Unanimous Prediction For 100% Precision With Application To Learning Semantic Mappings

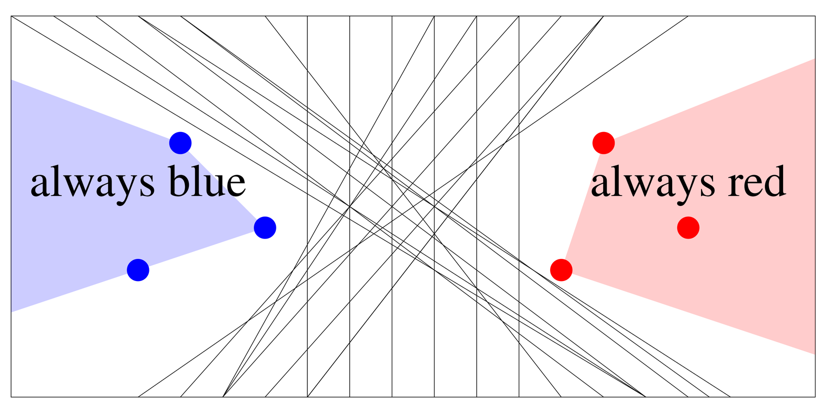
Fereshte Khani, Martin Rinard, Percy Liang

Stanford University

Introduction

If a user asks a system "*How many painkillers should I take?*", it is much better for the system to say "don't know" rather than making a costly incorrect prediction.

Analogy

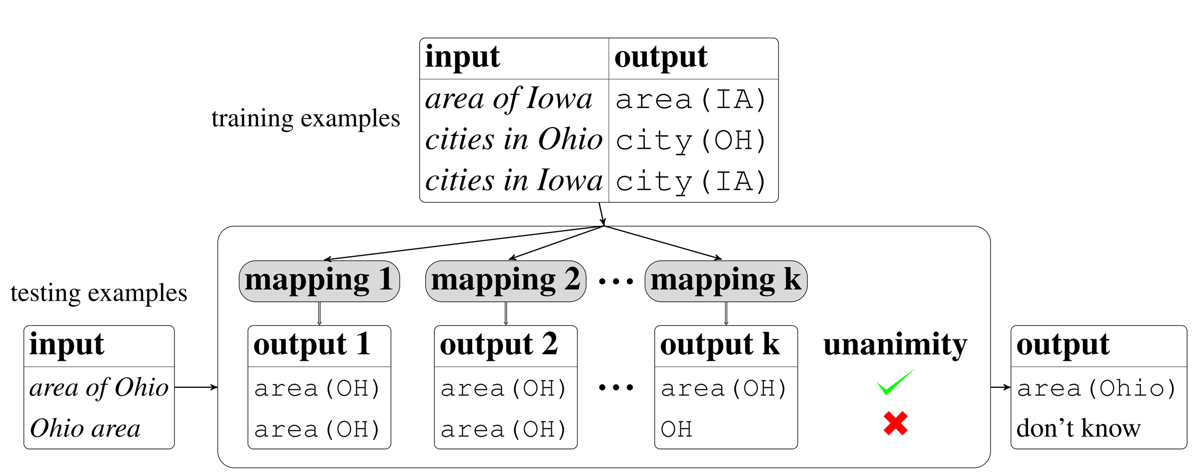


Goal

We present a system which **learns** a semantic mapping which **guarantees 100% precision** under its model assumptions.

area of Ohio → {area, OH}

Unanimity principle



Framework

Training set:

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

Source atoms

utterences		source atoms
area of Iowa	→	{*area, of, Iowa*}
cities in Ohio		{*cities, in, Ohio*}
cities in Iowa		{*cities, in, Iowa*}

Target atoms

logical forms		target atoms
area(IA)	→	{area, IA}
city(OH)		{city, OH}
city(IA)		{city, IA}

Framework

Hypothesis space ($\mathcal{M}$):

mapping 1	mapping 2	…	mapping k
cities → {city}	*cities* → {}		*cities* → {city, area, IA, OH}
in → {}	*in* → {}		*in* → {}
of → {}	*of* → {}		*of* → {}
area → {area}	*area* → {}		*area* → {}
Iowa → {IA}	*Iowa* → {}		*Iowa* → {}
Ohio → {OH}	*Ohio* → {}		*Ohio* → {area, area, city, city}

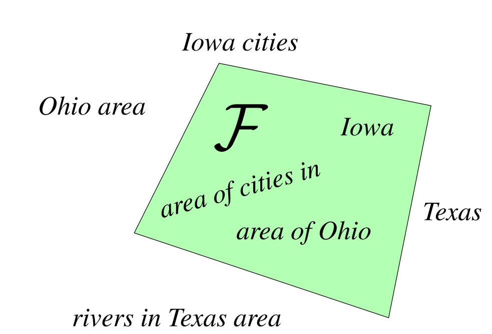
Consistent mappings ($\mathcal{C}$):

$$\mathcal{C} \stackrel{\text{def}}{=} \{M \in \mathcal{M} \mid \forall i, M(x_i) = y_i\}$$

mapping 1	mapping 2	mapping 3	mapping 4
cities → {city}	*cities* → {}	*cities* → {city}	*cities* → {}
in → {}	*in* → {city}	*in* → {}	*in* → {city}
of → {}	*of* → {}	*of* → {area}	*of* → {area}
area → {area}	*area* → {area}	*area* → {}	*area* → {}
Iowa → {IA}	*Iowa* → {IA}	*Iowa* → {IA}	*Iowa* → {IA}
Ohio → {OH}	*Ohio* → {OH}	*Ohio* → {OH}	*Ohio* → {OH}

Safe set ($\mathcal{F}$):

$$\mathcal{F} \stackrel{\text{def}}{=} \{x : |\{M(x) : M \in \mathcal{C}\}| = 1\}$$



Linear algebraic formulation

$$S = \begin{array}{c} \\ \textit{area of Iowa} \\ \textit{cities in Ohio} \\ \textit{cities in Iowa} \end{array} \begin{array}{c} \textit{area of Ohio cities in Iowa} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

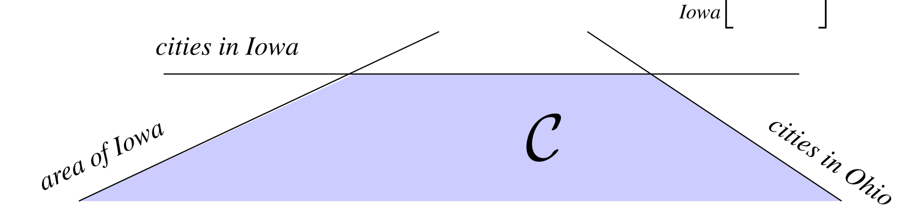
$$T = \begin{array}{c} \\ \texttt{area(IA)} \\ \texttt{city(OH)} \\ \texttt{city(IA)} \end{array} \begin{array}{c} \texttt{area city OH IA} \\ \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

$$\mathrm{M} = \begin{array}{c} \textit{area} \\ \textit{of} \\ \textit{Ohio} \\ \textit{cities} \\ \textit{in} \\ \textit{Iowa} \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$xM = y \qquad [1\ 1\ 0\ 0\ 0\ 1] \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [1\ 0\ 0\ 1]$$

Linear algebraic formulation

$$SM = T \qquad \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} ? \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$



Integer linear programming

$$\mathcal{C} = \{M \in \mathbb{Z}_{\geq 0}^{n_s \times n_t} : SM = T\}$$

Proposition. Let v be a random vector.

$$\begin{aligned} &\min.\ xMv && \max.\ xMv \\ &\text{s.t. } SM = T && \text{s.t. } SM = T \\ &M \succeq 0 && M \succeq 0 \end{aligned}$$

With probability 1, $x \in \mathcal{F}$ iff both ILPs have same answer.

Computation. Linear at training time, solving two ILPs at test time

Linear programming

$$\mathcal{C}_{\text{LP}} \stackrel{\text{def}}{=} \{M \in \mathbb{R}_{\geq 0}^{n_s \times n_t} \mid SM = T\}$$

Proposition. Let M_1 and M_2 be two "random enough" mappings inside $\mathcal{C}_{\text{LP}}$. With probability 1, $x \in \mathcal{F}_{\text{LP}}$ iff $xM_1 = xM_2$.

Computation. Solving one LP at training time, linear at test time

Linear system

$$\mathcal{C}_{\text{LS}} \stackrel{\text{def}}{=} \{M \in \mathbb{R}^{n_s \times n_t} \mid SM = T\}$$

Proposition. The vector x is in row space of S iff $x \in \mathcal{F}_{\text{LS}}$.

A linear combination of training examples:

M(*area of Ohio*) = M(*area of Iowa*) + M(*cities in Ohio*) - M(*cities in Iowa*)

$$S = \begin{array}{c} \textit{area of Iowa} \\ \textit{cities in Ohio} \\ \textit{cities in Iowa} \end{array} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{array}{c} 1 \\ 1 \\ -1 \end{array} \qquad T = \begin{array}{c} \texttt{area(IA)} \\ \texttt{city(OH)} \\ \texttt{city(IA)} \end{array} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{array}{c} 1 \\ 1 \\ -1 \end{array}$$

$$x = {}_{\textit{area of Ohio}}[1\ 1\ 1\ 0\ 0\ 0] \qquad y = {}_{\texttt{area(OH)}}[1\ 0\ 1\ 0]$$

Details

- **Source atoms:** Replace words with n-grams to handle polysemy.
- **Target atoms:** Add ordering information to predicates to reconstruct logical forms.
- **Removing noise:** Use a relaxed constraint, $\|SM - T\|_1 \leq \gamma$, instead of $SM = T$.

Other applications

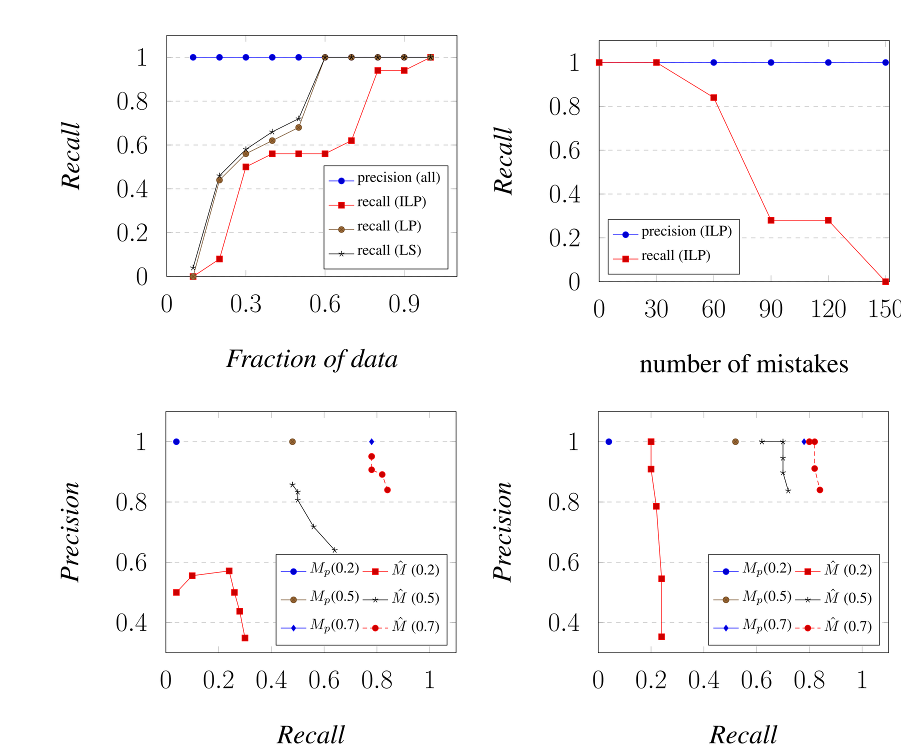
- **Active learning:** Choose linearly independent sentences to be annotated.
- **Paraphrasing:** Find all sentences that yield the same thing under all consistent semantic mappings.
- **Learning from denotations:** Training data consists of (question, answer) pairs.

Results

Artificial dataset

Input/output vocabulary size is 70.

w34, w22, w17, w12 → p10, p15, p10, p20, p40, p47



GeoQuery dataset

880 (question, logical form) pairs

how long is the mississippi → (answer (len (riverid mississippi)))