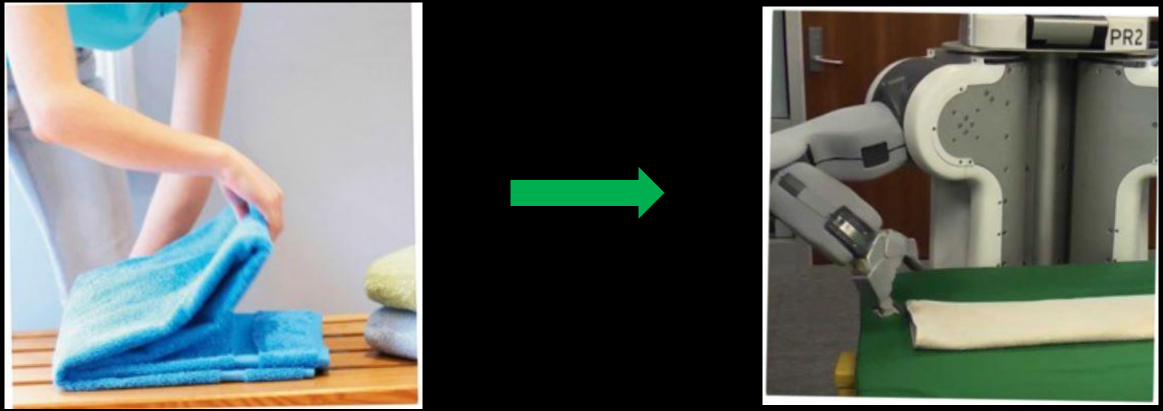
Programming by Examples



Sumit Gulwani

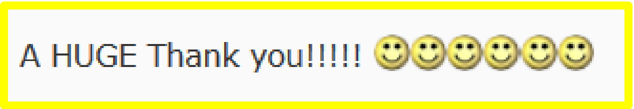
Microsoft

ECML/PKDD Conference

Sep 2019

Example-based help-forum interaction

300_w30_aniSh_c1_b → w30
300_w5_aniSh_c1_b → w5

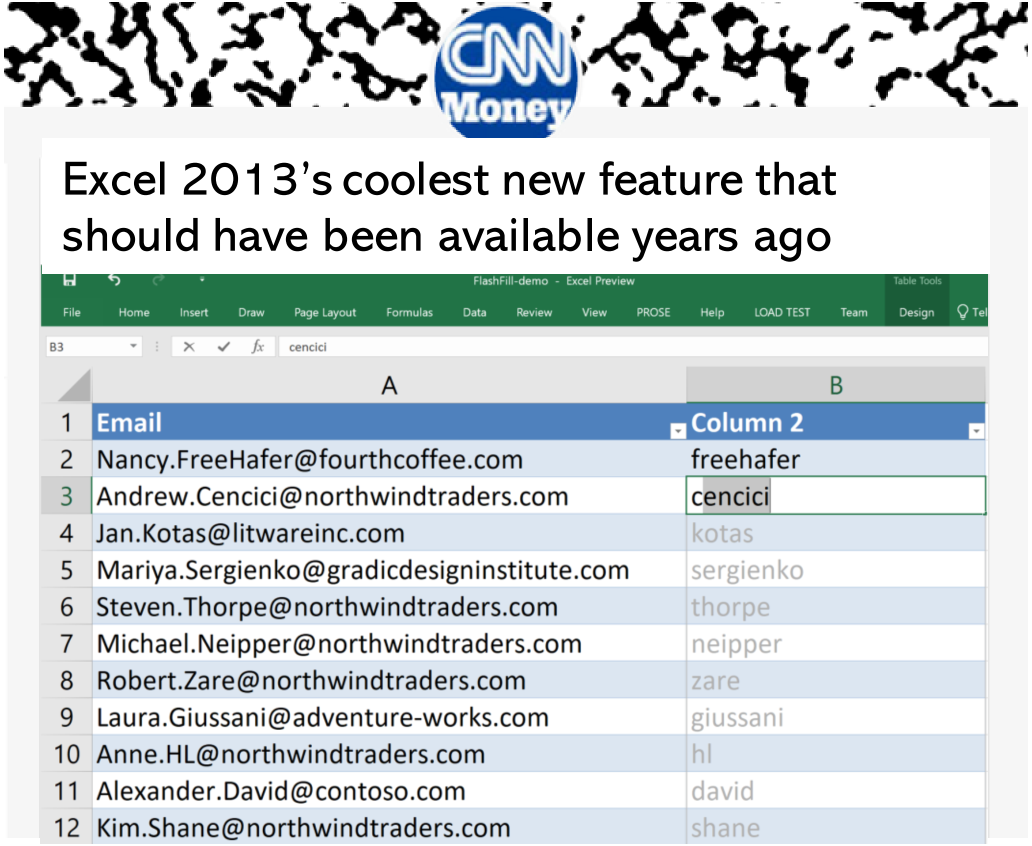


A HUGE Thank you!!!!! ☺☺☺☺☺☺

=MID(B1,5,2)

=MID(B1,FIND(“_”,$B:$B)+1,
FIND(“_”,REPLACE($B:$B,1,FIND(“_”,$B:$B),””))-1)

Flash Fill (Excel feature)

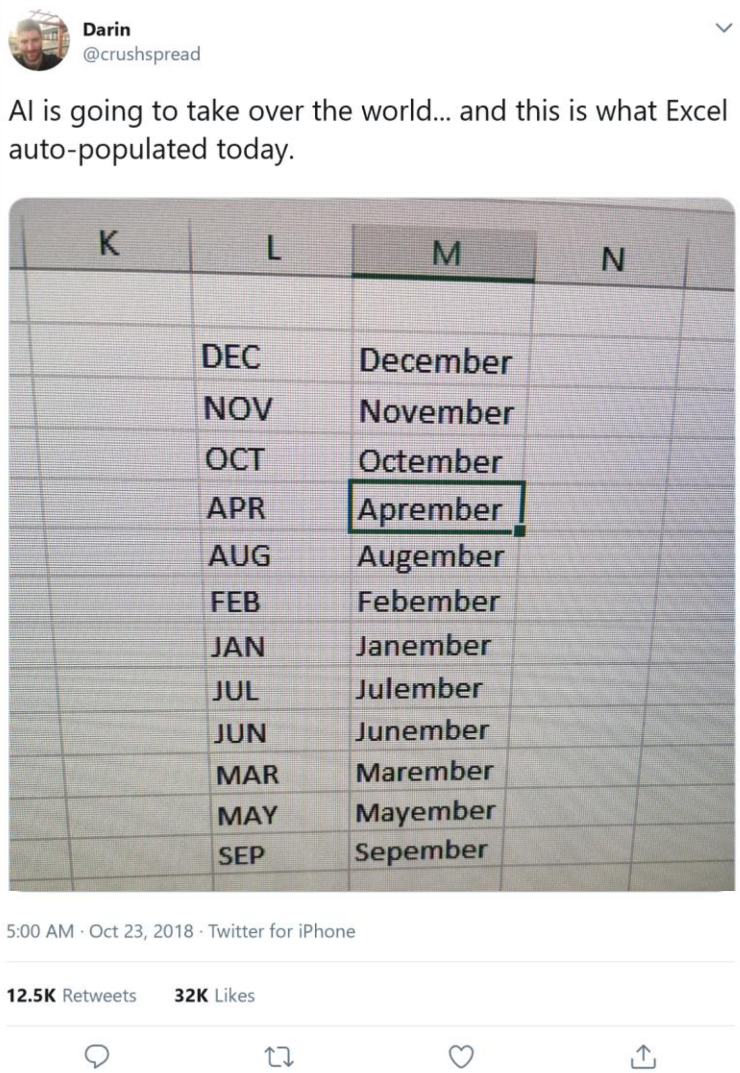


Excel 2013's coolest new feature that should have been available years ago

B3 cencici

	A	B
1	Email	Column 2
2	Nancy.FreeHafer@fourthcoffee.com	freehafer
3	Andrew.Cencici@northwindtraders.com	cencici
4	Jan.Kotas@litwareinc.com	kotas
5	Mariya.Sergienko@gradicdesigninstitute.com	sergienko
6	Steven.Thorpe@northwindtraders.com	thorpe
7	Michael.Neipper@northwindtraders.com	neipper
8	Robert.Zare@northwindtraders.com	zare
9	Laura.Giussani@adventure-works.com	giussani
10	Anne.HL@northwindtraders.com	hl
11	Alexander.David@contoso.com	david
12	Kim.Shane@northwindtraders.com	shane

"Automating string processing in spreadsheets using input-output examples"
[POPL 2011] Sumit Gulwani



Darin
@crushspread

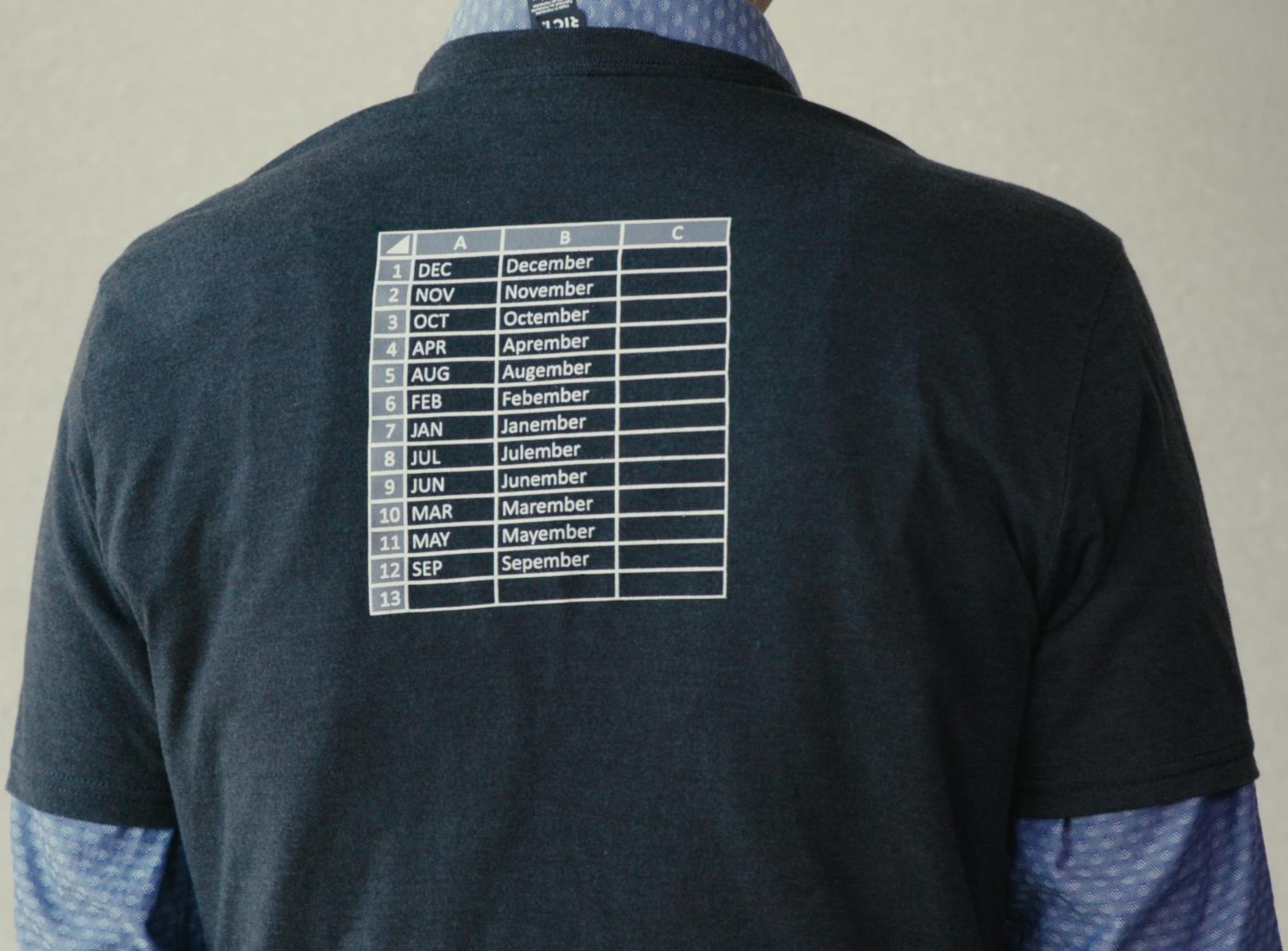
AI is going to take over the world... and this is what Excel auto-populated today.

K	L	M	N
	DEC	December	
	NOV	November	
	OCT	Octember	
	APR	Aprember	
	AUG	Augember	
	FEB	Febember	
	JAN	Janember	
	JUL	Julember	
	JUN	Junember	
	MAR	Marember	
	MAY	Mayember	
	SEP	Sepember	

5:00 AM · Oct 23, 2018 · Twitter for iPhone

12.5K Retweets **32K** Likes

PROgram Synthesis using Examples
"AI is going to take over the world..."
Octember 23, 2018



	A	B	C
1	DEC	December	
2	NOV	November	
3	OCT	Octember	
4	APR	Aprember	
5	AUG	Augember	
6	FEB	Febember	
7	JAN	Janember	
8	JUL	Julember	
9	JUN	Junember	
10	MAR	Marember	
11	MAY	Mayember	
12	SEP	Sepember	
13			

Number, DateTime Transformations

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

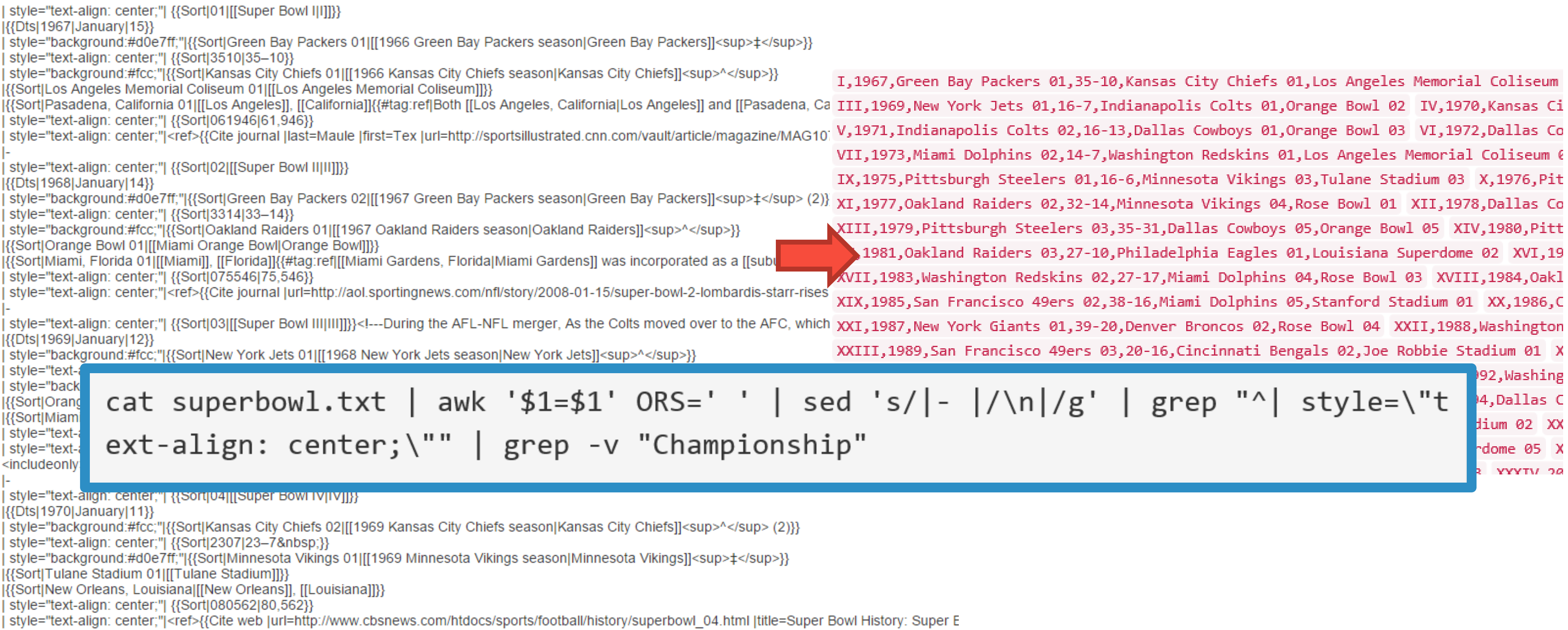
Excel/C#:	#.00
Python/C:	.2f
Java:	#.##

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

[CAV 2012] *“Synthesizing Number Transformations from Input-Output Examples”; Singh, Gulwani*

[POPL 2015] *“Transforming Spreadsheet data types using Examples”; Singh, Gulwani*

Table Extraction



```
| 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 [[subu
| 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|Orang
|{{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&nbsp;}}
| 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
```

```
cat superbowl.txt | awk '$1=$1' ORS=' ' | sed 's/|- |/\n|/g' | grep "^| style=\"text-align: center;\"" | grep -v "Championship"
```

```
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
92,Washing
4,Dallas C
dium 02   XX
rdome 05   X
XXXIV 20
```

"*FlashExtract: A Framework for data extraction by examples*"
[PLDI 2014] Vu Le, Sumit Gulwani

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 →

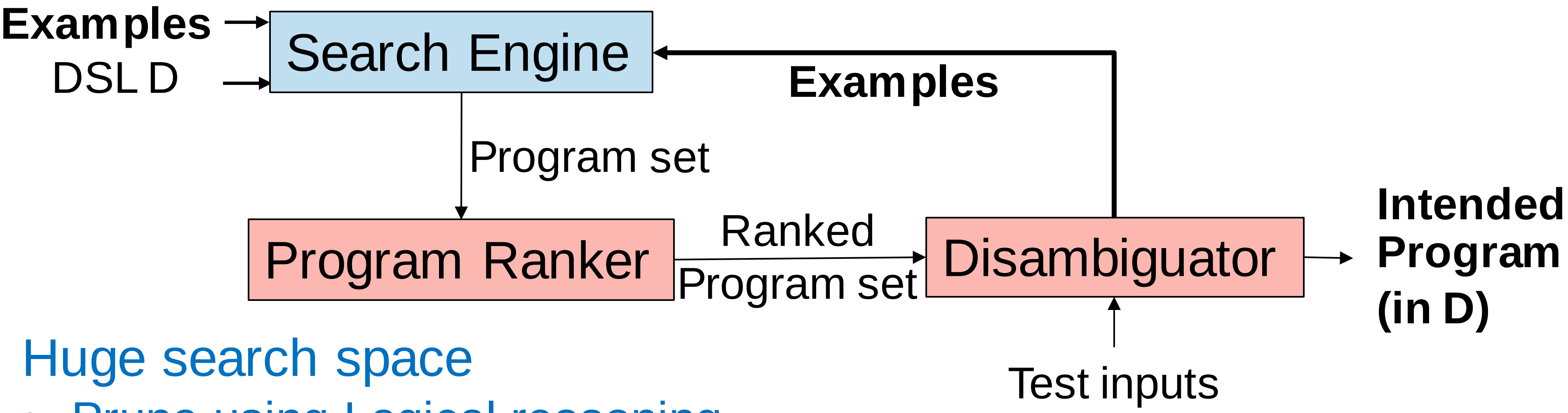
From few examples of rows in output table

	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.

"FlashRelate: Extracting Relational Data from Semi-Structured Spreadsheets Using Examples"
[PLDI 2015] Dan Barowy, Sumit Gulwani, Ted Hart, Ben Zorn

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

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

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.

“Automating string processing in spreadsheets using input-output examples”
[POPL 2011] Sumit Gulwani

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 $:= G_1 \mid G_2$

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

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

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

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

$$
\begin{aligned}
[G \vDash (i \rightsquigarrow o)] = \quad & F([G_1 \vDash (i \rightsquigarrow u)], [G_2 \vDash (i \rightsquigarrow v)]) \\
\mid \ & F([G_1 \vDash (i \rightsquigarrow u')], [G_2 \vDash (i \rightsquigarrow v')])
\end{aligned}
$$

"*FlashMeta: A Framework for Inductive Program Synthesis*"
[OOPSLA 2015] Alex Polozov, Sumit Gulwani

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

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

"Predicting a correct program in Programming by Example"
[CAV 2015] Rishabh Singh, Sumit Gulwani

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

"Learning to Learn Programs from Examples: Going Beyond Program Structure"
[IJCAI 2017] Kevin Ellis, Sumit Gulwani

Disambiguation

Communicate actionable information back to user.

Program-based disambiguation

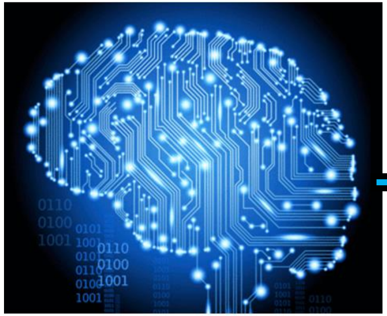
- Enable effective navigation between top-ranked programs.
- Highlight ambiguity based on *distinguishing inputs*.

Heuristics that can be machine learned

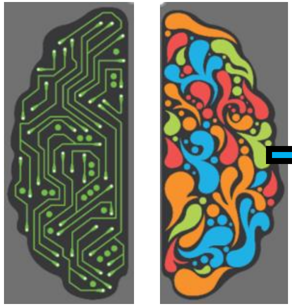
- Highlight ambiguity based on clustering of inputs/outputs.
- When to stop highlighting ambiguity?

[UIST '15] "*User Interaction Models for Disambiguation in Programming by Example*"
[OOPSLA '18] *"FlashProfile: A Framework for Synthesizing Data Profiles"*

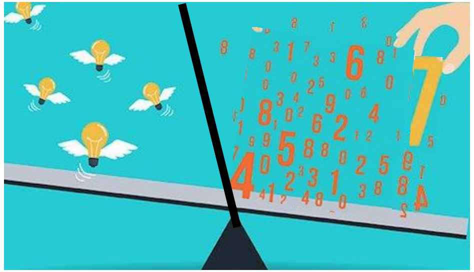
ML in PBE



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

"Programming by Examples: PL meets ML"
[APLAS 2017] Sumit Gulwani, Prateek Jain

Mode-less Synthesis

Non-intrusively watch, learn, and make suggestions

Advantages: Usability, Avoids Discoverability

Applications: Document Editing, Code Refactoring, Robotic Process Automation

Key Idea: Identify related examples within noisy action traces

“On the Fly Synthesis of Edit Suggestions”
[OOPSLA 2019] Miltner, Gulwani, Le, Luang, Radhakrishna, Soares, Tiwari, Udupa

Predictive Synthesis

Synthesis of intended programs from just the input.

Predictive Synthesis : PBE :: Unsupervised : Supervised ML

Applications: Tabular data extraction, Join, Sort, Split

Key Idea: Structure inference over inputs

"Automated Data Extraction using Predictive Program Synthesis"
[AAAI 2017] Mohammad Raza, Sumit Gulwani

Synthesis of Readable Code

Synthesis in target language of choice.

- Python, R, Scala, PySpark

Advantages:

- Transparency
- Education
- Integration with existing workflows in IDEs, Notebooks

Challenges: Quantify readability, Quantitative PBE

Key Idea: Observationally-equivalent (but non-semantic preserving) transformation of an intended program

Program Synthesis meets Notebooks

A match made in heaven!

PS can synthesize small code fragments. Sufficient for notebook cell-based programming.

PS can synthesize code in different languages.
A good solution for polyglot challenge in notebooks.

PS needs interactivity. Notebooks provide that.

Other Topics 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]
- Intent specification:
  - Natural language [Huang et.al., NAACL-HLT 2018; Gulwani, Marron SIGMOD 2014, Shin et al. NeurIPS 2019]
  - Conversational pair programming
- Applications:
  - Super-optimization for model training/inference
  - Personalized Learning [Gulwani; CACM 2014]

Conclusion

Program Synthesis: key to next-generational programming

- Future: Multi-modal programming with Examples and NL
- 100x more programmers
- 10-100x productivity increase in several domains.

Next-generational AI techniques under the hood

- Logical Reasoning + Machine Learning

Questions/Feedback: Contact me at **sumitg@microsoft.com**