Driving Semantic Parsing from the World's Response

<u>James Clarke</u>, Dan Goldwasser, Ming-Wei Chang, Dan Roth

Cognitive Computation Group
University of Illinois at Urbana-Champaign

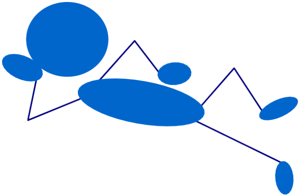
CoNLL 2010

What is Semantic Parsing?

Meaning Representation

`make(coffee, sugar=0, milk=0.3)`

I'd like a coffee with no sugar and just a little milk

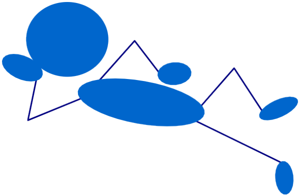


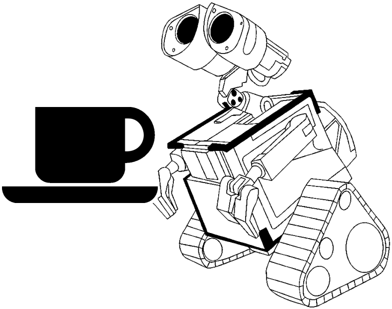
What is Semantic Parsing?

Meaning Representation

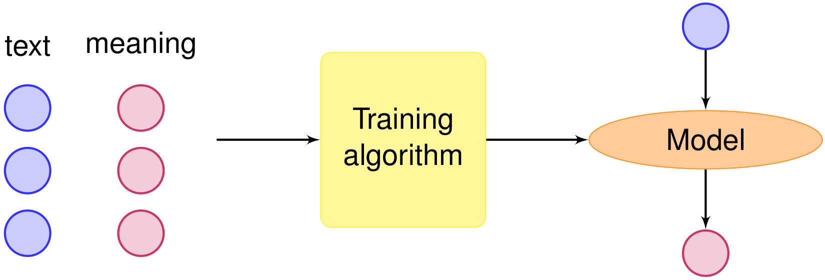
`make(coffee, sugar=0, milk=0.3)`

I'd like a coffee with no sugar and just a little milk





Supervised Learning Problem



Challenges:

- Structured Prediction problem
- Model part of the structure as hidden?

Lots of previous work

Multiple approaches to the problem:

- KRISP (Kate & Mooney 2006)
  - SVM-based parser using string kernels.
- Zettlemoyer & Collins 2005; Zettlemoyer & Collins 2007
  - Probabilistic parser based on relaxed CCG grammars.
- WASP (Wong & Mooney 2006; Wong & Mooney 2007)
  - Based on Synchronous CFG.
- Ge & Mooney 2009
  - Integrated syntactic and semantic parser.

Lots of previous work

Multiple approaches to the problem:

- KRISP (Kate & Mooney 2006)
  - SVM-based parser using string kernels.
- Zettlemoyer & Collins 2005; Zettlemoyer & Collins 2007
  - Probabilistic parser based on relaxed CCG grammars.
- WASP (Wong & Mooney 2006; Wong & Mooney 2007)
  - Based on Synchronous CFG.
- Ge & Mooney 2009
  - Integrated syntactic and semantic parser.

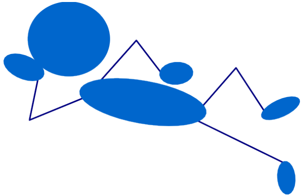
Assumption: A training set consisting of natural language and meaning representation pairs.

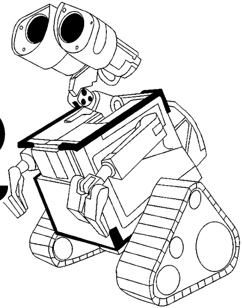
Using the World's response

Meaning Representation

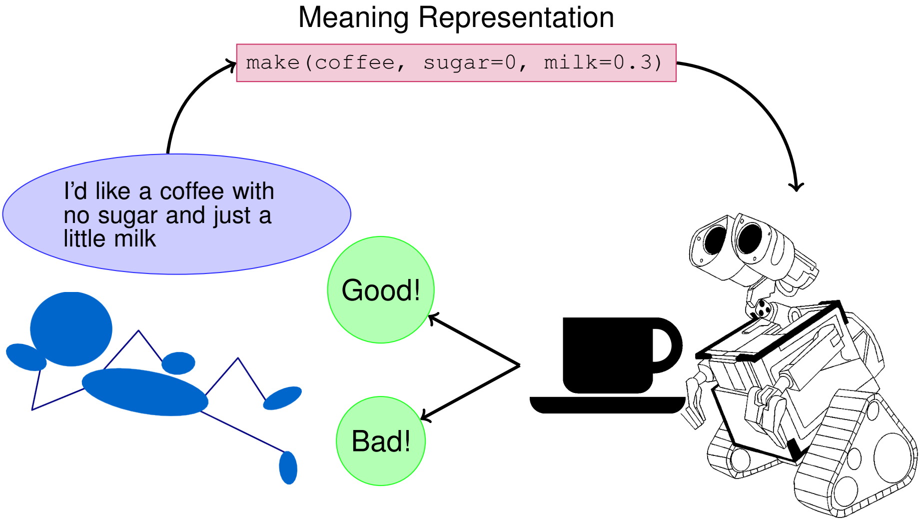
`make(coffee, sugar=0, milk=0.3)`

I'd like a coffee with no sugar and just a little milk

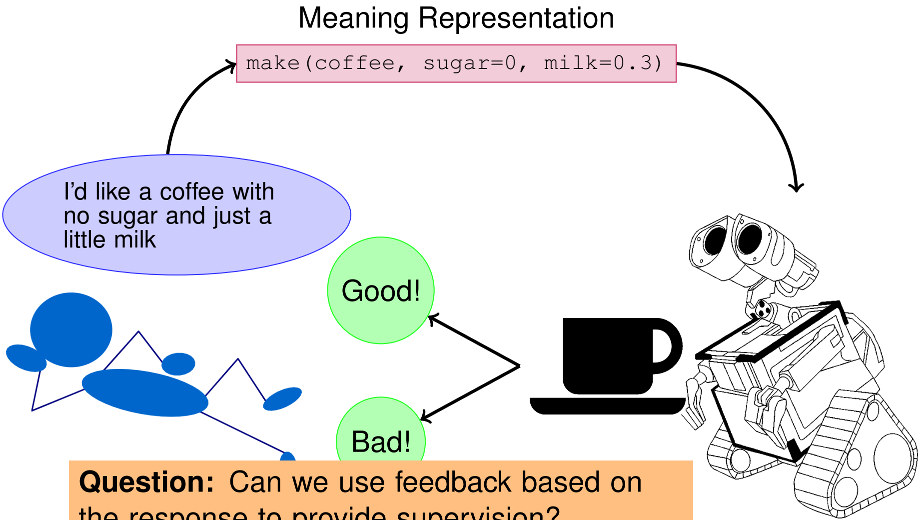




Using the World's response



Using the World's response



Question: Can we use feedback based on the response to provide supervision?

This work

We aim to:

- Reduce the burden of annotation for semantic parsing.

We focus on:

- Using the World's response to learn a semantic parser.
- Developing new training algorithms to support this learning paradigm.
- A lightweight semantic parsing model that doesn't require annotated data.

This results in:

- Learning a semantic parser using zero annotated meaning representations.

Outline

Outline

1. Semantic Parsing
2. Learning
   - DIRECT Approach
   - AGGRESSIVE Approach
3. Semantic Parsing Model
4. Experiments

Semantic Parsing

INPUT **x**

HIDDEN **y**

OUTPUT **z**

What is the largest state that borders Texas?

`largest(state(next_to(texas)))`

Semantic Parsing

INPUT $\mathbf{x}$

HIDDEN $\mathbf{y}$

OUTPUT $\mathbf{z}$

What is the largest state that borders Texas?

`largest(state(next_to(texas)))`

$$F : \mathcal{X} \to \mathcal{Z}$$

$$\hat{\mathbf{z}} = F_{\mathbf{w}}(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathcal{Y},\mathbf{z}\in\mathcal{Z}} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

Semantic Parsing

INPUT $\mathbf{x}$

HIDDEN $\mathbf{y}$

OUTPUT $\mathbf{z}$

What is the largest state that borders Texas?

`largest(state(next_to(texas)))`

$$F : \mathcal{X} \rightarrow \mathcal{Z}$$

$$\hat{\mathbf{z}} = F_{\mathbf{w}}(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}, \mathbf{z} \in \mathcal{Z}} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

- **Model** The nature of inference and feature functions.
- **Learning Strategy** How we obtain the weights.

Semantic Parsing

INPUT $\mathbf{x}$ — What is the largest state that borders Texas?

HIDDEN $\mathbf{y}$

OUTPUT $\mathbf{z}$ — `largest(state(next_to(texas)))`

Response r — New Mexico

$$F : \mathcal{X} \rightarrow \mathcal{Z}$$

$$\hat{\mathbf{z}} = F_{\mathbf{w}}(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathcal{Y},\mathbf{z}\in\mathcal{Z}} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

- **Model** The nature of inference and feature functions.
- **Learning Strategy** How we obtain the weights.

Outline

Learning

Inputs:

- Natural language sentences.
- *Feedback* : $\mathcal{X} \times \mathcal{Z} \rightarrow \{+1, -1\}$.
- Zero meaning representations.

Learning

Inputs:

- Natural language sentences.
- *Feedback* : $\mathcal{X} \times \mathcal{Z} \rightarrow \{+1, -1\}$.
- Zero meaning representations.

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

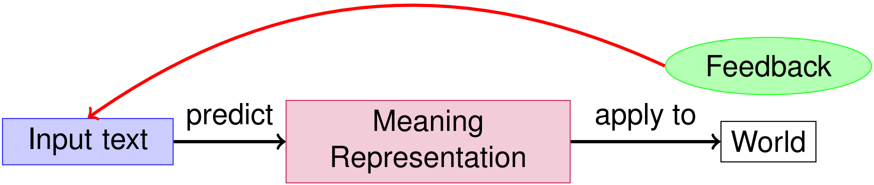
Learning

Inputs:

- Natural language sentences.
- *Feedback* : $\mathcal{X} \times \mathcal{Z} \rightarrow \{+1, -1\}$.
- Zero meaning representations.

Goal: A **weight vector** that scores the correct meaning representation higher than all other meaning representations.

Response Driven Learning:



Learning Strategies

$\mathbf{x}_1$

$\mathbf{x}_2$

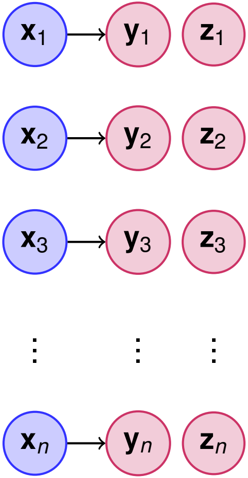
$\mathbf{x}_3$

$\vdots$

$\mathbf{x}_n$

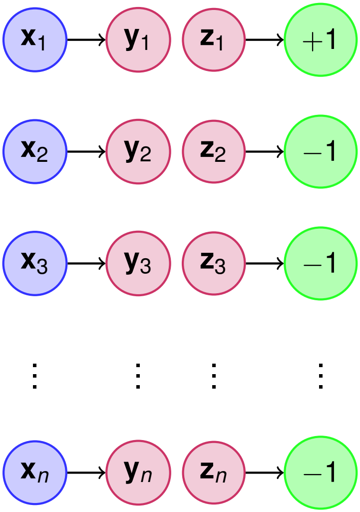
repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new **w** using feedback
until Convergence

Learning Strategies



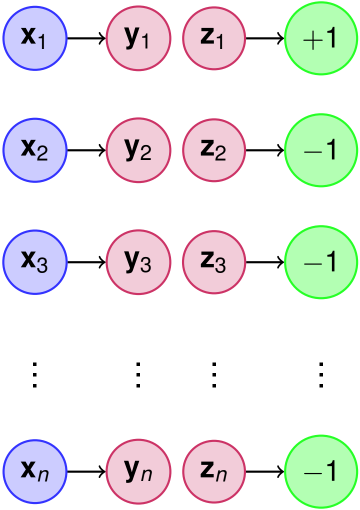
repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new **w** using feedback
until Convergence

$$\mathbf{y}, \mathbf{z} = \arg\max \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$$



repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new $\mathbf{w}$ using feedback
until Convergence

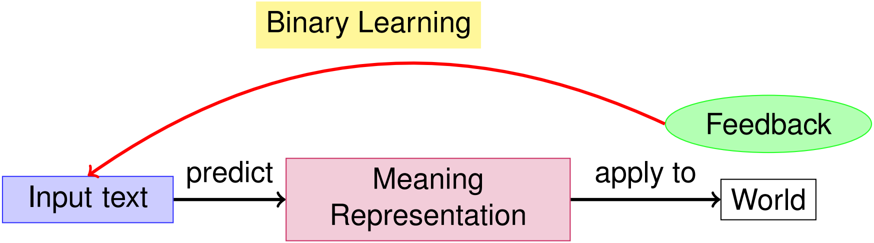
Learning Strategies



repeat
    for all input sentences **do**
        Solve the inference problem
        Query *Feedback* function
    end for
    Learn a new **w** using feedback
until Convergence

Outline

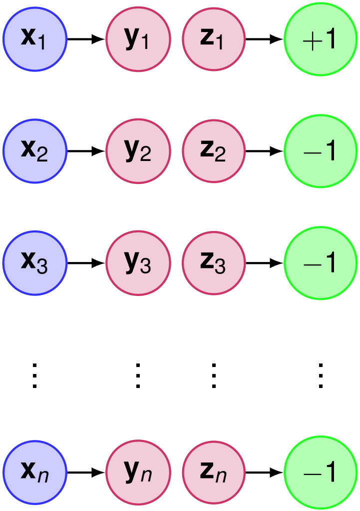
DIRECT Approach



DIRECT

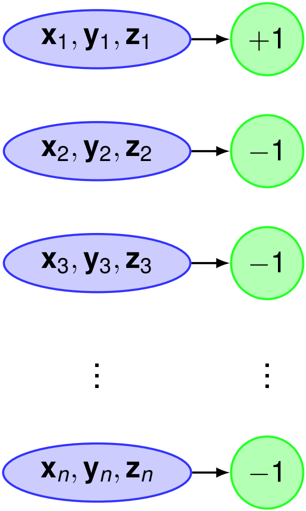
Learn a binary classifier to discriminate between good and bad meaning representations.

DIRECT Approach



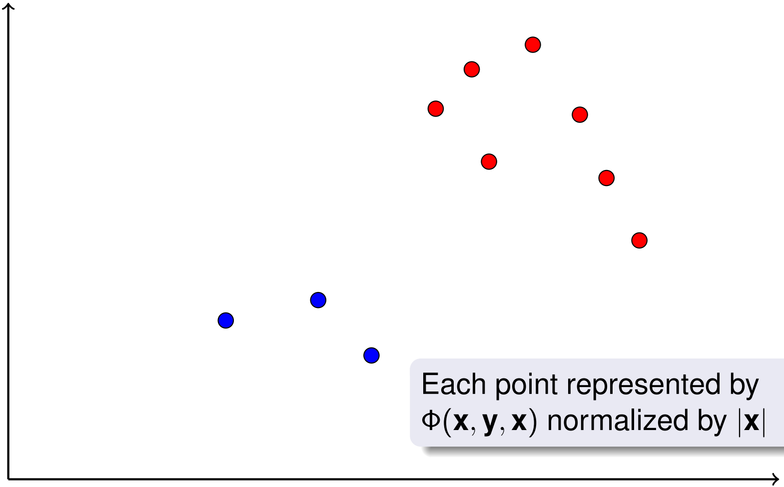
- Use $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ as a training example with label from feedback.

DIRECT Approach



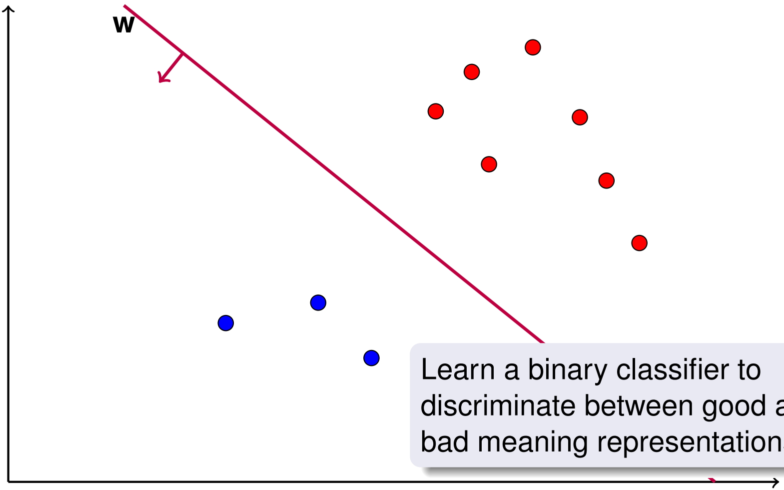
- Use $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ as a training example with label from feedback.
- Find $\mathbf{w}$ such that $f \cdot \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) > 0$

DIRECT Approach



Each point represented by $\Phi(\mathbf{x}, \mathbf{y}, \mathbf{x})$ normalized by $|\mathbf{x}|$

DIRECT Approach



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

DIRECT Approach

repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new **w** using feedback
until Convergence

DIRECT Approach

$\mathbf{x}_1$

$\mathbf{x}_2$

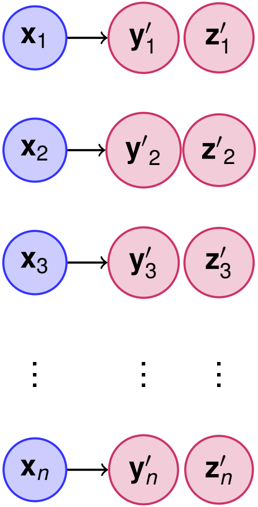
$\mathbf{x}_3$

$\vdots$

$\mathbf{x}_n$

repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new **w** using feedback
until Convergence

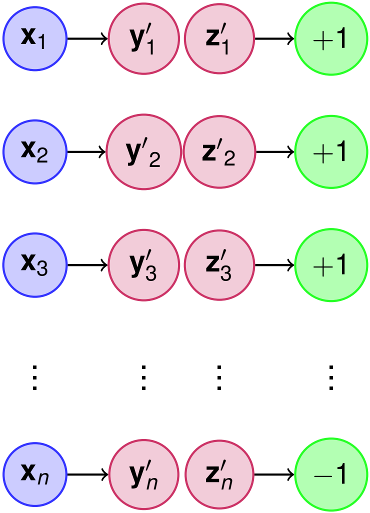
DIRECT Approach



repeat
    for all input sentences **do**
        Solve the inference problem
        Query *Feedback* function
    end for
    Learn a new **w** using feedback
until Convergence

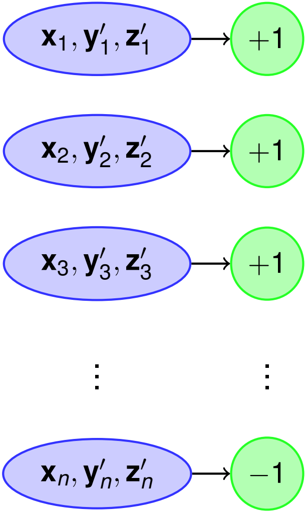
$$\mathbf{y}, \mathbf{z} = \arg\max \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

DIRECT Approach



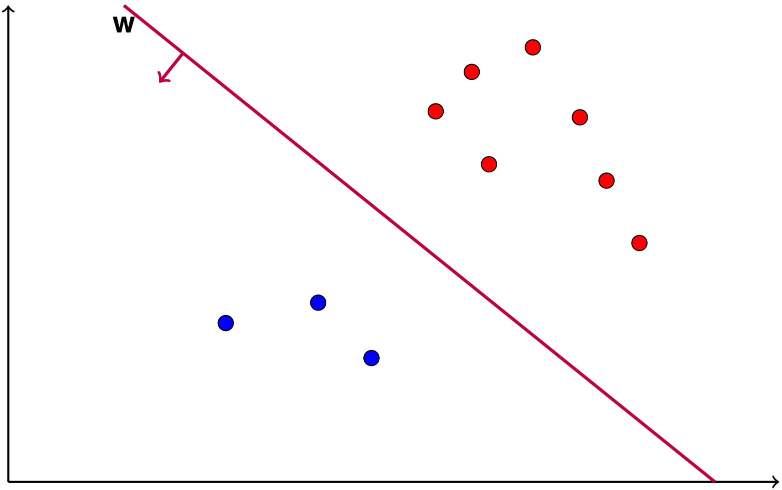
repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new **w** using feedback
until Convergence

DIRECT Approach

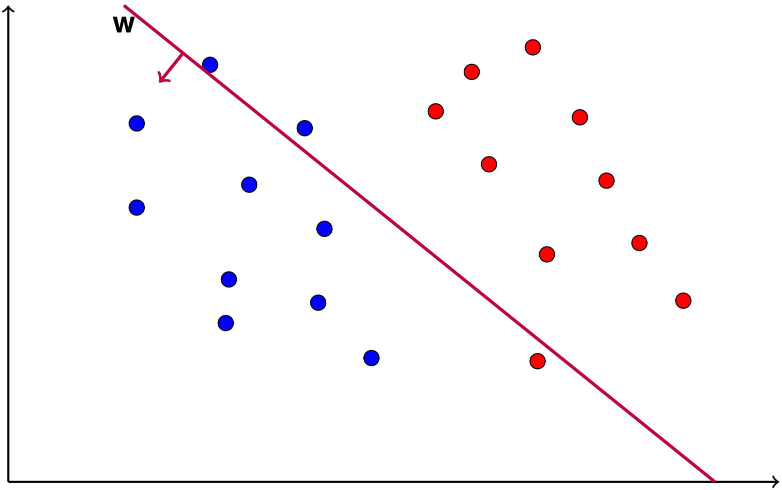


repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new **w** using feedback
until Convergence

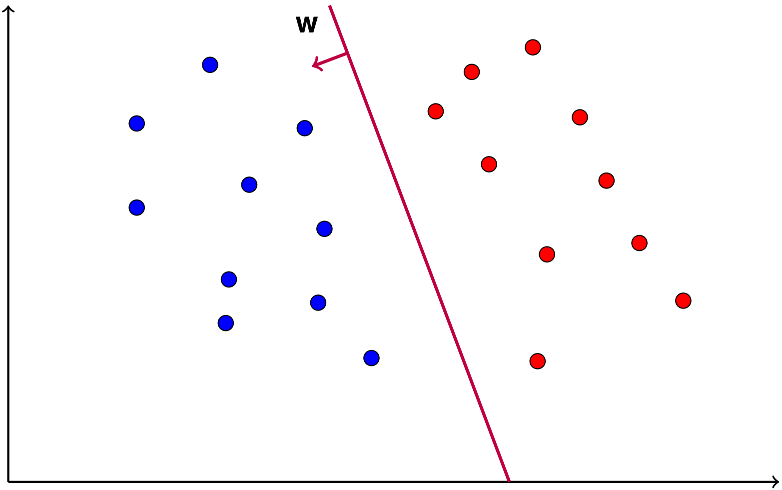
DIRECT Approach



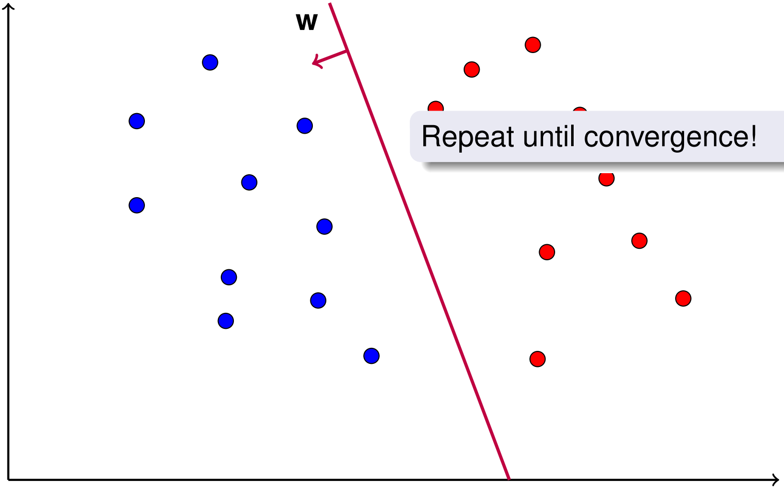
DIRECT Approach



DIRECT Approach



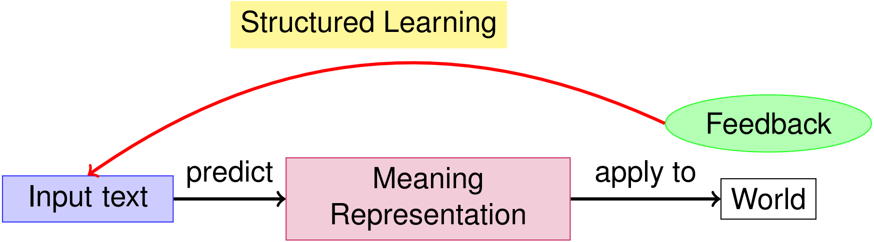
DIRECT Approach



Repeat until convergence!

Outline

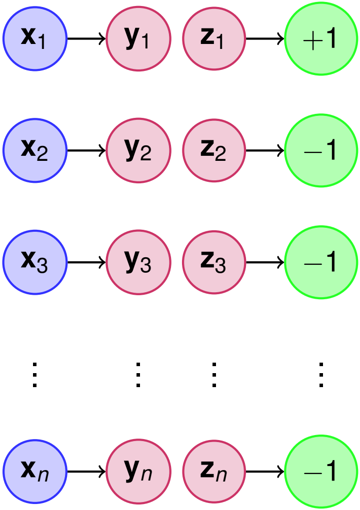
AGGRESSIVE Approach



AGGRESSIVE

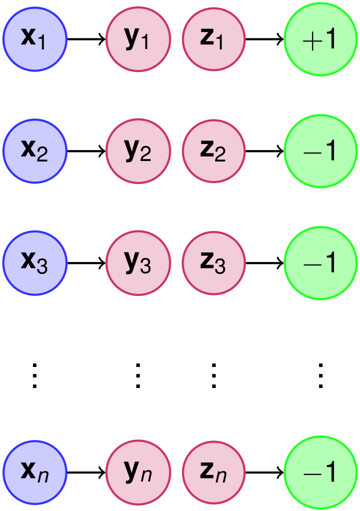
- Positive feedback is a good indicator of the correct meaning representation.
- Use data with positive feedback as training data for structured learning.

AGGRESSIVE Approach



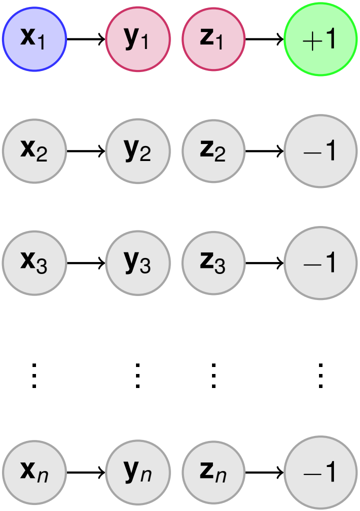
repeat
    for all input sentences **do**
        Solve the inference problem
        Query *Feedback* function
    end for
    Learn a new **w** using feedback
until Convergence

AGGRESSIVE Approach



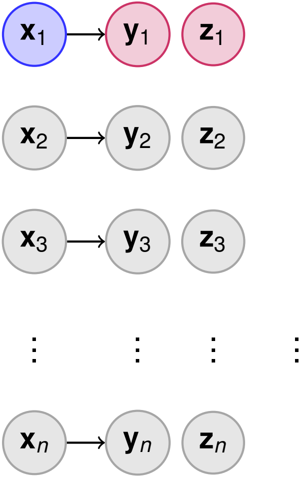
- Use items with positive feedback as training data for a structured learner.

AGGRESSIVE Approach



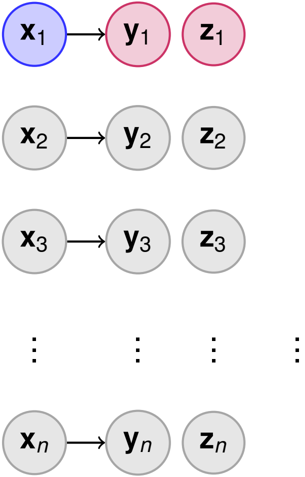
- Use items with positive feedback as training data for a structured learner.

AGGRESSIVE Approach



- Use items with positive feedback as training data for a structured learner.

AGGRESSIVE Approach



- Use items with positive feedback as training data for a structured learner.
- Implicitly consider all other meaning representations for these examples as bad.
- Find **w** such that $\mathbf{w}^T\Phi(\mathbf{x}, \mathbf{y}^*, \mathbf{z}^*) > \mathbf{w}^T\Phi(\mathbf{x}, \mathbf{y}', \mathbf{z}')$

AGGRESSIVE Approach

repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new **w** using feedback
until Convergence

AGGRESSIVE Approach

$\mathbf{x}_1$

$\mathbf{x}_2$

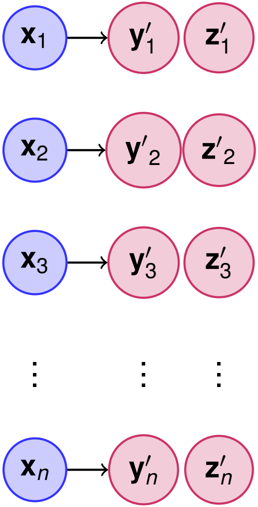
$\mathbf{x}_3$

$\vdots$

$\mathbf{x}_n$

repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new **w** using feedback
until Convergence

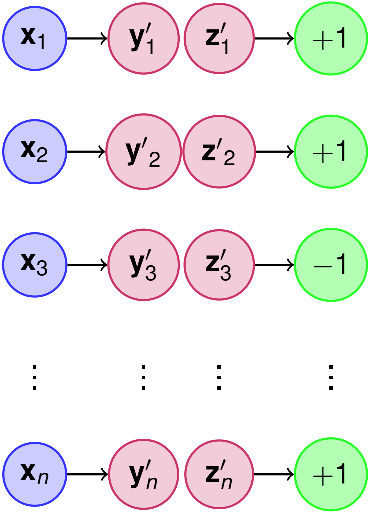
AGGRESSIVE Approach



repeat
    for all input sentences **do**
        Solve the inference problem
        Query *Feedback* function
    end for
    Learn a new **w** using feedback
until Convergence

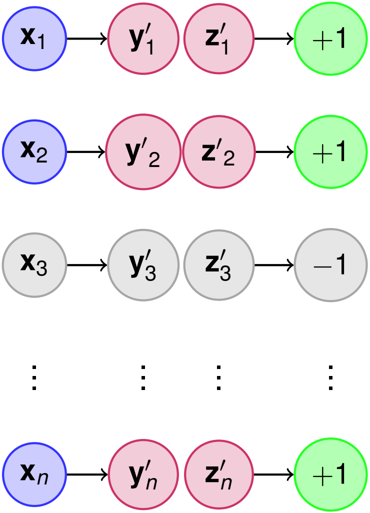
$$\mathbf{y}, \mathbf{z} = \arg\max \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

AGGRESSIVE Approach



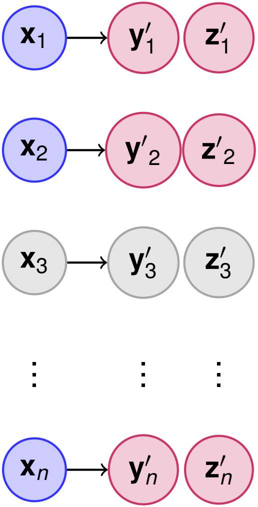
repeat
    for all input sentences **do**
        Solve the inference problem
        Query *Feedback* function
    end for
    Learn a new **w** using feedback
until Convergence

AGGRESSIVE Approach



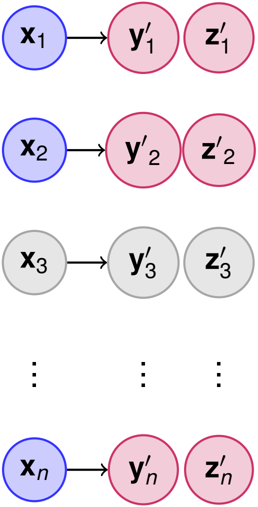
repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new **w** using feedback
until Convergence

AGGRESSIVE Approach



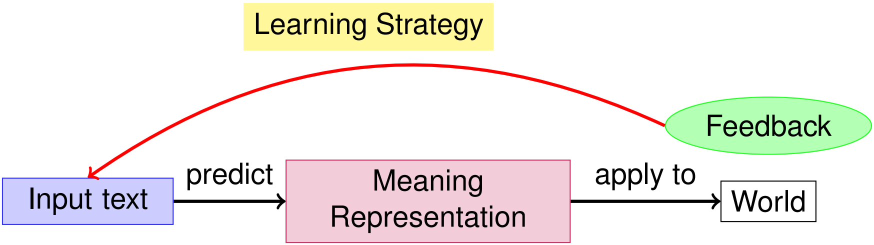
repeat
  for all input sentences **do**
    Solve the inference problem
    Query *Feedback* function
  end for
  Learn a new **w** using feedback
until Convergence

AGGRESSIVE Approach



repeat
    for all input sentences **do**
        Solve the inference problem
        Query *Feedback* function
    end for
    Learn a new $\mathbf{w}$ using feedback
until Convergence

Summary of Learning Strategies



- DIRECT Uses both positive and negative feedback as examples to train a binary classifier.
- AGGRESSIVE Adapts the feedback signal and uses only positive feedback to train a structured predictor.

Outline

Model

INPUT $\mathbf{x}$

HIDDEN $\mathbf{y}$

OUTPUT $\mathbf{z}$

What is the largest state that borders Texas?

`largest(state(next_to(texas)))`

$$\hat{\mathbf{z}} = F_{\mathbf{w}}(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathcal{Y},\mathbf{z}\in\mathcal{Z}} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$$

- **First-order**: Map lexical items. largest $\rightarrow$ `largest`
- **Second-order**: Composition. `next_to(state(·))` or `state(next_to(·))`

Inference procedure leverages the typing information of the domain.

First-order Decisions

How many people live in the state of Texas ?

Goal: `population(state(texas))`

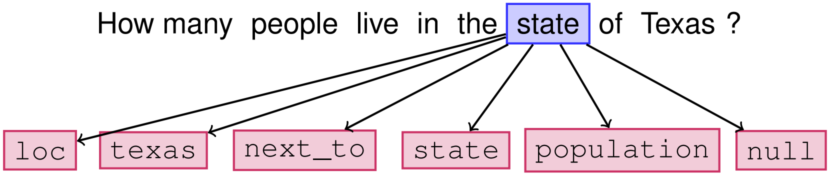
First-order Decisions

How many people live in the state of Texas ?

`loc` `texas` `next_to` `state` `population` `null`

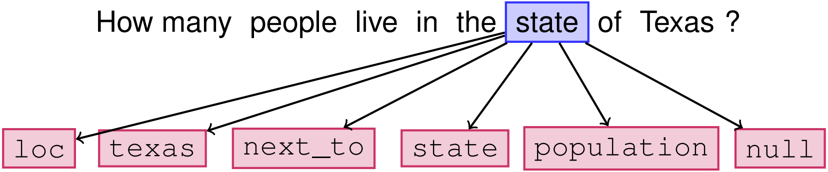
Goal: `population(state(texas))`

First-order Decisions



Goal: `population(state(texas))`

First-order Decisions

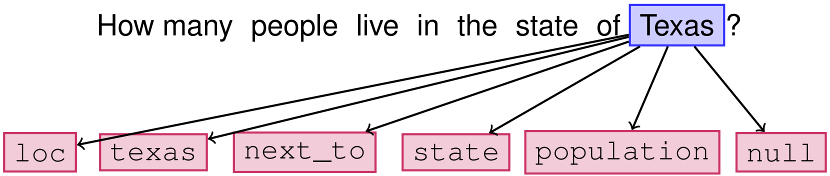


- Use a simple lexicon to bootstrap the process.

> texas
texas
> state
state
> population
population
> loc
in
> next_to
next
borders
adjacent

Goal: `population(state(texas))`

First-order Decisions

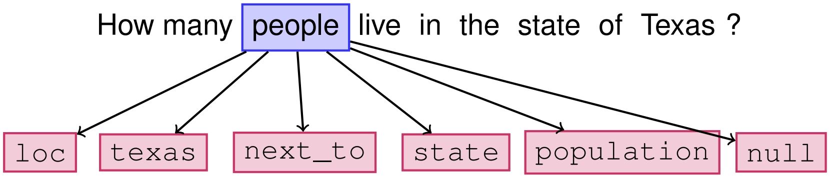


- Use a simple lexicon to bootstrap the process.

> texas
texas
> state
state
> population
population
> loc
in
> next_to
next
borders
adjacent

Goal: `population(state(texas))`

First-order Decisions



- Use a simple lexicon to bootstrap the process.

> `texas`
texas
> `state`
state
> `population`
population
> `loc`
in
> `next_to`
next
borders
adjacent

Goal: `population(state(texas))`

First-order Decisions

How many **people** live in the state of Texas ?

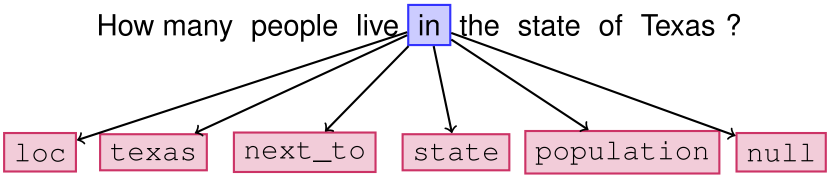
`loc` `texas` `next_to` `state` `population` `null`

- Use a simple lexicon to bootstrap the process.
- Lexical resources help us move beyond the lexicon.
  `wordnet_sim(people,population)`

```
> texas
texas
> state
state
> population
population
> loc
in
> next_to
next
borders
adjacent
```

Goal: `population(state(texas))`

First-order Decisions

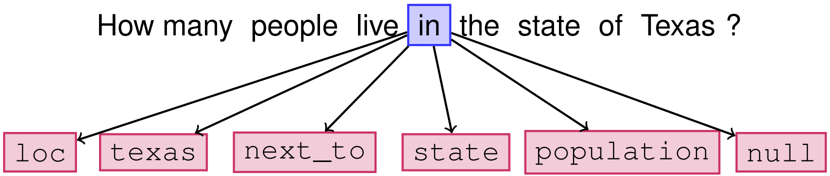


- Use a simple lexicon to bootstrap the process.
- Lexical resources help us move beyond the lexicon.
  `wordnet_sim(people,population)`

> `texas`
texas
> `state`
state
> `population`
population
> `loc`
in
> `next_to`
next
borders
adjacent

Goal: `population(state(texas))`

First-order Decisions



- Use a simple lexicon to bootstrap the process.
- Lexical resources help us move beyond the lexicon.

  `wordnet_sim(people,population)`
- Context helps disambiguate between choices.

> `texas`
texas
> `state`
state
> `population`
population
> `loc`
in
> `next_to`
next
borders
adjacent

Goal: `population(state(texas))`

Second-order Decisions

How do we compose the predicates and constants.

Domain dependent:

- Encode typing information inherent in the domain into the inference procedure.
- `population(state(·))` vs `state(population(·))`

Features:

- Dependency path distance.
- Word position distance.
- Predicate "bigrams".
- `next_to(state(·))` vs `state(next_to(·))`

Outline

Evaluation

Domain:

GEOQUERY U.S Geographical Questions.

- Response 250. $(\mathbf{x}, r)$ pairs. Zero meaning representations.
- Query 250. $(\mathbf{x})$ sentences.

Evaluation metric:

Accuracy (percentage of meaning representations that return the correct answer).

Learning Behavior

Algorithm	R250	Q250
NoLearn	22.2	—
Direct		
Aggressive		
Supervised	87.6	80.4

Learning Behavior

Algorithm	R250	Q250
NoLearn	22.2	—
Direct		
Aggressive		
Supervised	87.6	80.4

- NoLearn used to initialize both learning approaches.

Learning Behavior

Algorithm	R250	Q250
NoLearn	22.2	—
Direct		
Aggressive		
Supervised	87.6	80.4

- **Q:** How good is our model when trained in a fully supervised manner?

Learning Behavior

Algorithm	R250	Q250
NoLearn	22.2	—
Direct		
Aggressive		
Supervised	87.6	80.4

- **Q:** How good is our model when trained in a fully supervised manner?
- **A:** 80% on test data. Other supervised methods range from 60% to 85% accuracy.

Learning Behavior

Algorithm	R250	Q250
NoLearn	22.2	—
Direct		
Aggressive		
Supervised	87.6	80.4

- **Q:** Is it possible to learn without any meaning representations?

Learning Behavior

Algorithm	R250	Q250
NoLearn	22.2	—
Direct	75.2	69.2
Aggressive	82.4	73.2
Supervised	87.6	80.4

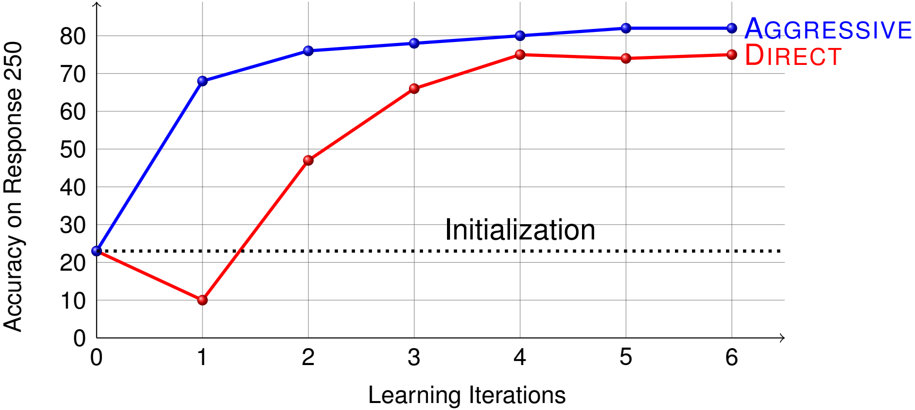
- **Q:** Is it possible to learn without any meaning representations?
- **A:** Yes!
- **A:** Learns to cover more of the Response data set.
- **A:** And only 7% below the Supervised upper bound.

Learning Behavior

Algorithm	R250	Q250
NoLearn	22.2	—
Direct	75.2	69.2
Aggressive	82.4	73.2
Supervised	87.6	80.4

- **Q:** Is it possible to learn without any meaning representations?
- **A:** Yes!
- **A:** Learns to cover more of the Response data set.
- **A:** And only 7% below the Supervised upper bound.

Learning Behavior



- AGGRESSIVE correctly interprets 16% that DIRECT does not. 9% vice-versa. Leaving only 9% incorrect.

Learning from Indirect Supervision

Similar to indirect learning protocols:

- Learning a binary classifier with “hidden explanation”. Supervision only required for binary data. No labeled structures. NAACL 2010 (Chang, Goldwasser, Roth, Srikumar 2010a).
- Structured learning with binary and structured labels. Mix of supervision for binary data and structured data. Binary label indicates whether input has a “good” structure. ICML 2010 (Chang, Goldwasser, Roth, Srikumar 2010b).

Conclusions

Contributions:

- Response Driven Learning. A new learning paradigm that doesn't rely on annotated meaning representations. Supervised at the response level. Natural supervision signal.
- Two learning algorithms capable of working within response driven learning.
- A shallow semantic parsing model.

Future work:

- Can we combine the two learning algorithms?
- Other semantic parsing domains?
- Response driven learning for other tasks?