Programming by Examples: PL meets ML

Summit on Machine Learning meets Formal Methods
July 2018

Sumit Gulwani
Microsoft

Joint work with many collaborators
Example-based help-forum interaction

300_w30_aniSh_c1_b → w30
300_w5_aniSh_c1_b → w5

=A HUGE Thank you!!!!!! 😊😊😊😊😊😊

=MID(B1,5,2)

=MID(B1,FIND("_",$B:$B)+1,
FIND("_",REPLACE($B:$B,1,FIND("_",$B:$B),"")))-1)
Flash Fill (Excel feature)

“Automating string processing in spreadsheets using input-output examples”
[POPL 2011] Sumit Gulwani
### Number, DateTime Transformations

<table>
<thead>
<tr>
<th>Input</th>
<th>Output (round to 2 decimal places)</th>
<th>Excel/C#:</th>
<th>Python/C#:</th>
<th>Java:</th>
</tr>
</thead>
<tbody>
<tr>
<td>123.4567</td>
<td>123.46</td>
<td>#.00</td>
<td>.2f</td>
<td>#.##</td>
</tr>
<tr>
<td>123.4</td>
<td>123.40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78.234</td>
<td>78.23</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Output (3-hour weekday bucket)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEDAR AVE &amp; COTTAGE AVE; HORSHAM; 2015-12-11 @ 13:34:52;</td>
<td>Fri, 12PM - 3PM</td>
</tr>
<tr>
<td>RT202 PKWY; MONTGOMERY; 2016-01-13 @ 09:05:41-Station:STA18;</td>
<td>Wed, 9AM - 12PM</td>
</tr>
<tr>
<td>; UPPER GWYNEDD; 2015-12-11 @ 21:11:18;</td>
<td>Fri, 9PM - 12AM</td>
</tr>
</tbody>
</table>

“Synthesizing Number Transformations from Input-Output Examples”
[CAV 2012] Rishabh Singh, Sumit Gulwani
"FlashExtract: A Framework for data extraction by examples"
[PLDI 2014] Vu Le, Sumit Gulwani
# Table Reshaping

50% spreadsheets are semi-structured.
KPMG, Deloitte budget millions of dollars for normalization.

"FlashRelate: Extracting Relational Data from Semi-Structured Spreadsheets Using Examples"


<table>
<thead>
<tr>
<th>Bureau of I.A.</th>
<th>Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regional Dir.</td>
<td></td>
</tr>
<tr>
<td>Niles C.</td>
<td>Tel: (800)645-8397</td>
</tr>
<tr>
<td></td>
<td>Fax: (907)586-7252</td>
</tr>
<tr>
<td>Jean H.</td>
<td>Tel: (918)781-4600</td>
</tr>
<tr>
<td></td>
<td>Fax: (918)781-4604</td>
</tr>
<tr>
<td>Frank K.</td>
<td>Tel: (615)564-6500</td>
</tr>
<tr>
<td></td>
<td>Fax: (615)564-6701</td>
</tr>
</tbody>
</table>

FlashRelate
From few examples of rows in output table

<table>
<thead>
<tr>
<th></th>
<th>Tel</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Niles C.</td>
<td>(800)645-8397</td>
<td>(907)586-7252</td>
</tr>
<tr>
<td>Jean H.</td>
<td>(918)781-4600</td>
<td>(918)781-4604</td>
</tr>
<tr>
<td>Frank K.</td>
<td>(615)564-6500</td>
<td>(615)564-6701</td>
</tr>
</tbody>
</table>
PBE Architecture

Huge search space
- Prune using Logical reasoning
- Guide using Machine learning

Under-specification
- Guess using Ranking (PL features, ML models)
- Interact: leverage extra inputs (clustering) and programs (execution)

“Programming by Examples: PL meets ML”
[APLAS 2017] Sumit Gulwani, Prateek Jain
Flash Fill DSL

\[ \text{Tuple}(\text{String } x_1, \ldots, \text{String } x_n) \rightarrow \text{String} \]

top-level expr \( T := C \mid \text{ifThenElse}(B, C, T) \)

condition-free expr \( C := A \mid \text{Concat}(A, C) \)

atomic expression \( A := \text{SubStr}(X, P, P) \mid \text{ConstantString} \)

input string \( X := x_1 \mid x_2 \mid \ldots \)

position expression \( P := K \mid \text{Pos}(X, R_1, R_2, K) \)

\( K^{\text{th}} \) position in \( X \) whose left/right side matches with \( R_1/R_2 \).
Search Idea 1: Deduction

Let $[G \models \phi]$ denote programs in grammar $G$ that satisfy spec $\phi$

$\phi$ is a Boolean constraint over (input state $i \rightsquigarrow$ output value $o$)

**Divide-and-conquer style problem reduction**

$$[G \models \phi_1 \land \phi_2] = \text{Intersect}([G \models \phi_1], [G \models \phi_2])$$

$$= [G_1 \models \phi_2] \text{ where } G_1 = [G \models \phi_1]$$

Let $G := G_1 \mid G_2$

$$[G \models \phi] = [G_1 \models \phi] \mid [G_2 \models \phi]$$

“FlashMeta: A Framework for Inductive Program Synthesis”

[OOPSLA 2015] Alex Polozov, Sumit Gulwani
Search Idea 1: Deduction

Inverse Set: \( F^{-1}(o) \overset{\text{def}}{=} \{ (u, v) \mid F(u, v) = o \} \)

E.g. \( \text{Concat}^{-1}("\text{Abc}") = \{ ("A", "bc"), ("Ab", "c"), \ldots \} \)

Let \( G := F(G_1, G_2) \)
Let \( F^{-1}(o) \) be \( \{ (u, v), (u', v') \} \)

\[
\begin{align*}
[G \models (i \mapsto o)] &= F([G_1 \models (i \mapsto u)], [G_2 \models (i \mapsto v)]) \\
&\quad \mid F([G_1 \models (i \mapsto u')], [G_2 \models (i \mapsto v')])
\end{align*}
\]
Search Idea 2: Learning

Machine Learning for ordering search
• Which grammar production to try first?
• Which sub-goal resulting from inverse semantics to try first?

Prediction based on supervised training
• standard LSTM architecture
• Training: 100s of tasks, 1 task yields 1000s of sub-problems.
• Results: Up to 20x speedup with average speedup of 1.67

“Neural-guided Deductive Search for Real-Time Program Synthesis from Examples”
[ICLR 2018] Mohta, Kalyan, Polozov, Batra, Gulwani, Jain
Ranking Idea 1: Program Features

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vasu Singh</td>
<td>v.s.</td>
</tr>
<tr>
<td>Stuart Russell</td>
<td>s.r.</td>
</tr>
</tbody>
</table>

P1: Lower(1st char) + “.s.”
P2: Lower(1st char) + “.” + 3rd char + “.”
P3: Lower(1st char) + “.” + Lower(1st char after space) + “.”

Prefer programs (P3) with simpler Kolmogorov complexity
• Fewer constants
• Smaller constants

“Predicting a correct program in Programming by Example”
[CAV 2015] Rishabh Singh, Sumit Gulwani
Ranking Idea 2: Output Features

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>Output of P1</th>
</tr>
</thead>
<tbody>
<tr>
<td>[CPT-123]</td>
<td>[CPT-123]</td>
<td>[CPT-123]</td>
</tr>
<tr>
<td>[CPT-456]</td>
<td>[CPT-456]</td>
<td>[CPT-456]</td>
</tr>
</tbody>
</table>

P1: Input + “]”
P2: Prefix of input upto 1st number + “]”

Examine features of outputs of a program on extra inputs:

- IsYear, Numeric Deviation, # of characters, IsPerson
Disambiguation

Communicate actionable information back to user.

PL aspects

• Enable effective navigation between top-ranked programs.
• Highlight ambiguity based on distinguishing inputs.

Heuristics that can be machine learned

• Highlight ambiguity based on clustering of inputs/outputs.
• When to stop highlighting ambiguity?

“FlashProfile: A Framework for Synthesizing Data Profiles”
[OOPSLA 2018 submission] Padhi, Jain, Perelman, Polozov, Gulwani, Millstein
ML in intelligent software creation

Intelligent software (e.g., PBE component)

Advantages
• Better models
• Less time to author
• Online adaptation, personalization

Logical strategies
Creative heuristics

Features
Model

Can be learned and maintained by ML-backed runtime

“Programming by Examples: PL meets ML”
[APLAS 2017] Sumit Gulwani, Prateek Jain
New frontiers in Program Synthesis

- **Search methodology**: Code repositories [Murali et.al., ICLR 2018]
- **Language**: Neural program induction
  - [Graves et al., 2014; Reed & De Freitas, 2016; Zaremba et al., 2016]
- **Applications**:
  - Code Transformations [Rolim et.al; ICSE 2017]
  - Personalized Learning [Gulwani; CACM 2014]
- **Intent specification**:
  - Natural language [Huang et.al., NAACL-HLT 2018; Gulwani & Marron, SIGMOD 2014]
  - Predictive [Raza & Gulwani; AAAI 2017]
- **Objectives**: Efficiency, Readability
Conclusion

Program Synthesis is a new frontier in AI.
• 10-100x productivity increase in some domains.
  – Data Wrangling: Data scientists spend 80% time.
  – Code Refactoring: Developers spend 40% time in migration.
• 99% of end users are non-programmers.

Next-generational AI techniques under the hood
• Logical Reasoning + Machine Learning

The Future: Multi-modal programming with Examples and NL