Visualization Question Answering Using Introspective Program Synthesis

Yanju Chen, Xifeng Yan, Yu Feng
University of California, Santa Barbara
Deep Learning Models

- Motivations -
  
  • Data-Driven/-Hungry
  • Deep & Large-Scale

... if they work

The predictions are wrong. What's going on?
VQA: A Motivating Example (Visualization Question Answering)

- Motivations

Given a stacked bar chart that represents opinions for future economic growth for different countries, a user describes her query based on the visualization in natural language:

Which *country*'s economy will get *most worse* over next 12 months?

A Motivating Query

- A Visualization Question Answering (VQA) task is to design an algorithm that automatically finds the answer to a natural language query based on a given visualization.
Existing Approaches & Challenges

- **Motivations**

  **Fully Supervised Machine Learning:** SmBoP\(^1\), NL2code\(^2\)
  - Requires manual annotated logic forms / programs as supervised training data

  **Weakly Supervised Machine Learning:** TaPas\(^3\)
  - Requires only question-answer pairs for training

---

\(^3\) TaPas: Weakly Supervised Table Parsing via Pre-training. Herzig, J. et al. ACL 2020.
- Motivations -

**Observations**

- For mainstream weakly supervised approaches that directly output VQA answers, they are:
  - non-trivial for human beings to understand, and
  - hard to fix if there’s error in model reasoning/answer.

- In this work, we investigate such a slightly different problem setting where:
  - not all the predictions are correct (usually only one of them is correct, or even sometimes none), and
  - predictions may conflict with each other.

Can we use program synthesis techniques to get “background explanation” programs based on the model’s predictions?  
input: visualization  
output: model prediction

![A Simple Visualization and Its Table](image)

A Simple Query

Which country has **highest** Improve value?

“Brazil”, “Japan”, “China”, “U.S.” ...

Model Predictions

\[
\text{project}(\text{aggregate}(I, \text{null}, \max, \text{null}), ["Country"])
\]

⇒ feasible for "Brazil", "Japan"

\[
\text{project}(\text{aggregate}(I, \text{null}, \text{null}, \text{null}), ["Country"])
\]

⇒ feasible for all: "Brazil", "Japan", "China", "U.S."
A Straw-Man Proposal

Can we use program synthesis techniques to get "background explanation" programs based on the model's predictions?

input: visualization  ➔  output: model prediction

• The Straw-Man Proposal
  1. For every single prediction of the deep learning model, we trigger an off-the-shelf synthesizer to solve for program(s)
  2. Then we end up with a bunch of programs and (maybe) ask the user to pick the "best" one

• There are potential issues:
  • **Not scalable**: For cases where large number of predictions are produced, this won't scale well
  • **Model dependent**: If predictions do not contain the correct answer, synthesis done will be meaningless
  • **Unclear of best-fitting definition**: There's no formal definition of best-fitting program; need to connect 3 parties: elements in visualization, language units in query and production rules in explanation programs
Overview: POE

- Fixing Deep Learning Model's (Noisy) Outputs via **Introspective Program Synthesis**
- Search Space Induction via **Abstract Program Synthesis**
- Finding Best Consistent Programs via **Optimal Program Synthesis**

In the context of this work, we use *programs* and *explanations* interchangeably.
A Walkthrough of POE

Data

<table>
<thead>
<tr>
<th>Country</th>
<th>opinion</th>
<th>%</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>Improve</td>
<td>84</td>
<td>blue</td>
</tr>
<tr>
<td>Brazil</td>
<td>Remain the same</td>
<td>12</td>
<td>orange</td>
</tr>
<tr>
<td>Brazil</td>
<td>Worsen</td>
<td>5</td>
<td>red</td>
</tr>
<tr>
<td>China</td>
<td>Improve</td>
<td>63</td>
<td>blue</td>
</tr>
<tr>
<td>China</td>
<td>Remain the same</td>
<td>9</td>
<td>orange</td>
</tr>
<tr>
<td>China</td>
<td>Worsen</td>
<td>2</td>
<td>red</td>
</tr>
<tr>
<td>Tunisia</td>
<td>Improve</td>
<td>75</td>
<td>blue</td>
</tr>
</tbody>
</table>

Explanation#1

T0 = (pivot(T, "opinion", ";")
T1 = (select(T0, "Improve", eqmax, null)
T2 = (project(T1, [{"Country"}])

Query
Which country’s economy will get most worse over next 12 months?

Explanation#2

T0 = (pivot(T, "opinion", ";")
T1 = (select(T0, "Worsen", eqmax, null)
T2 = (project(T1, [{"Country"}])
A Walkthrough of POE

- Synthesis Using POE -

• Original TAPAS Outputs:

(0.78, Brazil), (0.67, Japan), (0.55, Greece), ...

• POE’s Abstract Program Synthesis Outputs:

1. `project(select(pivot(T, ⊥, ⊥), ⊥, ⊥, ⊥, ⊥, ⊥))`

2. `project(select(T, ⊥, ⊥, ⊥, ⊥))`

3. ...

• POE’s Optimal Program Synthesis Outputs:

```python
project(select(pivot(
    T, "opinion", "%"), "Improve", eqmax, null), ["Country"])  
```

`project(select(pivot(
    T, "opinion", "%"), "Worsen", eqmax, null), ["Country"])`

Visualization Question Answering Using Introspective Program Synthesis

Data

<table>
<thead>
<tr>
<th>Country</th>
<th>opinion</th>
<th>%</th>
<th>color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>Improve</td>
<td>84</td>
<td>blue</td>
</tr>
<tr>
<td>Brazil</td>
<td>Remain the same</td>
<td>12</td>
<td>orange</td>
</tr>
<tr>
<td>Brazil</td>
<td>Worsen</td>
<td>5</td>
<td>red</td>
</tr>
<tr>
<td>China</td>
<td>Improve</td>
<td>83</td>
<td>blue</td>
</tr>
<tr>
<td>China</td>
<td>Remain the same</td>
<td>9</td>
<td>orange</td>
</tr>
<tr>
<td>China</td>
<td>Worsen</td>
<td>2</td>
<td>red</td>
</tr>
<tr>
<td>Tunisia</td>
<td>Improve</td>
<td>76</td>
<td>blue</td>
</tr>
<tr>
<td>Japan</td>
<td>Improve</td>
<td>16</td>
<td>blue</td>
</tr>
<tr>
<td>Japan</td>
<td>Remain the same</td>
<td>49</td>
<td>orange</td>
</tr>
<tr>
<td>Japan</td>
<td>Worsen</td>
<td>33</td>
<td>red</td>
</tr>
<tr>
<td>Czech Rep.</td>
<td>Improve</td>
<td>13</td>
<td>blue</td>
</tr>
<tr>
<td>Czech Rep.</td>
<td>Remain the same</td>
<td>27</td>
<td>orange</td>
</tr>
<tr>
<td>Czech Rep.</td>
<td>Worsen</td>
<td>60</td>
<td>red</td>
</tr>
<tr>
<td>Greece</td>
<td>Improve</td>
<td>9</td>
<td>blue</td>
</tr>
<tr>
<td>Greece</td>
<td>Remain the same</td>
<td>10</td>
<td>orange</td>
</tr>
<tr>
<td>Greece</td>
<td>Worsen</td>
<td>81</td>
<td>red</td>
</tr>
</tbody>
</table>

Syntax of A Motivating Toy DSL

```latex
\langle Table \rangle ::= \langle project \rangle, \langle ColList \rangle  
| \langle select \rangle, \langle BoolOp \rangle, \langle ColInt \rangle, \langle ConstVal \rangle  
| \langle pivot \rangle, \langle Table \rangle, \langle ColInt \rangle, \langle ColInt \rangle  
| \langle aggregate \rangle, \langle Table \rangle, \langle ColList \rangle, \langle AggrOp \rangle, \langle ColInt \rangle

\langle AggrOp \rangle ::= \mid select | \mid min | \mid max | \mid sum | \mid mean

\langle BoolOp \rangle ::= \mid < | \mid <= | \mid == | \mid >= | \mid != | \mid eqmax | \mid eqmin

\langle Table \rangle \in \langle tables \rangle, \langle ConstVal \rangle \in \langle constants \rangle  
\langle ColInt \rangle \in \langle columns \rangle, \langle ColList \rangle \in \langle columns \rangle^a
```

A Motivating Query

Which country’s economy will get most worse over next 12 months?
System Workflow in POE

A System Workflow in POE

- Synthesis Using POE -

Visualization Question Answering Using Introspective Program Synthesis
Abstract Program Synthesis with Noisy Specification

- Synthesis Using POE -

- Intuition: Narrow down program search space to such a sweet spot that:
  - respects the model outputs, and
  - promote synthesis efficiency.

Model Outputs: "Brazil", "Japan", "China", "U.S."

Abstract Synthesis Breakdown:

- feasible for all examples
- \( \text{project}(\cdot) \) feasible for all examples
- \( \text{project}(\text{aggregate}(I, \text{null}, \phi_0, \phi_1), ["Country"]) \) feasible for "Brazil", "Japan", "China"
- \( \text{project}(\text{aggregate}(I, \text{null}, \text{max}, \phi_1), ["Country"]) \) feasible for "Brazil", "Japan"
Optimal Program Synthesis for Explanation Refinement

- Intuition: Maximize consistency between explanation, visualization and query.
- Hard Constraints (Syntactic Correctness)
- Soft Constraints (Semantic Approximation)
  - NSYN: Near-Synonym Linguistic Engine
  - A linguistic engine that determines whether two linguistic units are near-synonyms (semantically similar)

NSYN(“high”, “highest”) > NSYN(“high”, “low”)

- Objective Function

\[
\sum_{w \in W} \sum_{t \in T} (1 - NSYN(w, t)) \cdot x_{w} + \sum_{p \in \mathcal{P}} PPL(P) \cdot u^{p}
\]

Maximize consistency matching.

More common abstract programs are preferred.
Evaluation

• Research Questions
  • **RQ1. Performance**: How does POE compare against state-of-the-art tools on visualization queries?
  • **RQ2. Effectiveness**: Can POE rectify wrong answers proposed by other tools?
  • **RQ3. Explainability**: Does POE synthesize explanations that well capture the question intentions and make sense to human end-users?
  • **RQ4. Ablation**: How significant is the benefit of abstract synthesis and optimal alignment?

• Benchmarks
  • **629 Visualization Question Answering Tasks from VisQA**[1]
    • Real-World Data Sources
    • Non-Trivial Questions from Real Users
    • Wide Coverage of Question Types

---

- Evaluation

**Performance**

- Comparison against TaPas\(^1\) and VisQA\(^2\)
  - VisQA: +8%
  - POE (top-1): +23%
  - POE (top-3): +27%

- Stats of Different Questions Types
  - Retrieval
  - Comparison
  - Aggregation
  - Other
  - Total

---

\(^1\) TaPas: Weakly Supervised Table Parsing via Pre-training. Herzig, J. et al. ACL 2020.

Discussions & Conclusions

• Discussions
  • Incomprehensive Questions
  • Limitation of NLP Modules

• Conclusions: A Tool for Visualization Question Answering via Introspective Program Synthesis
  • helps understand deep learning model's VQA predictions
  • fixes potentially wrong predictions by refinement

POE is open-sourced and publicly available.

https://github.com/chyanju/Poe