Jointly Learning to Parse and Perceive: Connecting Natural Language to the Physical World

Josue Alfaro, Ryan Wolter
Goal / Motivation

● Task: Grounded language acquisition
● Goal: Logical Semantics with Perception (LSP)
  ○ Jointly learns to parse language and perceive the world in a weakly supervised fashion
● Motivation
  ○ Understanding the mapping of NL to physical environments enable NL interactions with robots or other agents
  ○ Acquiring both parsing and perception knowledge is necessary to understand novel language in novel environments
    ■ Both modes together improve performance
  ○ In other research, perception is often ignored, they assume direct access to logical representation of environment
    ■ When Perception is included there have been several drawbacks
      ● Fully supervised models require large amounts of annotation
      ● You have limited semantic representation
    ■ However, weakly supervised techniques allow cheaper annotation
Walkthrough

These are sets!

Perception

Environment \( d \)

Know. Base \( \Gamma \)

- \( \text{mug}(1) \)
- \( \text{mug}(3) \)
- \( \text{blue}(1) \)
- \( \text{table}(4) \)
- \( \text{on-rel}(1, 4) \)
- \( \text{on-rel}(3, 4) \)

... 

Parse

Language \( \ell \)

"blue mug on table"

Logical form \( \ell \)

\[
\lambda x. \exists y. \text{blue}(x) \land \\
\text{mug}(x) \land \\
\text{on-rel}(x, y) \land \\
\text{table}(y)
\]

Grounding: \( g = \{(1, 4)\} \), Denotation: \( \gamma = \{1\} \)

Evaluate

\( \{1\} \)

\{1\} \hspace{1cm} \{(1, 4), (3, 4)\} 

\{1\} \hspace{1cm} \{1, 3\} \hspace{1cm} \{(1, 4), (3, 4)\} \hspace{1cm} \{4\}

blue(x) \hspace{1cm} mug(x) \hspace{1cm} on-rel(x, y) \hspace{1cm} table(y)
Data - Scene Understanding

- A number of segmented images
- Indoor environments or objects like mugs, tables
- Descriptions collected via Mechanical Turk
- Authors then manually annotated the descriptions with their denotations and logical forms
- Same objects are in each scene, just in different locations
Data - GeoQA

- Several maps containing cities, states, parks, lakes etc.
- Components are given by polygons of lat/long coordinates marking boundaries
- Entities have known names
- Questions were handcrafted by other researchers, then manually annotated
Data

Each example is: \( \{(z^i, \gamma^i, d^i, l^i, \Gamma^i)\}_{i=1}^{n} \)

- \( z^i \) = natural language statements
- \( \gamma^i \) = annotated denotations
- \( d^i \) = environments
- \( l^i \) = gold standard logical form
- \( \Gamma^i \) = gold standard logical knowledge base
## Data

<table>
<thead>
<tr>
<th>Data Set Statistics</th>
<th>SCENE</th>
<th>GEOQA</th>
</tr>
</thead>
<tbody>
<tr>
<td># of environments</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>Mean entities / environment $d$</td>
<td>4.1</td>
<td>6.9</td>
</tr>
<tr>
<td>Mean # of entities in denotation $\gamma$</td>
<td>1.5</td>
<td>1.2</td>
</tr>
<tr>
<td># of statements</td>
<td>284</td>
<td>263</td>
</tr>
<tr>
<td>Mean words / statement</td>
<td>6.3</td>
<td>6.3</td>
</tr>
<tr>
<td>Mean predicates / log. form</td>
<td>2.6</td>
<td>2.8</td>
</tr>
<tr>
<td># of preds. in annotated worlds</td>
<td>46</td>
<td>38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lexicon Statistics</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># of words in lexicon</td>
<td>169</td>
<td>288</td>
</tr>
<tr>
<td># of lexicon entries</td>
<td>712</td>
<td>876</td>
</tr>
<tr>
<td>Mean parses / statement</td>
<td>15.0</td>
<td>8.9</td>
</tr>
</tbody>
</table>

- Only 25 total environments, or pictures from which questions or statements can be made
- Around 530 total natural language statements about the scenes
- Small vocab of between 170 and 290 words.
Logical Semantics with Perception (LSP) Model

- LSP is a linear model that is composed of three parts
  - Perception: $f_{\text{per}}$ -> constructs logical knowledge base from low-level features based on representations of the environments
  - Parsing: $f_{\text{prs}}$ -> semantically parses natural language into lambda calculus queries that can be evaluated against the knowledge base
  - Evaluation: $f_{\text{eval}}$ -> deterministically executes query against knowledge base to produce grounding & denotation
Perception Function

- Scores logical knowledge base using a set of **per-predicate** binary classifiers
  - Classifiers independently assign a score to whether each entity (entity pair), in the environment, is an element of each category (relation)

\[
 f_{per}(\Gamma, d; \theta_{per}) = \sum_{c \in C} h(\gamma^c, d; \theta^c_{per}) + \sum_{r \in R} h(\gamma^r, d; \theta^r_{per})
\]

Score of logical knowledge base factors into per-relation, per-category scores \( h \)

\[
 h(\gamma^c, d; \theta^c_{per}) = \sum_{e \in E_d} \gamma^c(e)(\theta^c_{per})^T \phi_{cat}(e)
\]

\[
 h(\gamma^r, d; \theta^r_{per}) = \sum_{(e_1, e_2) \in E_d} \gamma^r(e_1, e_2)(\theta^r_{per})^T \phi_{rel}(e_1, e_2)
\]

<table>
<thead>
<tr>
<th>Scene Data Features</th>
<th>GeoQA Data Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area and perimeter of entity</td>
<td>Area and perimeter of entity</td>
</tr>
<tr>
<td>Distance between entity centroids</td>
<td>Distance between entity centroids</td>
</tr>
<tr>
<td>Histogram of Oriented Gradients</td>
<td>Phrase co-occurrences with entity names in Clueweb09</td>
</tr>
<tr>
<td>RGB Color Histogram</td>
<td></td>
</tr>
</tbody>
</table>
Perception Function Example

Take every object and decide whether it’s a “mug” independently of every other decision.

Train independent classifier like $\theta^T_{\text{mug}}$ to decide whether or not entity is a mug from features.
Parsing Function

- Defined using Combinatory Categorial Grammar
- Grammar is given by a lexicon that maps to syntactic categories and logical forms
- A given sentence may have multiple parses, so the semantic parser scores each one learning the correct form

\[
f_{prs}(\ell, t, z; \theta_{prs}) = \theta_{prs}^T \phi_{prs}(\ell, t, z)
\]

\[
\phi_{prs}(\ell, t, z) = 1(\text{lexicon entry})
\]

\[
\phi_{prs}(\ell, t, z) = \phi_{prs}(\text{left}(\ell, t, z)) + \phi_{prs}(\text{right}(\ell, t, z)) + 1(\text{combinator})
\]

If parse tree is terminal then the function gets a vector with a single value indexed by the lexicon entry

For non-terminal parse trees, non-terminal features are defined over combinator rules in the parse tree
Parsing Function Example

<table>
<thead>
<tr>
<th>the</th>
<th>mugs</th>
<th>are</th>
<th>right</th>
<th>of</th>
<th>the</th>
<th>monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N/N$</td>
<td>$N$</td>
<td>$(S\backslash N)/N$</td>
<td>$N/PP$</td>
<td>$PP/N$</td>
<td>$N/N$</td>
<td>$N$</td>
</tr>
<tr>
<td>$\lambda f.f$</td>
<td>$\lambda x.mug(x)$</td>
<td>$\lambda f.\lambda g.\lambda x.g(x) \land f(x)$</td>
<td>$\lambda f.\lambda x.\exists y.\text{right-rel}(x, y) \land f(y)$</td>
<td>$\lambda f.f$</td>
<td>$\lambda x.mug(x)$</td>
<td>$\lambda x.monitor(x)$</td>
</tr>
<tr>
<td>$N$</td>
<td>$\lambda x.mug(x)$</td>
<td>$S\backslash N$</td>
<td>$S\backslash N: \lambda g.\lambda x.\exists y.g(x) \land \text{right-rel}(x, y) \land \text{monitor}(y)$</td>
<td>$S: \lambda x.\exists y.mug(x) \land \text{right-rel}(x, y) \land \text{monitor}(y)$</td>
<td>$\lambda x.mug(x)$</td>
<td>$\lambda x.monitor(x)$</td>
</tr>
</tbody>
</table>

First row is lexical categories

Remaining rows are applications of CCG combinators to end up with our logical form
Evaluation Function

- Evaluates the query \( l \) on the database gamma to produce a denotation
  - A score of 0 is assigned to the computed denotation and negative infinity to all others
- Deterministic

\[
\begin{align*}
\text{If } l = \lambda x. c(x) \text{ then } \gamma &= \gamma^c. \\
\text{If } l = \lambda x. \lambda y. r(x, y), \text{ then } \gamma &= \gamma^r.
\end{align*}
\]

Base Case

\[
\begin{align*}
\text{If } l &= \lambda x. l_1(x) \wedge l_2(x), \text{ then } \\
\gamma(e) &= 1 \text{ iff } \gamma_1(e) = 1 \wedge \gamma_2(e) = 1. \\
\text{If } l &= \lambda x. \exists y. l_1(x, y), \text{ then } \\
\gamma(e_1) &= 1 \text{ iff } \exists e_2. \gamma_1(e_1, e_2) = 1.
\end{align*}
\]

Complex logical forms are computed recursively by decomposing \( l \) according to its logical structure.
Evaluation Function Example

Environment $d$

Know. Base $\Gamma$
- mug(1)
- mug(3)
- blue(1)
- table(4)
- on-rel(1,4)
- on-rel(3,4)
...

Language $z$

“blue mug on table”

Logical form $\ell$

$\lambda x. \exists y. \text{blue}(x) \land \text{mug}(x) \land \text{on-rel}(x,y) \land \text{table}(y)$

Grounding: $g = \{(1,4)\}$, Denotation: $\gamma = \{1\}$
Training

- **Weakly supervised training**
  - Training the Perception and Parser components without the gold logical form and gold knowledge base.
  - Only have the denotation (answer)

- **How?**
  - We need to find the output of the Perception and Parser components that would best explain the answer, and give higher weighting to those features.

- **Stochastic subgradient method for training**
  - Essentially means they update the parameters toward the gold standard
Training

- But wait, we don’t have that gold standard!
  - The authors find the **best** explanation for the correction answer by using a beam search over possible explanations, and use this in place of the gold.
  - They do this using an integer linear program (which basically is just a solver)
  - This is done for both the parser and the knowledge base

New weights = Current weights + Regularization * ( Best Explanation - Best Prediction)
Evaluation

- Two methods of evaluation
  - Predict the correct denotation
    - Pick the answer
    - Less useful because can often be guessed easily
      (The mugs on the table - can just pick the mugs!)
  - Predict the correct grounding
    - Ground all words in a statement to the correct entities and relations
    - Given a logical form, list the tuples that are true for the form
    - ex:

<table>
<thead>
<tr>
<th>Language and predicted logical form $\ell$</th>
<th>Predicted grounding</th>
<th>True grounding</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>monitor to the left of the mugs</code></td>
<td>${(2, 1), (2, 3)}$</td>
<td>${(2, 1), (2, 3)}$</td>
</tr>
<tr>
<td>$\lambda x. \exists y. monitor(x) \land left\text{-}rel(x, y) \land mug(y)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evaluation

● Exact Match Accuracy used for measuring performance
● Random chance achieves low success
  ○ 1-6% for SCENE
  ○ 0-1% for GeoQA
  ○ Remember, the answer is a set, so there can be multiple choices.
Results - SCENE

- LSP-CAT = categories only, no relations modeled
- LSP-F = fully supervised
- LSP-W = weakly supervised

- All models perform well with 0 relations
- Scenes with other undefined relations perform poorly
  - Ex, the mug nearest to the monitor
- LSP-CAT doesn’t model relations, which is why it does so much worse
- LSP-F performs best overall at .70 for denotation, and .65 for grounding

<table>
<thead>
<tr>
<th>Denotation $\gamma$</th>
<th>0 rel.</th>
<th>1 rel.</th>
<th>other</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSP-CAT</td>
<td>0.94</td>
<td>0.45</td>
<td>0.20</td>
<td>0.51</td>
</tr>
<tr>
<td>LSP-F</td>
<td>0.89</td>
<td>0.81</td>
<td>0.20</td>
<td>0.70</td>
</tr>
<tr>
<td>LSP-W</td>
<td>0.89</td>
<td>0.77</td>
<td>0.16</td>
<td>0.67</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grounding $g$</th>
<th>0 rel.</th>
<th>1 rel.</th>
<th>other</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSP-CAT</td>
<td>0.94</td>
<td>0.37</td>
<td>0.00</td>
<td>0.42</td>
</tr>
<tr>
<td>LSP-F</td>
<td>0.89</td>
<td>0.80</td>
<td>0.00</td>
<td>0.65</td>
</tr>
<tr>
<td>LSP-W</td>
<td>0.89</td>
<td>0.70</td>
<td>0.00</td>
<td>0.59</td>
</tr>
</tbody>
</table>

| % of data           | 23     | 56     | 21    | 100   |
Results - GeoQA

- Scenes with other undefined relations perform poorly.
- LSP-W performs best overall at .51 for denotation, and .50 for grounding, but LSP-F is close.
- It is noted that for BOTH environments, weakly supervised and fully supervised training perform similarly, with accuracy differences ranging between 3%-6%.
Critique

● Severe lack of data (~25 environments) (-)
  ○ Scene data even reused the same objects, and number of object classes isn’t given
  ○ If the semi-supervised worked well, and it’s point was to be able to create a lot more data to train on cheaply, why didn’t they utilize this to demonstrate potential increases in accuracy with more data?

● Evaluation technique (holdout) (-)
  ○ Authors trained the model against all data with only one holdout environment
    ■ They then switch the holdout environment to another and retrained and performed evaluation again, summing the total performance results
    ■ This means they didn’t have any TRUE holdout data
      ● Should have left one environment out at least until final evaluation
      ● More holdout examples would be better
Critique

● Component Error Analysis (+)
  ○ Authors evaluated the performance of the perception classifier, and the semantic parser, showing how the model was erroring.
    ■ In general, perception performed well, semantic parser performed well but was tripped up by non-essential adjectives (such as LCD monitor)
    ■ They were able to show that modeling the relationships significantly improved performance

● Training Data is not available (-)
  ○ Website is down

● Model code is not published (-)
  ○ Makes it really hard to understand what the paper is specifically saying
Critique

● Lack of performance data (-)
  ○ How long does it take to train?
  ○ How long to produce a prediction?
  ○ Size of model (aka number of parameters)?
  ○ What was the hardware configuration?

● Requires pre-segmented input (-)
  ○ The bounding boxes must be given in advance

● Larger domain = hard time with inference during training (-)
  ○ Beam-search over a larger domain gets costly quickly
Future Work

- Consider expanding entities in environment to diverse collection of real world images
  - More than just mugs, screens -- give me a larger domain!
- Include many more entities in an environment with complex spatial relations
- Use more data as this is weakly supervised
- Incorporate neural networks, deep learning to this VQA problem
  - CNNs for the perception piece, then LSTMs for the parser, but still creating as an intermediate result the logical forms and knowledge base.
Discussion

● Pros?
● Cons?
● What would you have done differently?