transform by example
extract by example
learn by example
fix by example
search by example
analyze by example
generate by example
debug by example
predict by example
query by example
plot by example
transform by example
extract by example
learn by example
fix by example
search by example
analyze by example
generate by example
debug by example
predict by example
query by example
plot by example
transform by example
extract by example
learn by example
fix by example
search by example
analyze by example
generate by example
debug by example

x by example

Azza Abouzied, NYU Abu Dhabi
azza@nyu.edu

Started PhD in Database research

A tutorial on Visualization at VLDB by Joe Hellerstein & Jeff Heer

Retargeting my research at the 99%

Joined NYU

HadoopDB

InvisibleLoading

Hadapt

DataPlay

SEER

Qetch

WhyFlow

Synner

Texture

PackageBuilder

2008 2009 2012 2013 2018

@Yale, New Haven

@UC Berkeley

@Happy Island, Abu Dhabi

About me

Projects, Places & Events

How would you describe “furniture”?

A thought experiment

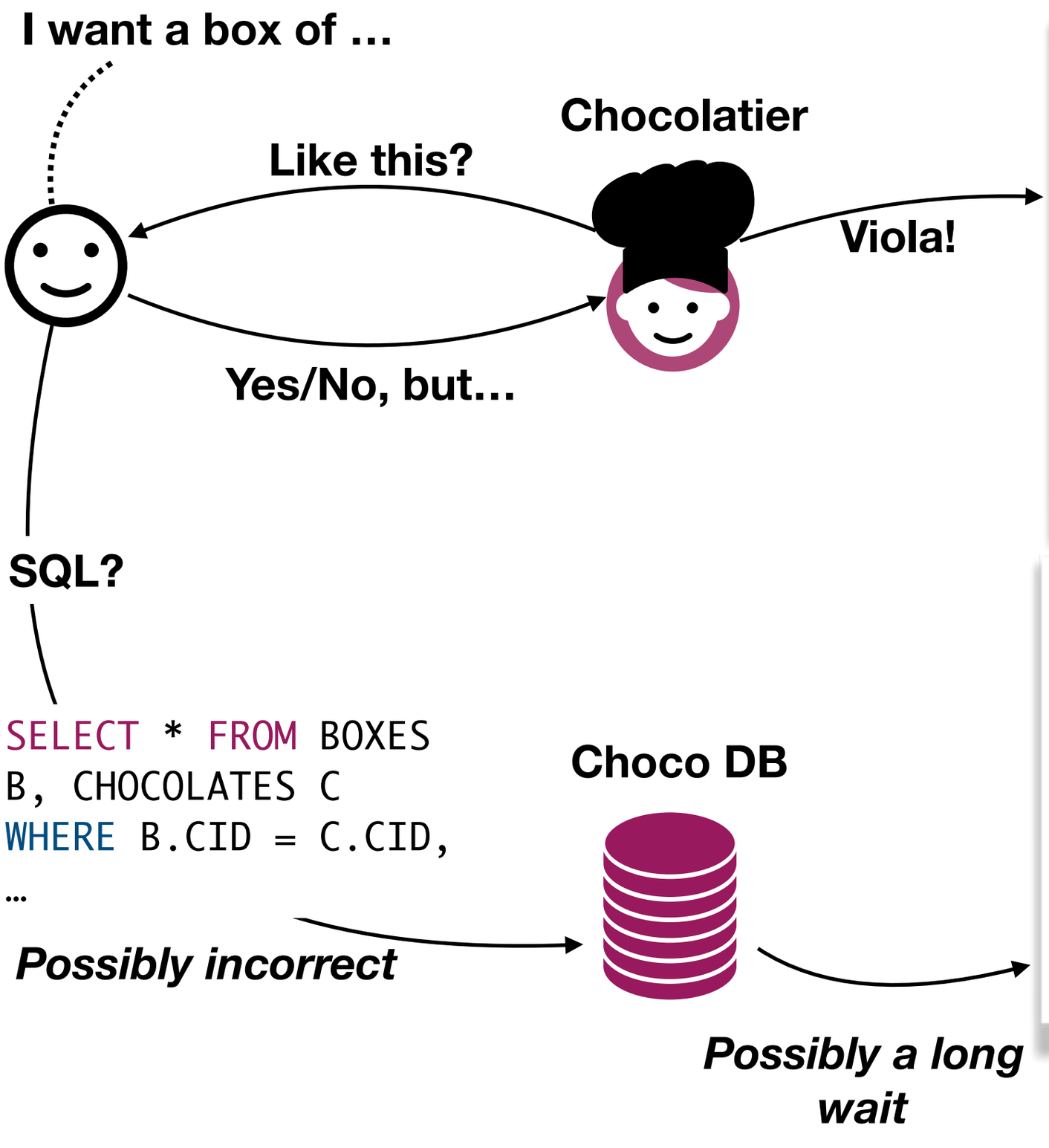
If you thought of an example,
you are not alone

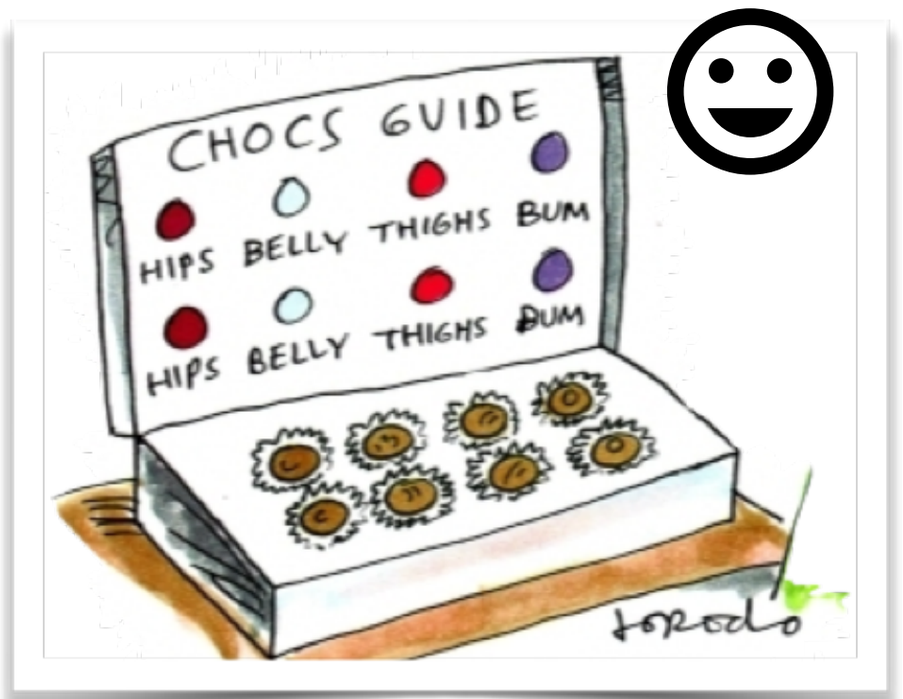
Prototyping
Exemplar-based reasoning
Recognition-primed decision making

How can we improve how we communicate with our data tools?

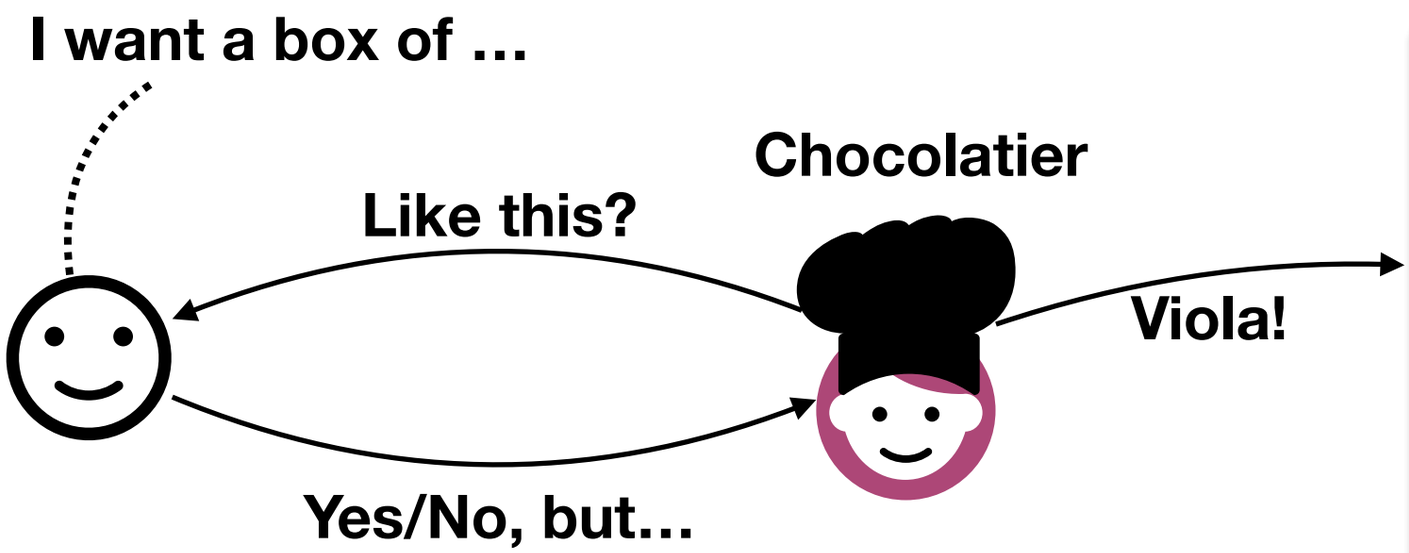
What are example-driven interfaces?

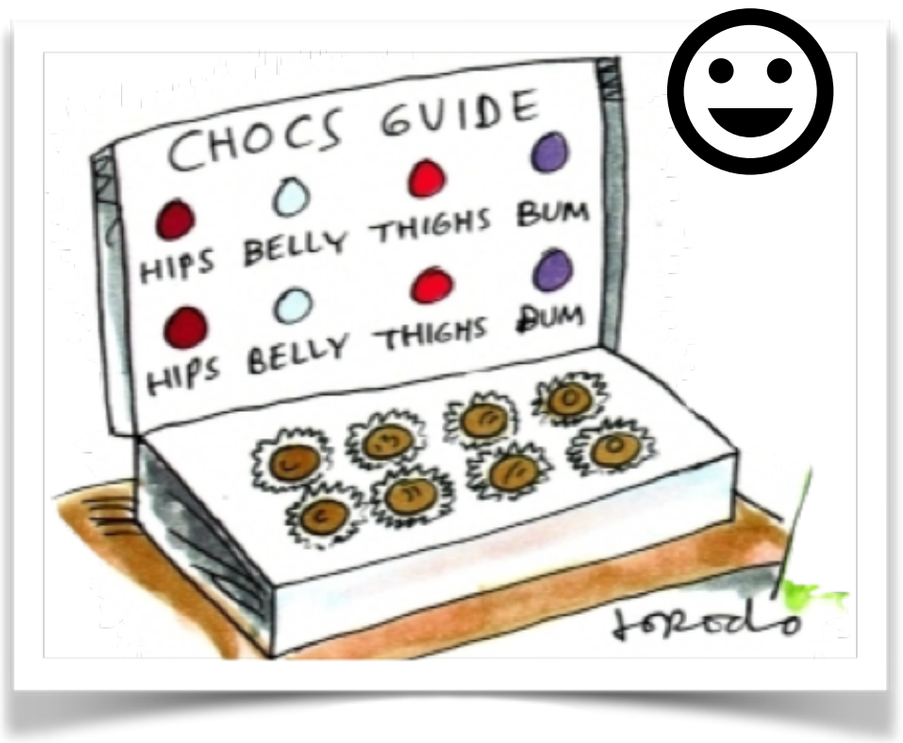
Suppose you want to buy a box of chocolates





Suppose you want to buy a box of chocolates





EDIs mimic human interactions: they allow examples of (un)expected behavior, which can be *underspecified* or *ambiguous*, and work towards a precise specification of behavior through further human interaction such as requesting more examples, counter-examples, partial specifications, constraints, etc.

EDIs can support a variety of data tasks: extraction, transformation, visualization, querying, analysis, debugging, generation, etc.

What
are EDIs?

Why
now?

How
to build them?

When
does it work?

Where
do we go from here?

Why is now the right time for example-driven interfaces?

A confluence of many maturing research areas

Query by Example

by MOSHÉ M. ZLOOF
IBM T. J. Watson Research Center
Yorktown Heights, New York

1975

INTRODUCTION

In the last few years we have witnessed a trend to appeal to the non-professional user who has little or virtually no computer or mathematical background.

The 'Query by Example' Language is an attempt in that direction. It operates on a relational Model of data as was introduced by Codd [1-5].

In this paper we deal only with normalized relations [1]. A relation is normalized if each of its domains is simple, i.e., no domain is itself a relation.

A normalized relation can be viewed as a table of n columns and a varying number of rows as illustrated in Figure I. Three properties of normalized relations are noteworthy to mention:

1. ALL rows of the table are distinct.
2. The ordering of the rows is immaterial.
3. The ordering of the columns is immaterial provided each has a distinct name.

EMP	NAME	SALARY	MANAGER	DEPARTMENT
	ANDERSON	8K	SMITH	TOY
	MORGAN	10K	LEE	COSMETICS
	.			
	.			
	.			

trations of queries and their answers, each illustration followed by a discussion to point out major features. The illustrations get progressively more complex until the whole scope of the Language is covered. In so doing, a user dealing with "simple" queries needs to study the system *only* to that point of complexity which is compatible with the level of sophistication required within the domain of those queries.

Furthermore, although the introduction of the concepts through illustrative examples reduces somewhat from the rigor of mathematical formulation through definitions, it is—in our opinion—more appealing to the casual user, which is one of the major aspects of Query by Example.

Most of the queries are drawn from the following tables (relations), which are part of a department store data base.

EMP (NAME, SAL, MGR, DEPT)
SALES (DEPT, ITEM)
SUPPLY (SUPPLIER, ITEM)
TYPE (ITEM, COLOR, SIZE)

—The EMP Table specifies the name, salary, manager and department of each employee.
—The SALES Table is a listing of the items sold by departments.
—The SUPPLY Table is a listing of the items supplied by suppliers.
—The TYPE Table describes each item by color and size.

At this point we are assuming that these tables are made available to the user upon calling them by name. In a sub-

1 Program Synthesis

Circuit Synthesis

1957
Alonzo Church. ***Application of recursive arithmetic to the problem of circuit synthesis***. In Summaries of Talks Presented at the Summer Institute for Symbolic Logic, Cornell University.

Solver-backed Synthesis

2008
Armando Solar-Lezama. ***Program Synthesis by Sketching***. PhD Thesis. UC-Berkeley

PBE is mainstream: FlashFill in Excel

2017
S. Gulwani, O. Polozov and R. Singh. ***Program Synthesis***. Foundations and Trends in Programming Languages, vol. 4, no. 1-2

2 Mixed Initiative User Interfaces

1997

1999
Eric Horvitz. ***Principles of mixed-initiative user interfaces***. CHI '99. ACM

2007

TRIFACTA

2015
Jeffrey Heer, Joseph M. Hellerstein, Sean Kandel. ***Predictive Interaction for Data Transformation.*** CIDR'15

3 Active Learning

Learning Theory: Membership Questions, Teaching Dimension, …

1987
Dana Angluin. ***Learning regular sets from queries and counterexamples***. Inf. Comput. 75, 2

Crowd-sourced & Function Labeling

2017


Christopher Ré. ***Software 2.0 and Snorkel: Beyond Hand-Labeled Data***. KDD '18.

4 Causality & Explanations

Lineage & Provenance in Databases

1748 — 2000
David Hume: *We may define a cause to be an object followed by another, and where all the objects, similar to the first, are followed by objects similar to the second. Or, in other words, where, if the first object had not been, the second never had existed."*

2005
Joseph Y. Halpern & Judea Pearl. ***Causes and explanations: A structural-model approach. Part I: Causes.*** British Journal for the Philosophy of Science 56 (4)

Causality & Explanations

2010
Meliou et al. ***Why so? or Why no? Functional Causality for Explaining Query Answers.*** MUD

5 Automatic, Declarative, Data Visualization

2003

2011

1987
Jock Mackinlay. ***Automating the Design of Graphical Presentations of Relational Information.*** ACM Transactions on Graphics 5(2).

1984
Cleveland & McGill. ***Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods***. Journal of the American Statistical Association 79(387)

2010
Heer & Bostock. ***Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design.*** CHI

2018
Moritz et al. ***Formalizing Visualization Design Knowledge as Constraints: Actionable and Extensible Models in Draco***. InfoVis

2016
Satyanarayan et al. ***Vega-Lite: A Grammar of Interactive Graphics***. InfoVis

How to build example-driven data tools?

An example recipe

The Dimensions of Example-Driven Interfaces

From Sumit Gulwani's Cookbook - Dimensions of Program Synthesis

Intent Specification

Inputs, Outputs
Positive & Negative
Program Sketch

Search Space

Scope & define your tasks
Create a syntactic bias
DSL
(Invertible) Operators
Templates

Search Techniques

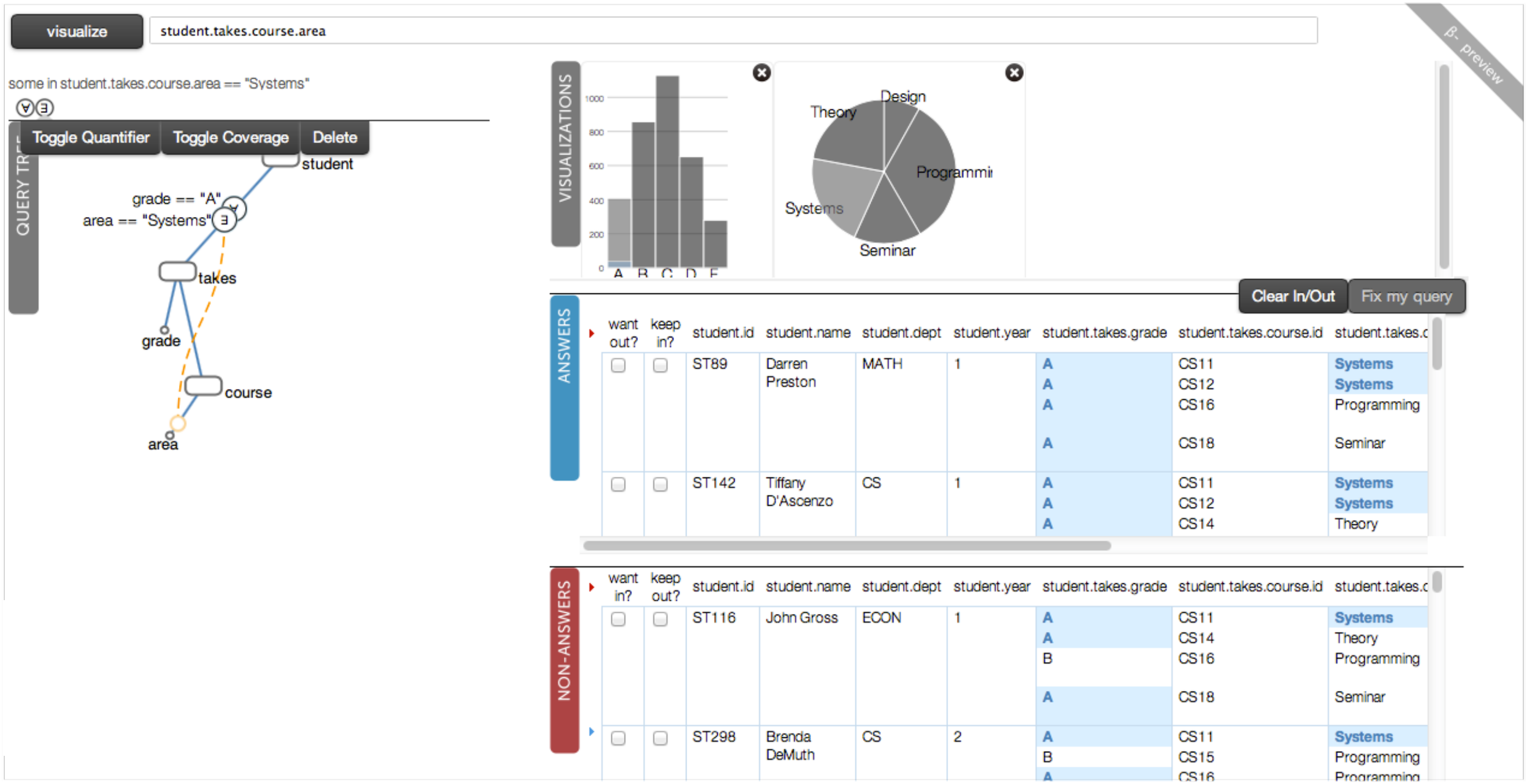
Version Space Algebras
SMT-guided search
Brute-force search

Ambiguity Resolution

Ranking
Distinguishing Inputs
Exposing Semantics

When do example-driven data tools work?

A few illustrative examples from my research

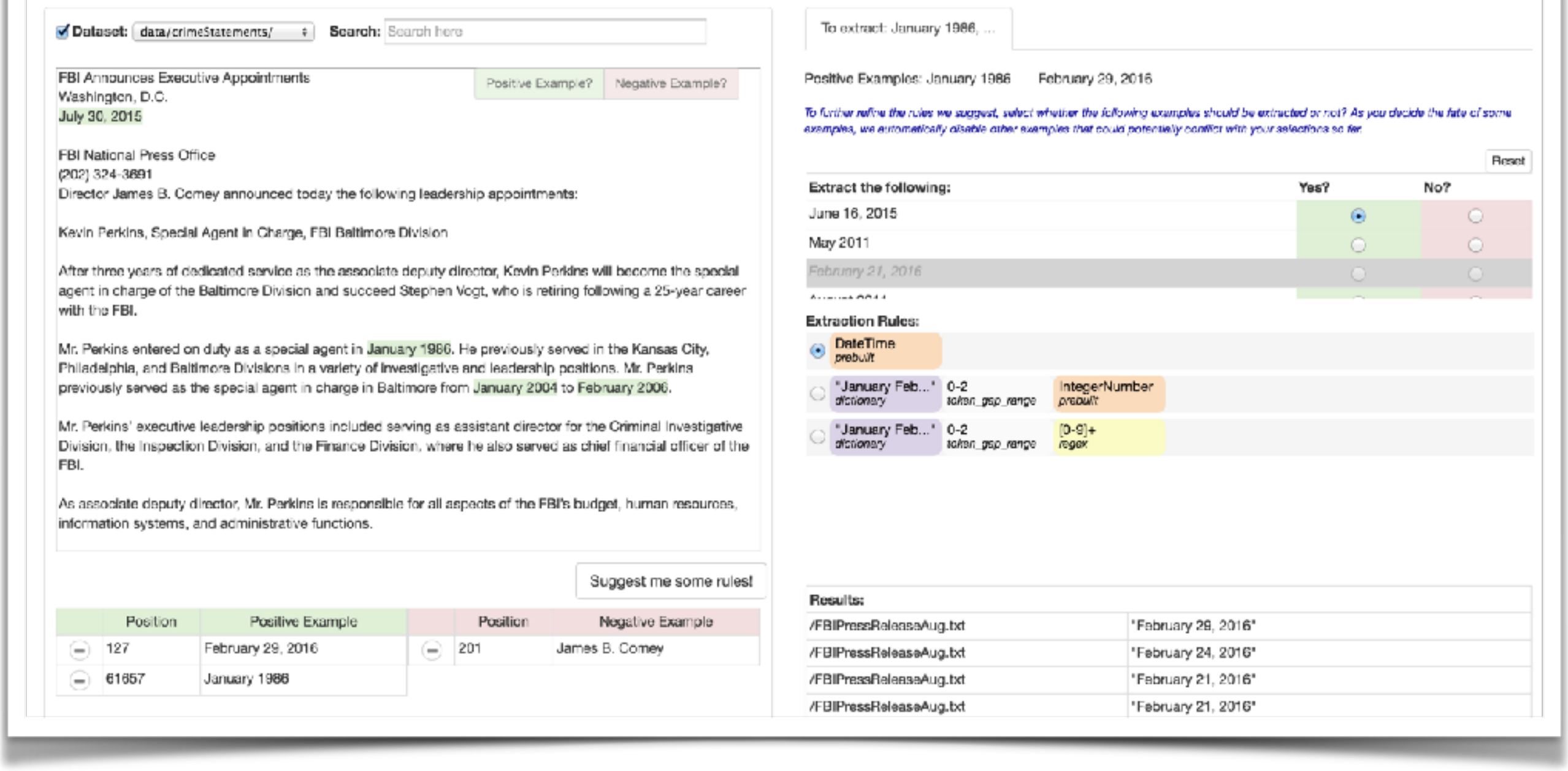


DataPlay

Example-driven database querying

Abouzied et al. *DataPlay: Interactive Tweaking and Example-driven Correction of Graphical Database Queries.* UIST 2012

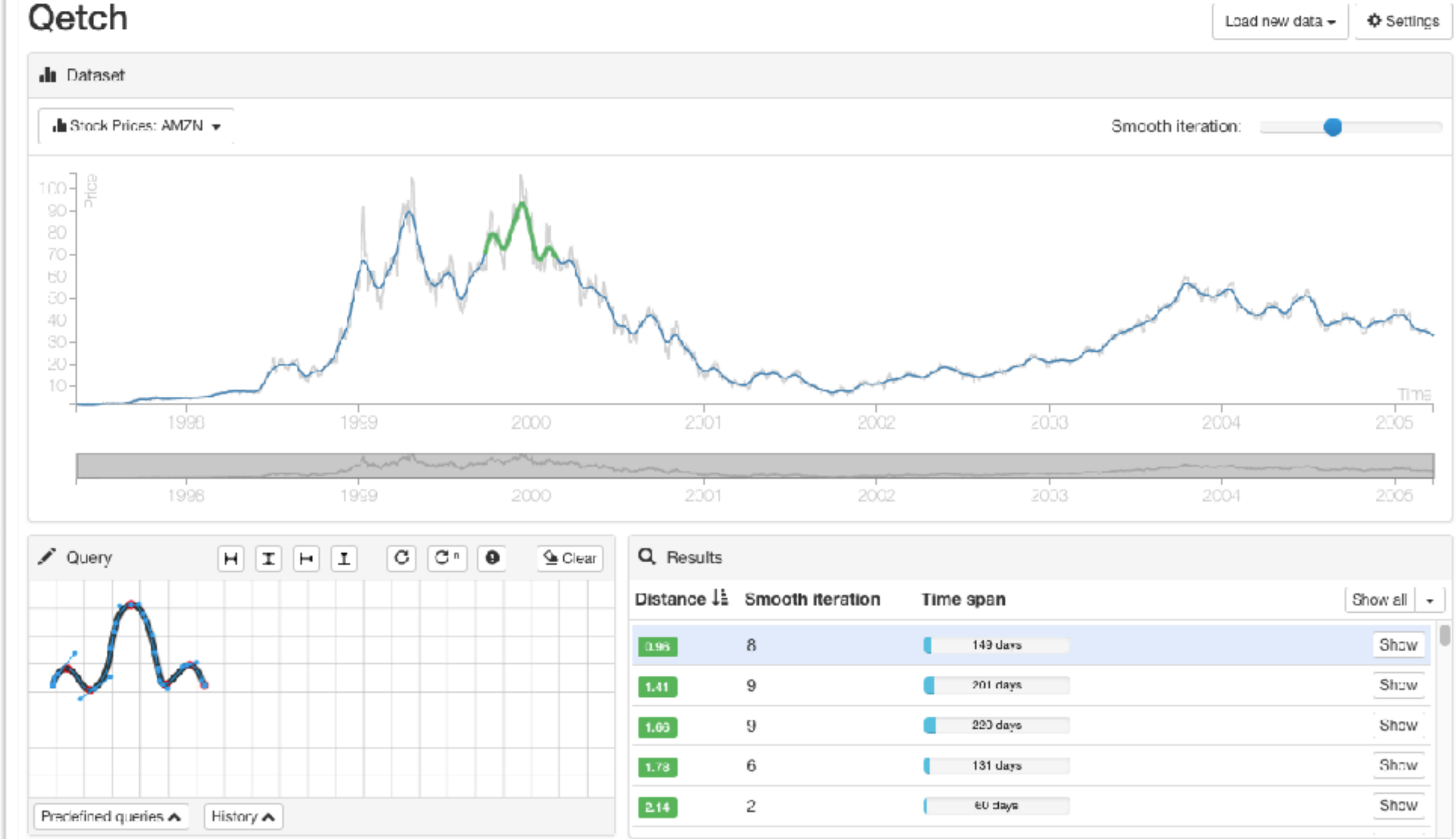
Abouzied et al. *Learning and verifying quantified boolean queries by example.* PODS 2013



SEER

Example-driven data extraction from text

Maeda Hanafi, Azza Abouzied, Laura Chiticariu, and Yunyao Li. *SEER: Learning Information Extraction Rules from User-Specified Examples.* CHI 2017



Qetch

Time series querying with hand-drawn sketches

Miro Mannino, Azza Abouzied. *Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches.* CHI 2018 - **Best Paper Award**

When do example-driven data tools fail?

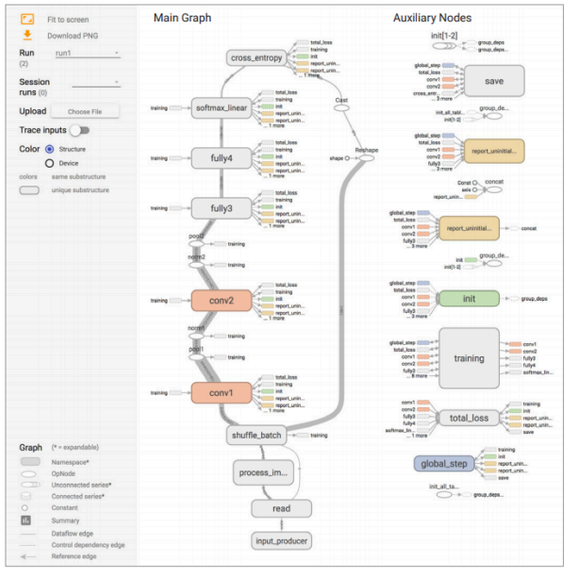
A case-study on debugging data processing pipelines by example

Where do we go from here?

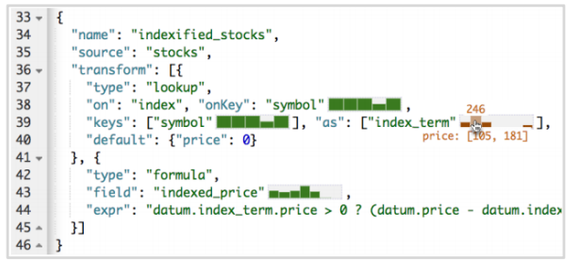
Some parting thoughts on the future of this research

Visualization Research

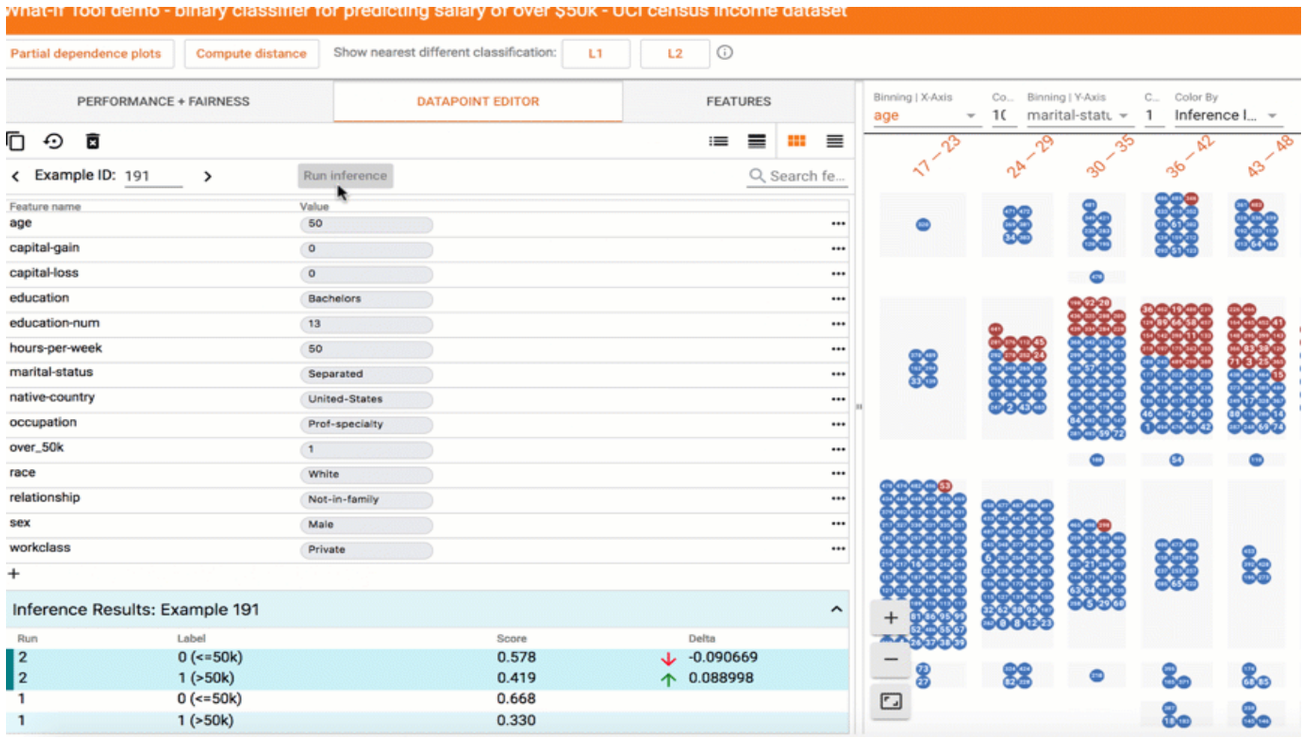
Exciting opportunities at the interplay of visualizing program artifacts & data



Wongsuphasawat, Kanit, Daniel Smilkov, James Wexler, Jimbo Wilson, Dandelion Mané, Doug Fritz, Dilip Krishnan, Fernanda B. Viégas, and Martin Wattenberg. ***Visualizing dataflow graphs of deep learning models in TensorFlow***. IEEE transactions on visualization and computer graphics 24, no. 1 (2018): 1-12.



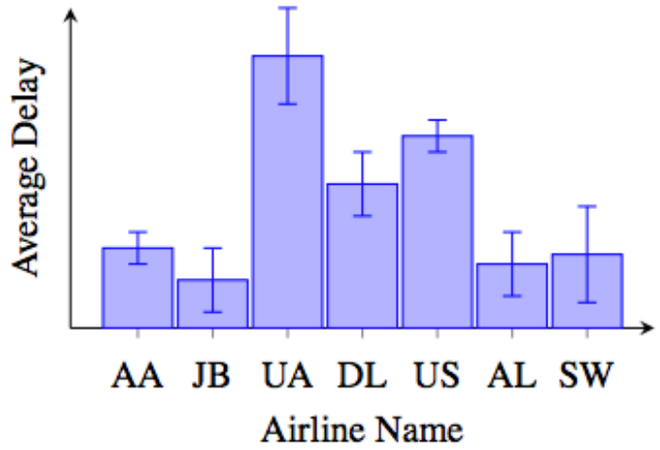
Jane Hoffswell, Arvind Satyanarayan, Jeffrey Heer. ***Augmenting Code with In Situ Visualizations to Aid Program Understanding***. CHI '18.



The What-If Tool: Code-Free Probing of Machine Learning Models. Google AI

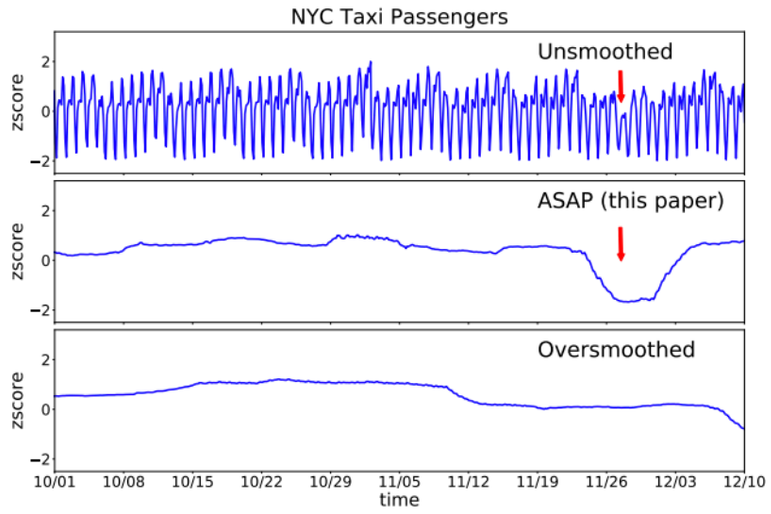
Data Research

Open up the database: The human interactions should drive the (re)design of abstractions and operations. Exciting opportunities in incremental & interactive querying and beyond relational data & queries



Kim et al. *Rapid sampling for visualizations with ordering guarantees*. VLDB '15

Siddiqui et al. *Effortless Visual Data Exploration with Zenvisage: An Interactive and Expressive Visual Analytics System*. VLDB '17



Kexin Rong, Peter Bailis. *ASAP: Prioritizing Attention via Time Series Smoothing*, VLDB '17

Bailis et al. *MacroBase: Prioritizing Attention in Fast Data*, SIGMOD 2017.

PaQL syntax specification

SELECT PACKAGE (∗|**column_name** [,...]) [AS] **package_name**
FROM **relation_name** [AS] **relation_alias**
[REPEAT **repeat**] [,...]
[WHERE **w_expression**]
[SUCH THAT **st_expression**]
[(MINIMIZE|MAXIMIZE) **obj_expression**]

PaQL query for Example 1

Q: SELECT **PACKAGE** (∗) AS P
FROM Recipes R **REPEAT** 0
WHERE R.gluten = 'free'
SUCH THAT COUNT (P.∗) = 3 AND
SUM(P.kcal) BETWEEN 2.0 AND 2.5
MINIMIZE SUM(P.sat_fat)

Brucato, Matteo, Azza Abouzied, and Alexandra Meliou. *Package queries: efficient and scalable computation of high-order constraints*. The VLDB Journal 2018

Thank you

Can’t wait to hear your thoughts, comments or questions.