Driving Semantic Parsing from World's Supervision

Ming Wei Chang, James Clarke, *Dan Goldwasser* and Dan Roth

ISCOL 2010

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

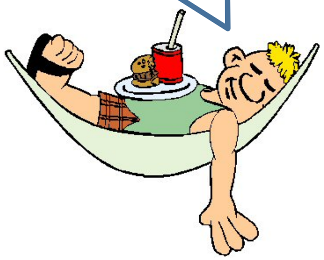
COGNITIVE COMPUTATION GROUP
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Mapping Language to Formal Meaning

what we would really like computers to do with NL inputs

Get us coffee!

"Can I get a coffee with no sugar and just a bit of milk "



MAKE(COFFEE,SUGAR=NO,MILK=LITTLE)

Getting coffee requires moving **from NL to a formal language**

➔ This process is commonly known as ***Semantic Parsing***

Current systems learn this mapping from **annotated data**

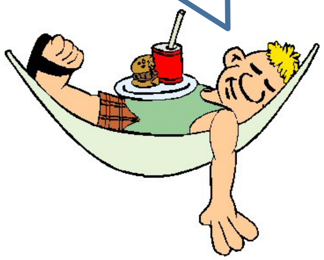
➔ pairs of ***(Sentence, Logical Form)***

➔ Requires **a lot** of training data – *"cup of hot java"*

Connecting Language to the World

*Interpretation is used to trigger a **response***

"Can I get a coffee with no sugar and just a bit of milk "



MAKE(COFFEE,SUGAR=NO,MILK=LITTLE)

Which we might **like** .. or **not**

Great!

Arggg

This is useful information!

➔ Can we rely on this interaction?

- **➔ Indirect, Response-based Learning protocols**
  - Exploit external supervision signal (no annotated data)
- **➔ Semantic parsing model**
  - Adapted for weak supervision

Outline

- **Semantic Parsing 101**
  - Basic definitions
- **Response based Learning**
  - DIRECT learning protocol
  - AGGRESSIVE learning protocols
- **Semantic Parsing Model**
  - Local decisions and global inference
- **Empirical Evaluation**
- **Conclusions and Further Steps**
  - Shameless promotion (take a look at our ICML paper!)

COGNITIVE COMPUTATION GROUP
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Semantic Parsing

- **Mapping NL to formal Meaning Representation (MR)**
  - Typically applied to NLDB access applications (e.g., GEOQUERY)
- **Use a subset of FOL to describe domain's semantics**
  - **Constants**: const(NY), const(NYC), const(Hudson_river)
  - **Functions**: state(x), city(x), river(x), loc(x)
  - **Complex Formulas**: "cities in NY" → city(loc (const(NY)))
- **Current works: map syntactic patterns to logical forms**
  - **Rule based mapping** [NP] [const(NY)] → "*NY*"
    - Rules involve both semantic and syntactic categories
  - **Early Systems** (e.g., Winograd'72): rule based, manually
  - **Recent works** use ML to extract and parameterize rules

Semantic Parsing

X	**Input**	*"What is the largest state that borders NY?"*
Z	**Output**	`largest( state( next_to( const(NY))))`

Formal definition: $F : X \to Z$

- **A high level task requiring many "small decisions"**
  - Which entities appear in the interpretation?
    - "***NY***" refers to the *state* or to the *city*?
  - How to compose the meaning from the fragments?
    - $state(next\_to()) \neq next\_to(state())$
- **Interdependency** between decisions
  - E.g., $state(NYC)$ is not very likely..

Cognitive Computation Group
University of Illinois at Urbana-Champaign

Semantic Parsing as Structure Prediction

x	**Input**	*"What is the largest state that borders NY?"*
y	**Hidden**	
z	**Output**	`largest( state( next_to( const(NY))))`

- a ***hidden structure prediction*** problem
  - Decompose the prediction into a set of decisions defined over segments of the input text
    - E.g., "*is this word span mapped to this logical symbol*?"
  - **Structured output** (z) : output composed of many decisions
  - **Hidden** (y) : segmentation and mapping is unknown
- **Predicted structure**: optimal *global* structure

$$z^* = F_w(x) = \arg\max_{y \in Y} score(x, y) = \arg\max_{y \in Y, z \in Z} w^T \Phi(x, y, z)$$

Learning Semantic Parsers

x	**Input**	*"What is the largest state that borders NY?"*
y	**Hidden**	
z	**Output**	`largest( state( next_to( const(NY))))`

- **Structure Learning Objective** : find a parameter set (W) that *minimizes the structured loss (i.e., distance) over **gold data***

$$\arg\min_{w} \sum_{i} loss(x_i, z_i, w)$$

- Current learning paradigms for semantic parsers:
  - **Fully supervised**: trained with labeled data: (x,y,z) triplets
  - ***"Hidden structure "***: trained with missing data: (x,z) pairs
- **Either way – lots of work..**

Response based Learning

x **Input** *"What is the largest state that borders NY?"*

y **Hidden**

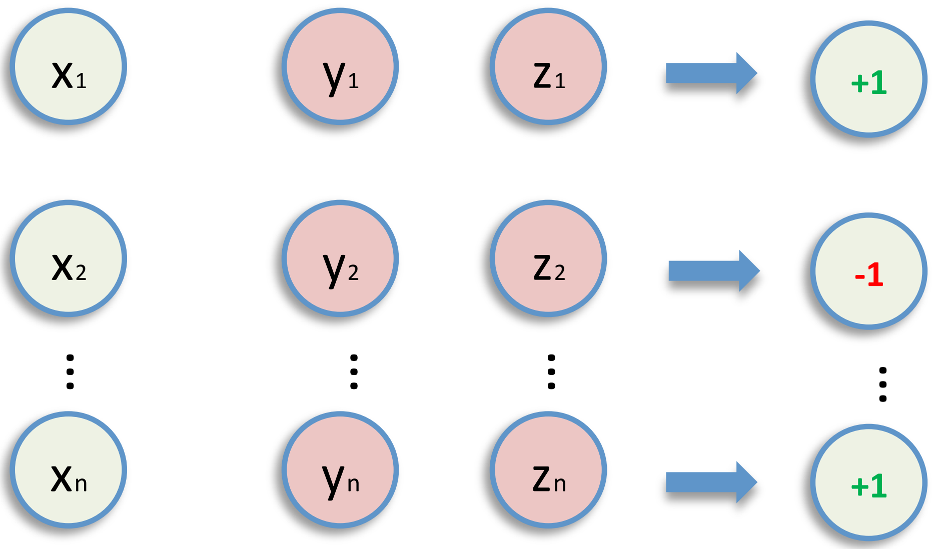
z **Output** `largest( state( next_to( const(NY))))`

- Use the ***expected response*** as supervision
  - Learning is based on a *feedback function*
    - *Is the **predicted** structure correct?*

$$Feedback(z,r) = \begin{cases} 1 & \text{if } execute(z) = r \\ -1 & \text{otherwise} \end{cases}$$

Response based Learning Protocols

Training Semantic Parser from World's Response!

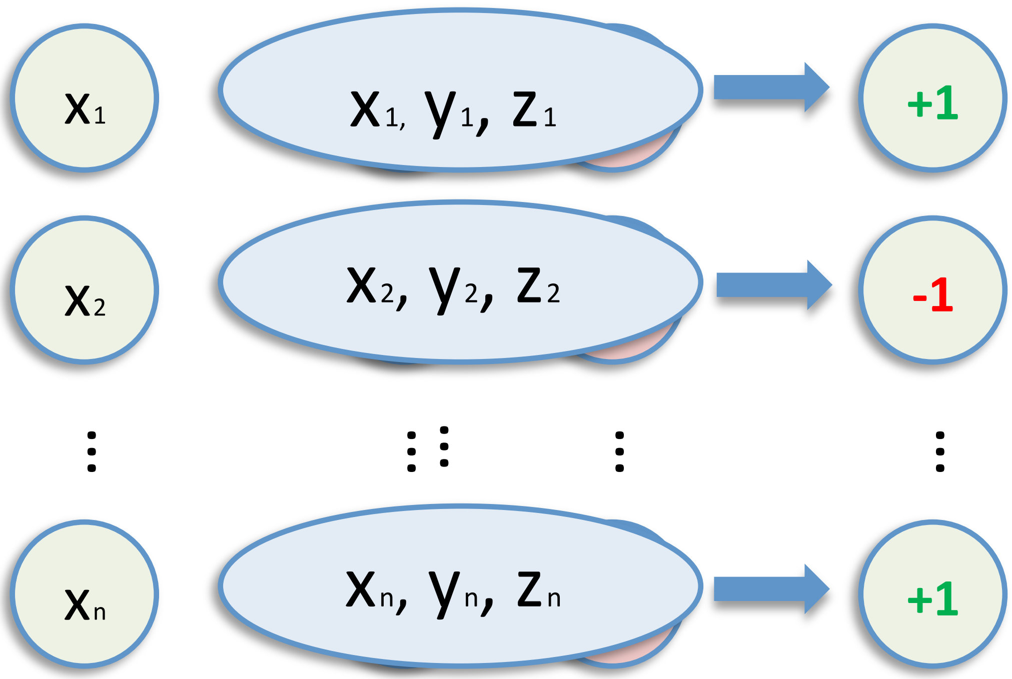


Repeat
  for all input sentences **do**
    Find best output
    Query *feedback* function
  end for
  Learn new W using feedback
Until Convergence

- **Bridging the gap**: ***Structure Learning with Binary feedback***
  - DIRECT protocol: Convert the learning problem into binary prediction
  - AGGRESSIVE protocol: Convert the feedback into structured supervision
- **Learning approach** – iteratively identify more correct structures
  - Learning terminates when no new structures are added

DIRECT Approach

Learn a binary classifier to discriminate between good and bad meaning representations

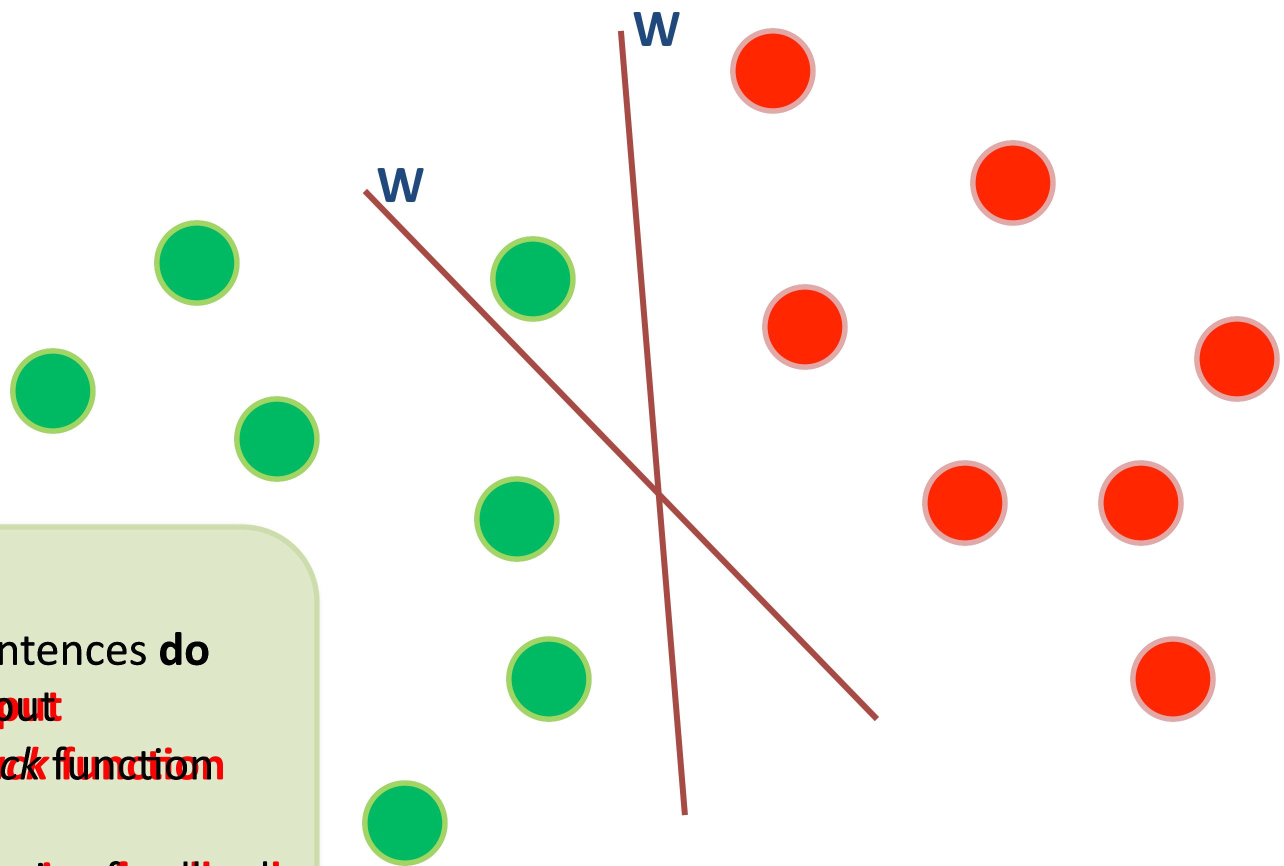


Treat (x,y,z) as a **binary** sample
➔ Labels provided by feedback function

Learning – Find a W such that

$$f(z,r)\cdot w^T\Phi(x,y,z) > 0$$

Geometric Interpretation



Repeat
  for all input sentences **do**
    Find best output
    Query *feedback* function
  end for
  Learn new W using feedback
Until Convergence

Cognitive Computation Group
University of Illinois at Urbana-Champaign

AGGRESSIVE Approach

Trains a structure predictor

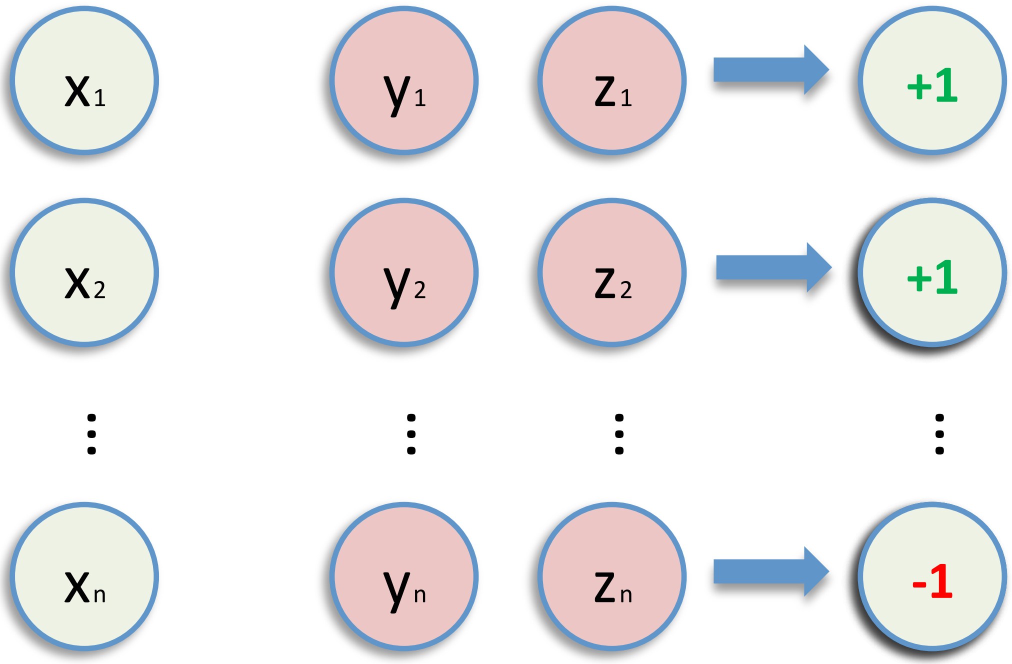
- The goal of structure learning:
  - Learning a scoring function for structures, s.t.-

$$w^T \Phi(x, y, z^*) > w^T \Phi(x, y, z')$$

- A structure learner needs positive examples
  - *Can only use the positive feedback*

AGGRESSIVE Approach

Positive feedback is a good indicator of correct structure



Repeat
  for all input sentences **do**
    Find best output
    Query *feedback* function
  end for
  Learn new W using feedback
Until Convergence

Cognitive Computation Group
University of Illinois at Urbana-Champaign

Semantic Parsing Model

x	**Input**	*"What is the largest state that borders NY?"*
y	**Hidden**	
z	**Output**	`largest( state( next_to( const(NY))))`

$$z^* = F_w(x) = \arg\max_{y \in Y, z \in Z} w^T \Phi(x, y, z)$$

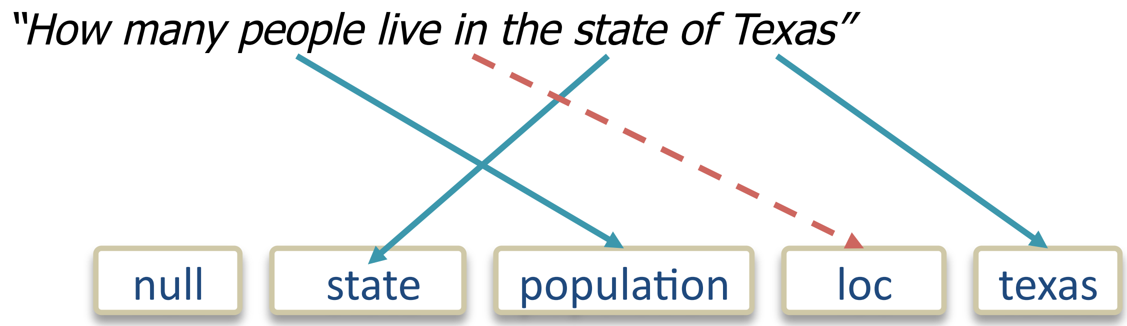
So far.. **..And now**

- Decompose into two types of decisions:
  - **First order:** Map lexical items to logical symbols
    - {*"largest"*→ `largest()`, *"borders"*→`next_to()`,.., *"NY"*→`const(NY)`}
  - **Second order:** Compose meaning from logical fragments
    - `largest(state(next_to(const(NY))))`

COGNITIVE COMPUTATION GROUP
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

First Order Decisions

Learn to map lexical items to logical symbols



- Bootstrap the process with a simple lexicon
- Use lexical resources to extend it
  - `Wordnet(people, population)`
- Use context for disambiguation
  - *What is the longest river* ***in*** *Texas?*
  - *How high is NY* ***in*** *meters?*

```
> texas
Texas
> state
state
>population
population
>loc
in
```

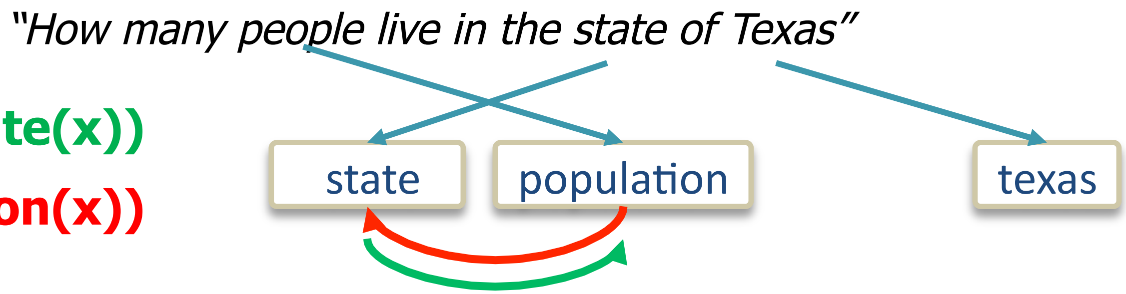
Cognitive Computation Group
University of Illinois at Urbana-Champaign

Second Order Decisions

Compose logical fragments into meaning

population(state(x))

state(population(x))



- Use domain's semantics to restrict inference:
  - constrain possible interpretations according to the domain
    - E.g., type consistency: *state(population(x))*
- Features:
  - Distance between originating words
    - Dependency path distance
    - Word position distance
  - Predicate bigrams
    - Captures frequent constructions

Cognitive Computation Group
University of Illinois at Urbana-Champaign

Inference

$$z^* = F_w(x) = \arg\max_{y \in Y, z \in Z} w^T \Phi(x, y, z)$$

..And now

- Two types of decisions, but **joint inference**
  - Decisions are **not** pipelined ➔ Global optimum
- Formulated as an **Integer Linear Program**
  - **Declarative**: allows encoding domain semantics
    - Type constraints, syntactic constraints

Empirical Evaluation

- Research Questions:
  - **Can a semantic parser be learned from 'weak' supervision?**
  - **Learning**: How does each algorithm utilize the binary signal?
  - **Parsing Model**: comparison with existing semantic parsers

- Experimental setup (GEOQUERY):
  - Training set (with binary signal): 250 queries
  - Testing set (no supervision) : 250 queries

Results: Learning Protocols

Algorithm	Training Accuracy	Testing Accuracy
NoLearn	22	--
Direct	75.2	69.2
Aggressive	**82.4**	**73.2**
Supervised	87.6	80.4

NoLearn : Lower bound – No learning

Supervised : Upper bound – supervised data

We can train a semantic parser using Response based Learning (no labeled data!)

Results: Parsing Model

Algorithm	# training structures	Test set (no feedback)
DIRECT	0	69.2 %
AGGRESSIVE	0	73.2 %
SUPERVISED	250	80.4%
W&M 2006	310	60%
W&M 2007	310	75%
Z&C 2005	600	79.29%
W&M 2007	600	86.07%

Our light-weight model is competitive with existing models trained with more data

[W&M 2006] Y.-W. Wong and R. Mooney. 2006. Learning for semantic parsing with statistical machine translation.
[W&M 2007] Y.-W. Wong and R. Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. ACL.
[Z&C 2005] L. Zettelmoyer and M. collins. 2005. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars,

Summary

Response based Semantic Parsing

- Minimize supervision for this task
  - Allow scaling up semantic parsing
- Indirect learning protocols
  - Use external supervision source
  - Find "hidden explanations" with Binary feedback (NAACL'10)
  - Combine binary and structured feedback (ICML'10)
- Shallow representations for Semantic Parsing
  - Easier to adapt to new text

Questions?