Programming by Examples: PL meets ML

Summit on Machine Learning meets Formal Methods

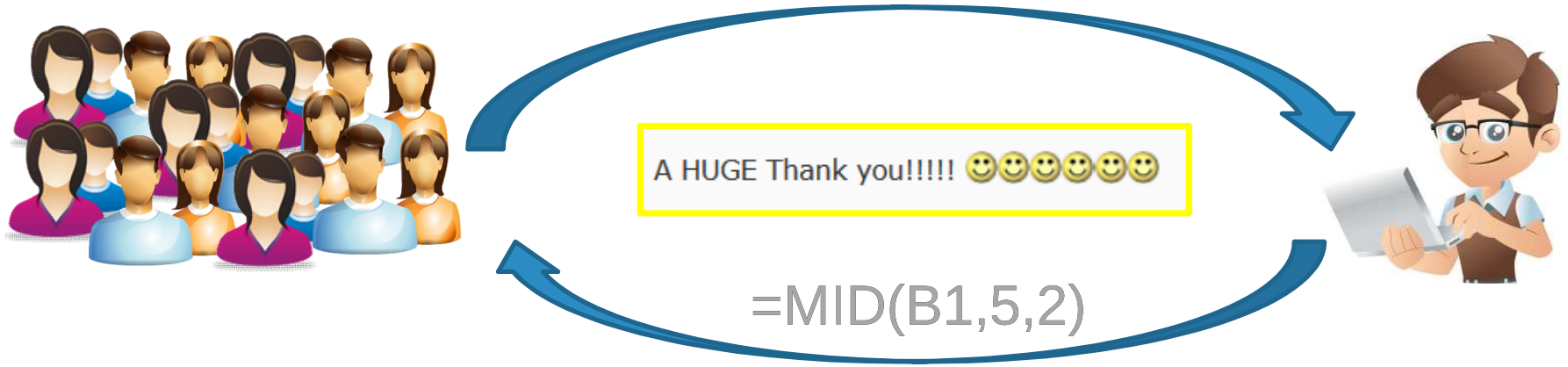
July 2018

Sumit Gulwani
Microsoft

Joint work with
many collaborators

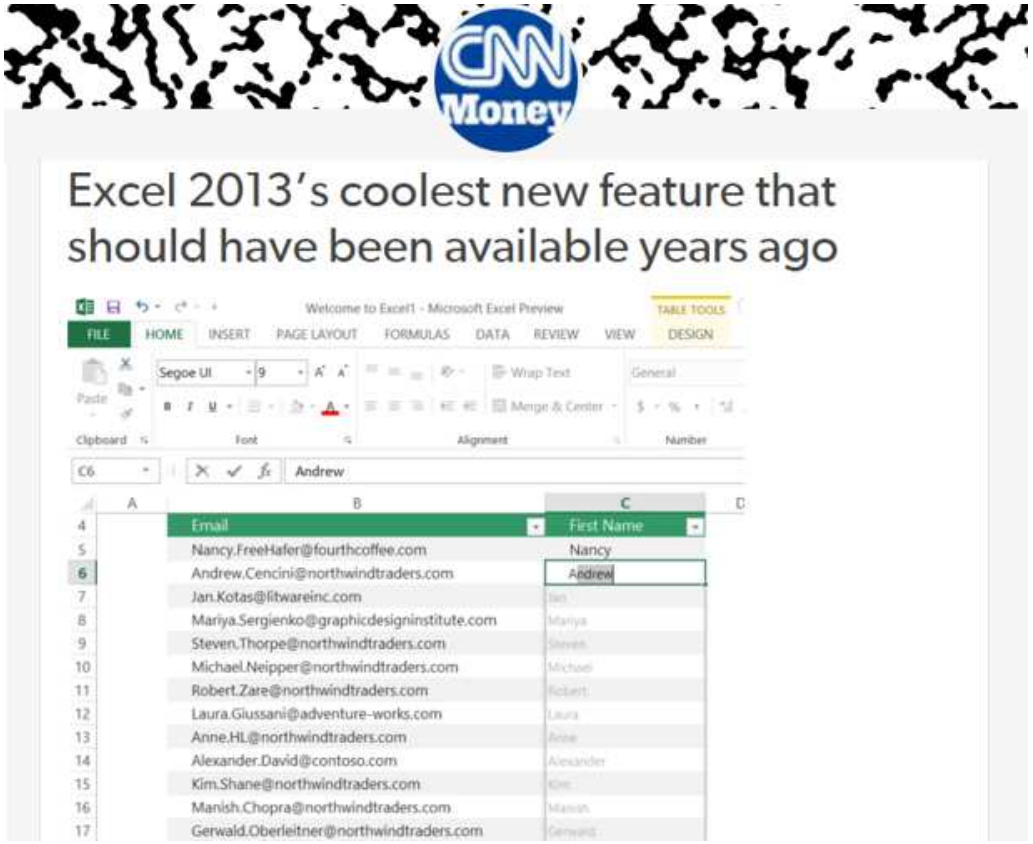
Example-based help-forum interaction

300_w30_aniSh_c1_b → w30
300_w5_aniSh_c1_b → w5



=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

Input	Output (round to 2 decimal places)
123.4567	123.46
123.4	123.40
78.234	78.23

Excel/C#: #.00
Python/C: .2f
Java: #.##

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

"Synthesizing Number Transformations from Input-Output Examples"
[CAV 2012] Rishabh Singh, Sumit Gulwani

Data Science Class Assignment

```
| style="text-align: center;"| {{Sort|01|[[Super Bowl I|I]]}}
|{{Dts|1967|January|15}}
| style="background:#d0e7ff;"|{{Sort|Green Bay Packers 01|[[1966 Green Bay Packers season|Green Bay Packers]]<sup>‡</sup>}}
| style="text-align: center;"| {{Sort|3510|35–10}}
| style="background:#fcc;"|{{Sort|Kansas City Chiefs 01|[[1966 Kansas City Chiefs season|Kansas City Chiefs]]<sup>^</sup>}}
|{{Sort|Los Angeles Memorial Coliseum 01|[[Los Angeles Memorial Coliseum]]}}
|{{Sort|Pasadena, California 01|[[Los Angeles]], [[California]]{{#tag:ref|Both [[Los Angeles, California|Los Angeles]] and [[Pasadena, Ca
| style="text-align: center;"| {{Sort|061946|61,946}}
| style="text-align: center;"|<ref>{{Cite journal |last=Maule |first=Tex |url=http://sportsillustrated.cnn.com/vault/article/magazine/MAG10
|-
| style="text-align: center;"| {{Sort|02|[[Super Bowl II|II]]}}
|{{Dts|1968|January|14}}
| style="background:#d0e7ff;"|{{Sort|Green Bay Packers 02|[[1967 Green Bay Packers season|Green Bay Packers]]<sup>‡</sup> (2)}}
| style="text-align: center;"| {{Sort|3314|33–14}}
| style="background:#fcc;"|{{Sort|Oakland Raiders 01|[[1967 Oakland Raiders season|Oakland Raiders]]<sup>^</sup>}}
|{{Sort|Orange Bowl 01|[[Miami Orange Bowl|Orange Bowl]]}}
|{{Sort|Miami, Florida 01|[[Miami]], [[Florida]]{{#tag:ref|[[Miami Gardens, Florida|Miami Gardens]] was incorporated as a [[sub
| style="text-align: center;"| {{Sort|075546|75,546}}
| style="text-align: center;"|<ref>{{Cite journal |url=http://aol.sportingnews.com/nfl/story/2008-01-15/super-bowl-2-lombardis-starr-rises
|-
| style="text-align: center;"| {{Sort|03|[[Super Bowl III|III]]}}<!---During the AFL-NFL merger, As the Colts moved over to the AFC, which
|{{Dts|1969|January|12}}
| style="background:#fcc;"|{{Sort|New York Jets 01|[[1968 New York Jets season|New York Jets]]<sup>^</sup>}}
| style="text-
| style="back
|{{Sort|Oran
|{{Sort|Miam
| style="text-
| style="text-
<includeonly
|-
| style="text-align: center;"| {{Sort|04|[[Super Bowl IV|IV]]}}
|{{Dts|1970|January|11}}
| style="background:#fcc;"|{{Sort|Kansas City Chiefs 02|[[1969 Kansas City Chiefs season|Kansas City Chiefs]]<sup>^</sup> (2)}}
| style="text-align: center;"| {{Sort|2307|23–7 }}
| style="background:#d0e7ff;"|{{Sort|Minnesota Vikings 01|[[1969 Minnesota Vikings season|Minnesota Vikings]]<sup>‡</sup>}}
|{{Sort|Tulane Stadium 01|[[Tulane Stadium]]}}
|{{Sort|New Orleans, Louisiana|[[New Orleans]], [[Louisiana]]}}
| style="text-align: center;"| {{Sort|080562|80,562}}
| style="text-align: center;"|<ref>{{Cite web |url=http://www.cbsnews.com/htdocs/sports/football/history/superbowl_04.html |title=Super Bowl History: Super E
```

```
I,1967,Green Bay Packers 01,35-10,Kansas City Chiefs 01,Los Angeles Memorial Coliseum
III,1969,New York Jets 01,16-7,Indianapolis Colts 01,Orange Bowl 02 IV,1970,Kansas Ci
V,1971,Indianapolis Colts 02,16-13,Dallas Cowboys 01,Orange Bowl 03 VI,1972,Dallas Co
VII,1973,Miami Dolphins 02,14-7,Washington Redskins 01,Los Angeles Memorial Coliseum 0
IX,1975,Pittsburgh Steelers 01,16-6,Minnesota Vikings 03,Tulane Stadium 03 X,1976,Pit
XI,1977,Oakland Raiders 02,32-14,Minnesota Vikings 04,Rose Bowl 01 XII,1978,Dallas Co
XIII,1979,Pittsburgh Steelers 03,35-31,Dallas Cowboys 05,Orange Bowl 05 XIV,1980,Pitt
1981,Oakland Raiders 03,27-10,Philadelphia Eagles 01,Louisiana Superdome 02 XVI,19
XVII,1983,Washington Redskins 02,27-17,Miami Dolphins 04,Rose Bowl 03 XVIII,1984,Oakl
XIX,1985,San Francisco 49ers 02,38-16,Miami Dolphins 05,Stanford Stadium 01 XX,1986,C
XXI,1987,New York Giants 01,39-20,Denver Broncos 02,Rose Bowl 04 XXII,1988,Washington
XXIII,1989,San Francisco 49ers 03,20-16,Cincinnati Bengals 02,Joe Robbie Stadium 01 X
```

```
cat superbowl.txt | awk '$1=$1' ORS=' ' | sed 's/|- |/\n|/g' | grep "^| style=\"t
ext-align: center;\"" | grep -v "Championship"
```

Table Reshaping

Bureau of I.A.	
Regional Dir.	Numbers
Niles C.	Tel: (800)645-8397
	Fax: (907)586-7252
Jean H.	Tel: (918)781-4600
	Fax: (918)781-4604
Frank K.	Tel: (615)564-6500
	Fax: (615)564-6701

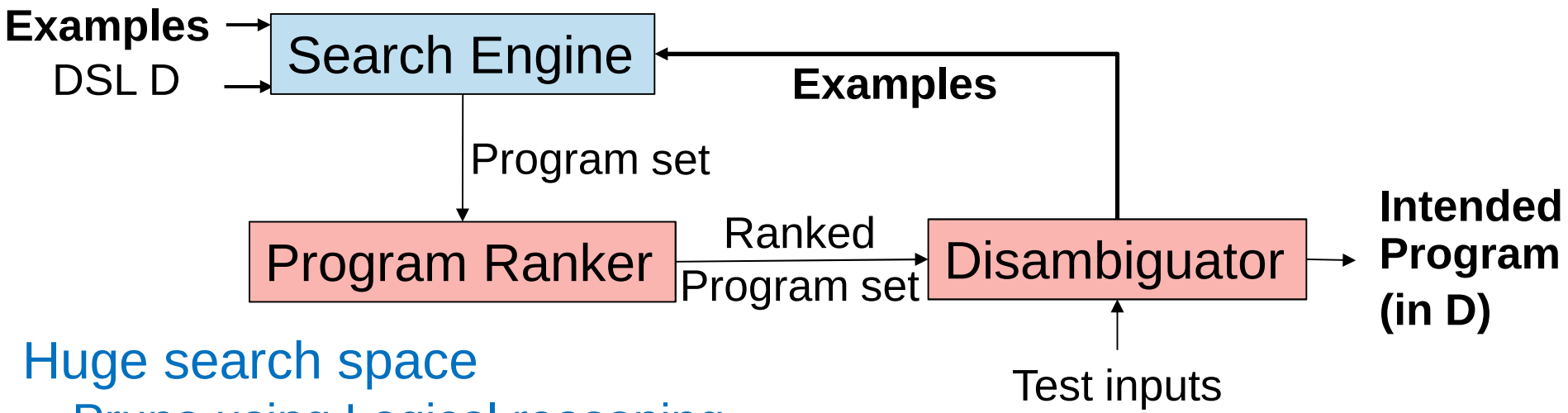
FlashRelate

From few examples of rows in output table

	Tel	Fax
Niles C.	(800)645-8397	(907)586-7252
Jean H.	(918)781-4600	(918)781-4604
Frank K.	(615)564-6500	(615)564-6701

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

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)

Flash Fill DSL

$$Tuple(String\ x_1, \ldots, String\ x_n) \rightarrow String$$

top-level expr $T := C \mid ifThenElse(B, C, T)$

condition-free expr $C := A \mid Concat(A, C)$

atomic expression $A := SubStr(X, P, P) \mid ConstantString$

input string $X := x_1 \mid x_2 \mid \ldots$

position expression $P := K \mid Pos(X, R_1, R_2, K)$

K^{th} position in X whose left/right side matches with R_1/R_2.

Search Idea 1: Deduction

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

ϕ is a Boolean constraint over (input state i ⇝ output value o)

Divide-and-conquer style problem reduction

$$[G \vDash \phi_1 \wedge \phi_2] = Intersect([G \vDash \phi_1], [G \vDash \phi_2])$$
$$= [G_1 \vDash \phi_2] \text{ where } G_1 = [G \vDash \phi_1]$$

Let $G := G_1 \mid G_2$

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

Search Idea 1: Deduction

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

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

Let $G := F(G_1, G_2)$

Let $F^{-1}(o)$ be $\{ (u, v), (u', v') \}$

$$[G \vDash (i \rightsquigarrow o)] = F([G_1 \vDash (i \rightsquigarrow u)], [G_2 \vDash (i \rightsquigarrow v)]) \mid F([G_1 \vDash (i \rightsquigarrow u')], [G_2 \vDash (i \rightsquigarrow v')])$$

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

Ranking Idea 1: Program Features

Input	Output
Vasu Singh	v.s.
Stuart Russell	s.r.

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

Ranking Idea 2: Output Features

Input	Output	Output of P1
[CPT-123	[CPT-123]	[CPT-123]
[CPT-456]	[CPT-456]	[CPT-456]]

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.

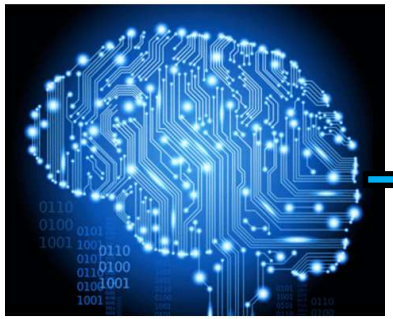
PL aspects

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

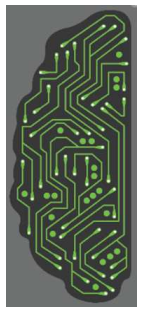
Heuristics that can be machine learned

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

ML in intelligent software creation

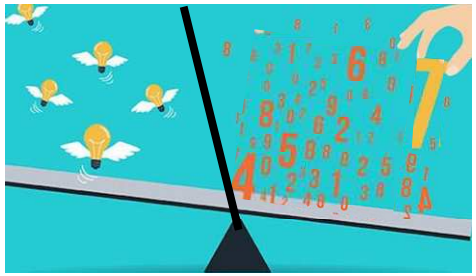


Intelligent software
(e.g., PBE component)



Logical
strategies

Creative
heuristics



Features

Model

Written by
developers

Can be learned
and maintained by
ML-backed runtime

Advantages

- Better models
- Less time to author
- Online adaptation, personalization

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