sigma_0</td>
</tr>
<tr>
<td>REENTRANCE(_k)-(l_r)</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G') \Rightarrow (\sigma_0</td>
</tr>
<tr>
<td>MERGE</td>
<td>((\sigma_0</td>
<td>\sigma', \beta_0</td>
<td>\beta', G') \Rightarrow (\tilde{\sigma}</td>
</tr>
<tr>
<td>NEXT NODE-(l_c)</td>
<td>((\sigma_0</td>
<td>\sigma_1</td>
<td>\sigma', [], G) \Rightarrow (\sigma_1</td>
</tr>
<tr>
<td>DELETE NODE</td>
<td>((\sigma_0</td>
<td>\sigma_1</td>
<td>\sigma', [], G) \Rightarrow (\sigma_1</td>
</tr>
</tbody>
</table>

Table 1: Transitions designed in our parser. \(CH(x, y)\) means getting all node \(x\)’s children in graph \(y\).
AMR Parsing (Wang et al. 2015,16)

Figure 4: SWAP action

Figure 5: REATTACH action

Figure 6: REPLACE-HEAD action

Figure 8: MERGE action