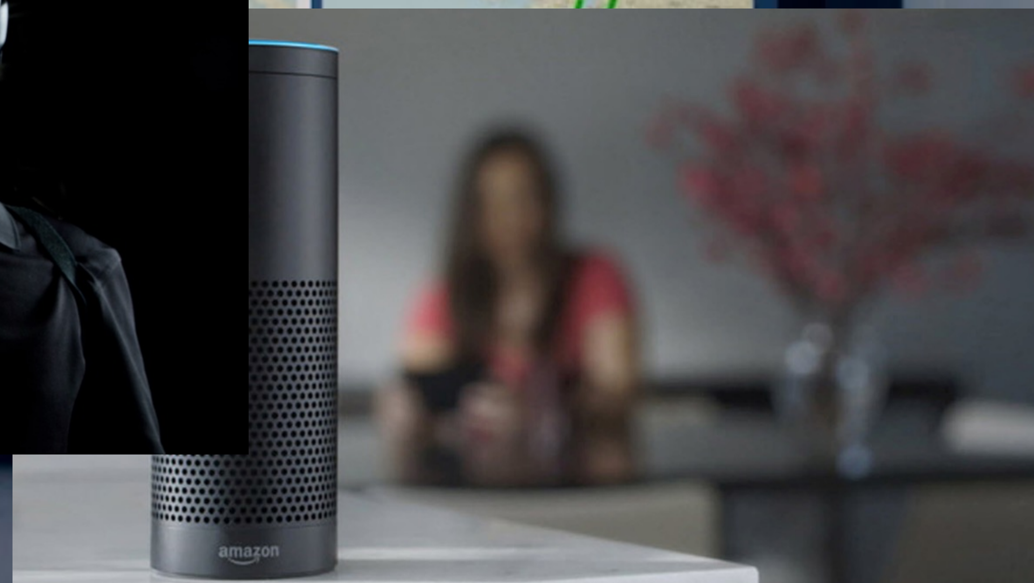
TOWARDS DEMOCRATIZING DATA SCIENCE

CARSTEN BINNIG

DATA MANAGEMENT LAB, TU DARMSTADT


Minority Report (2002)
VISION: DATA SCIENCE IN THE FUTURE

TODAY’S END USER DEVICES



… AND THE BIG DATA AND AI SYSTEMS?

Flink

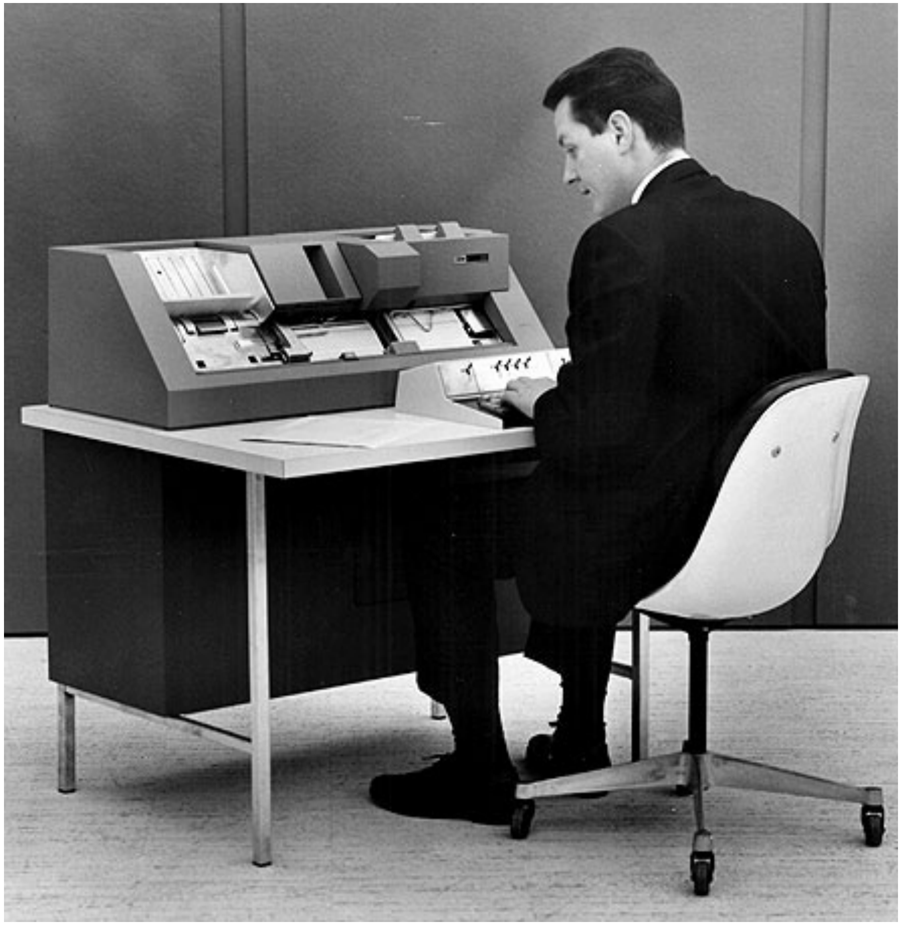
mxnet

Spark

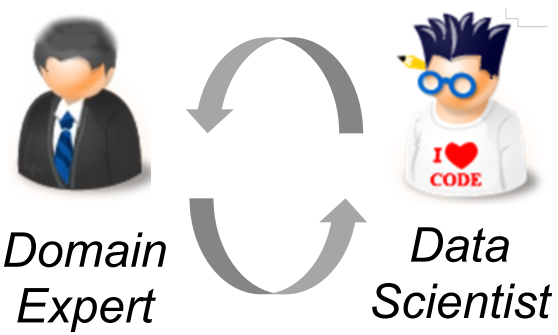
hadoop

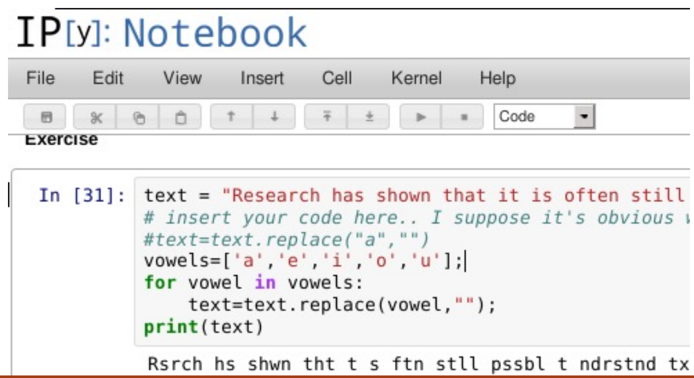
AsterixDB

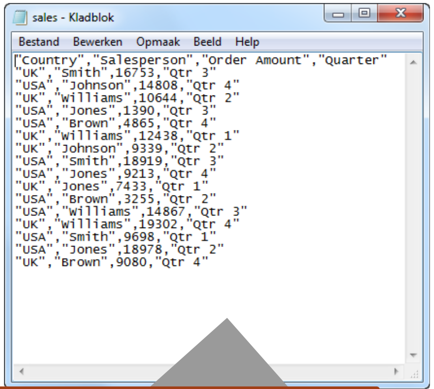
TensorFlow



WHAT ARE THE PAIN POINTS?



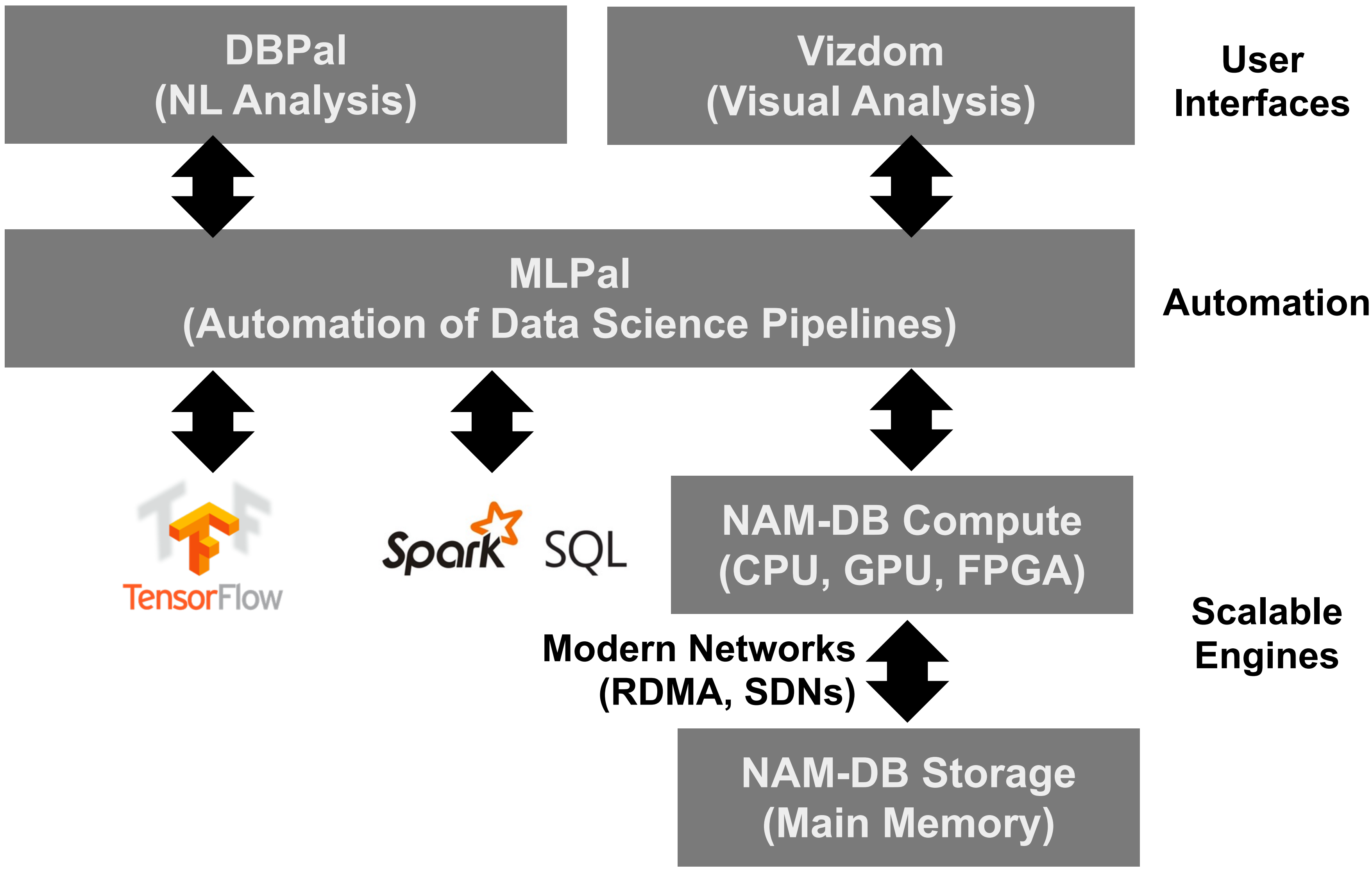




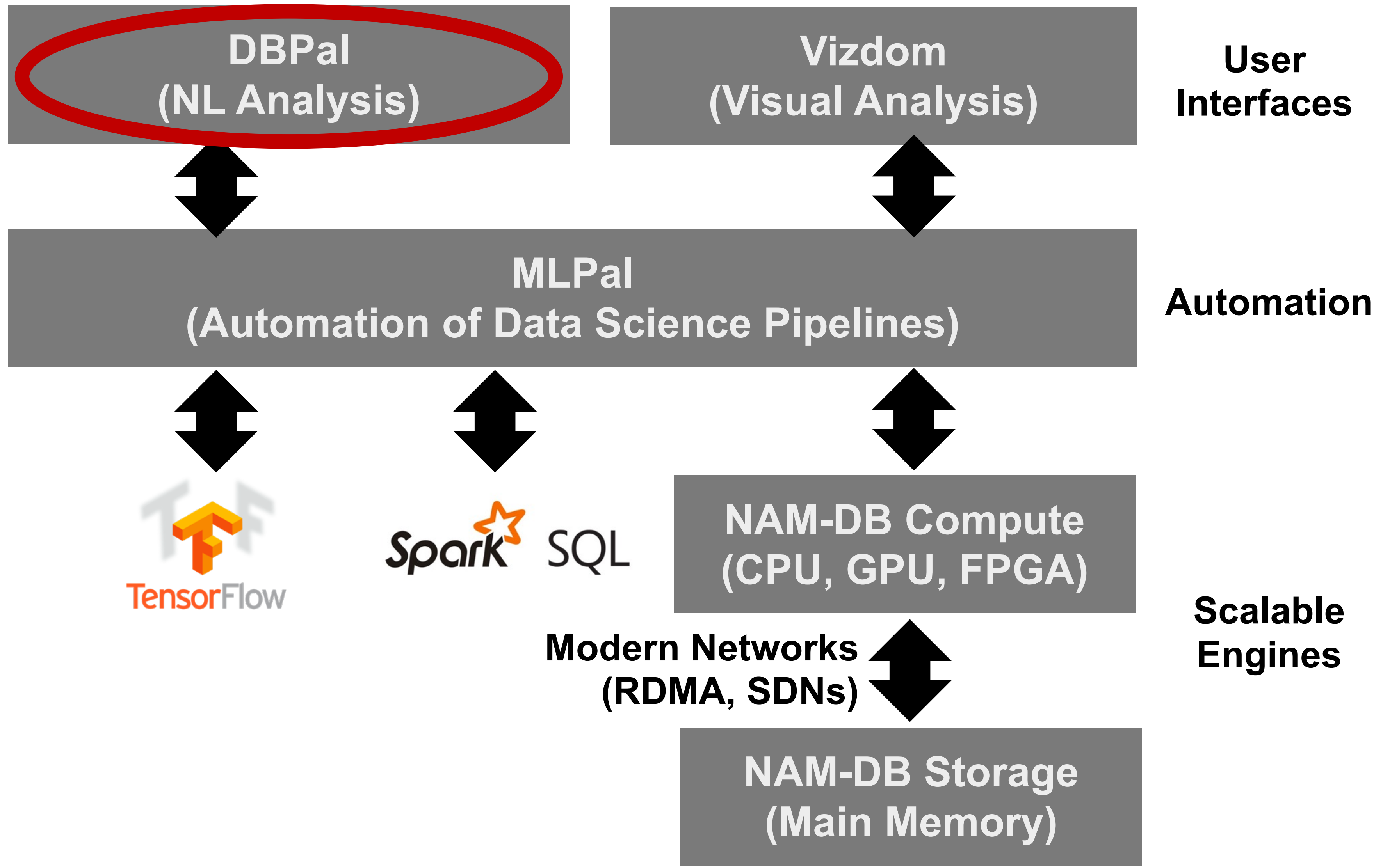
Research Agenda: Revisit Data Science Systems to tackle Pain Points

WHAT ARE THE PAIN POINTS?

OUR DATA SCIENCE STACK

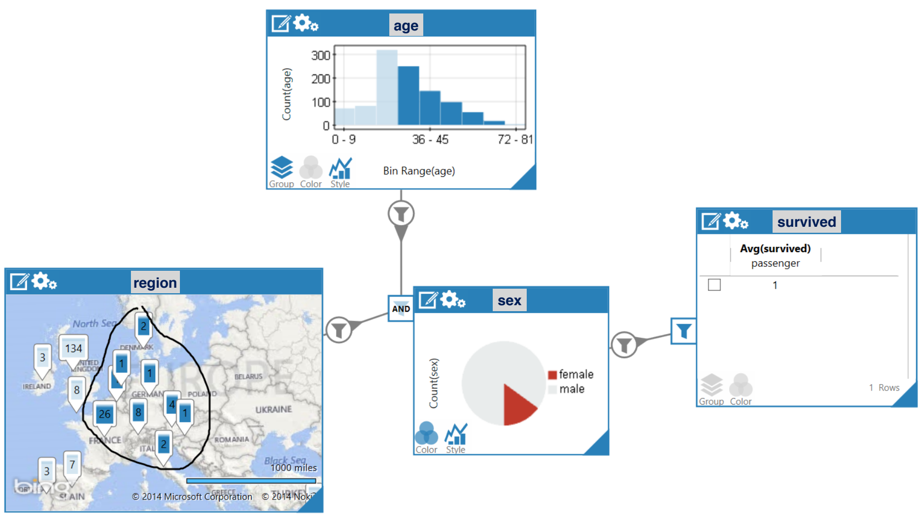


OUR DATA SCIENCE STACK



NL INTERFACE FOR DATABASES (NLIDB)

Visual Interface (e.g., Vizdom):



Natural Language (NL) Interface:

“How many females older than 30 survived the sinking of the Titanic?”

NL interfaces provide a very concise way to query data & can be used hands-free

CHALLENGES FOR NLIDBS

Paraphrased Queries:

- “Show me the patients diagnosed with fever?”
- “What are the patients with a diagnosis fever?”

Incomplete Queries:

- “Fever Patients?” **(fever = diagnosis?)**

Ambiguous Queries:

- How many patients with fever come from New York? **(New York = city or state?)**

NLIDB: DEEP LEARNING TO THE RESCUE

Language Translation Model

Natural Language → ? → SQL

TRAINING DATA IS THE PROBLEM

RECIPE FOR DEEP LEARNING

1. Pick task

2. Manually create training data (e.g., using crowd)

3. Train translation model

(Repeat for every new task)

TRAINING DATA IS THE PROBLEM

RECIPE FOR DEEP LEARNING

1. Pick task

(DATABASE SCHEMA)

2. Manually create training data (e.g., using crowd)

(NL-SQL PAIRS)

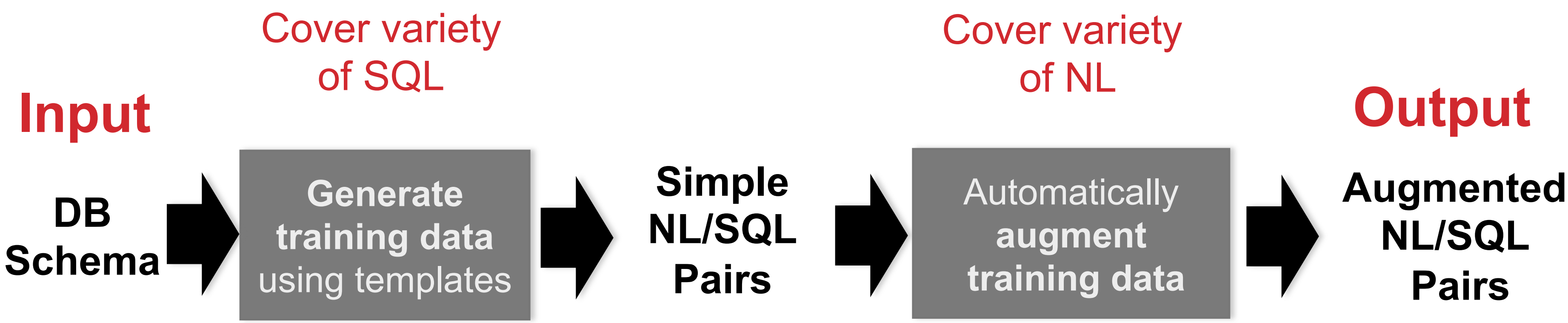
3. Train translation model

(SEQ2SEQ)

(Repeat for every new database OR if database changes)

DBPAL: GENERATING TRAINING DATA

Main Idea: **<u>Synthesize Training Data</u> from Schema**
(based on weak supervision)



Nathaniel Weir et. al.: DBPal: A Fully Pluggable NL2SQL Training Pipeline. SIGMOD Conference 2020: 2347-2361

DBPAL: GENERATING TRAINING DATA

Input → **DB Schema** → Generate training data using templates → **Simple NL/SQL Pairs** → Automatically augment training data → **Augmented NL/SQL Pairs** **Output**

Generate training data using templates: Cover variety of SQL

Automatically augment training data: Cover variety of NL

Template

SELECT <att>
FROM <table>
WHERE <filter>

Show me the <att>s of <table>s with <filter>?

name	age	diagnoses
Carsten	39	fever
Emilie	8	flu
Frederik	4	fever

Patient Database

NL/ SQL Pair

SELECT ***name***
FROM ***patient***
WHERE ***diagnoses = fever***

Show me the names of patients with diagnoses fever?

…

Augmentation

Paraphrasing

Show me the names of patients diagnosed fever?

Noising

Show the names of patients with ~~diagnosed~~ fever?

…

Millions of different NL/SQL pairs

DBPAL: EXPERIMENTAL EVALUATION

Benchmarks:

- Patient (simple schema, 400 queries)
- Geo (complex schema, 280 queries)

Baselines

- Traditional: NaLIR (rule-based)
- Deep Model: NSP and NSP++ (manually created training data)

Patient and Geo Benchmark

	Patients	GeoQuery
NaLIR (w/o feedback)	15.60%	7.14%
NaLIR (w feedback)	21.42%	N/A
NSP++	N/A	**83.9%**
NSP (template only)	10.60%	5.0%
DBPal (w/o augmentation)	74.80%	38.60%
DBPal (full pipeline)	**75.93%**	55.40%

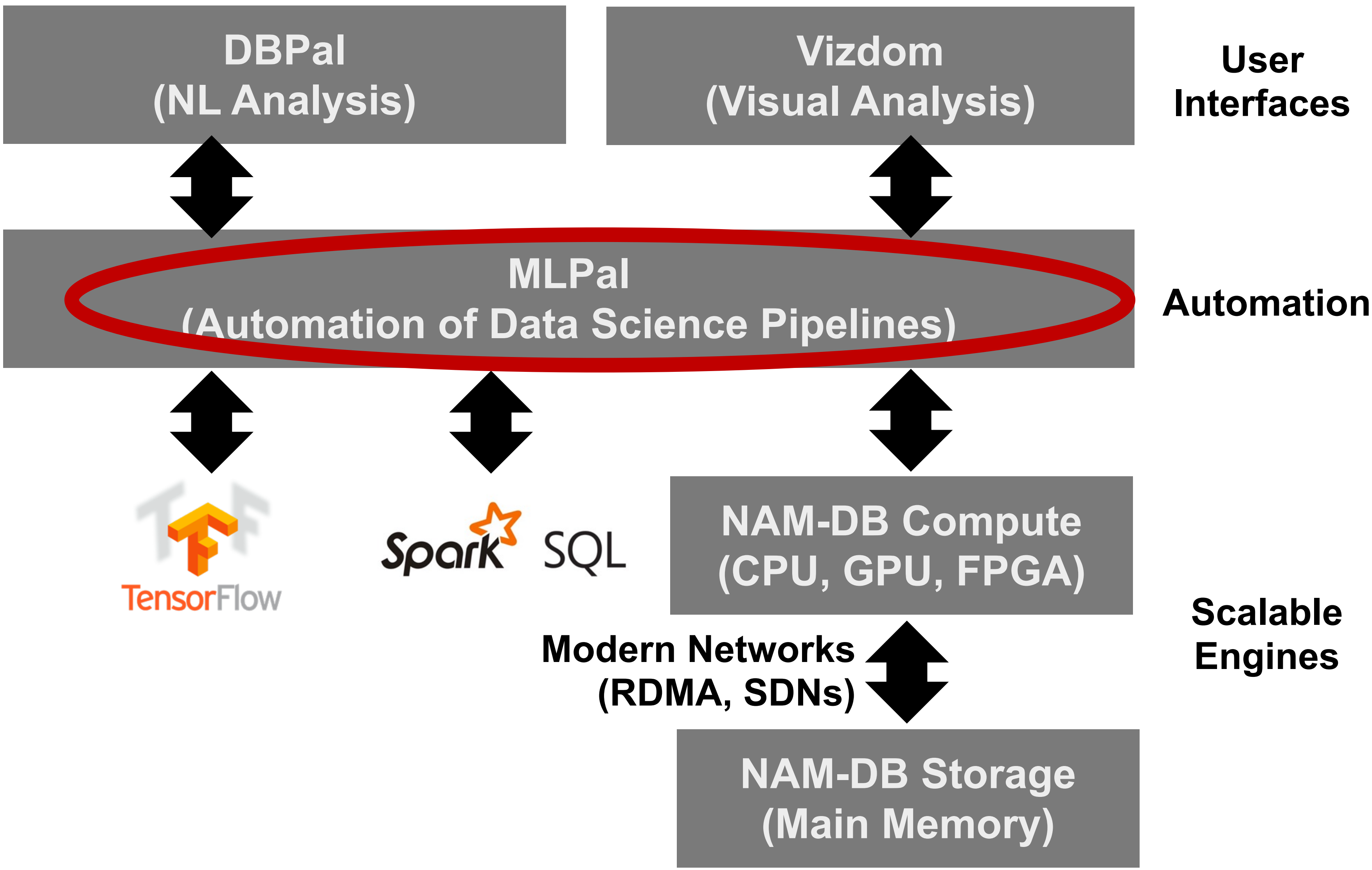
Patient Benchmark (Breakdown per Linguistic Category)

	Naive	Syntactic	Lexical	Morphological	Semantic	Missing	Mixed
NaLIR (w/o feedback)	19.29%	28.07%	14.03%	17.54%	7.01%	5.77%	17.54%
NaLIR (w feedback)	21.05%	38.59%	14.03%	19.29%	7.01%	5.77%	22.80%
NSP (template only)	19.29%	7.01%	5.20%	17.54%	12.96%	3.50%	8.70%
DBPal (full pipeline)	**96.49%**	**94.7%**	**75.43%**	**85.96%**	**57.89%**	**36.84%**	**84.20%**

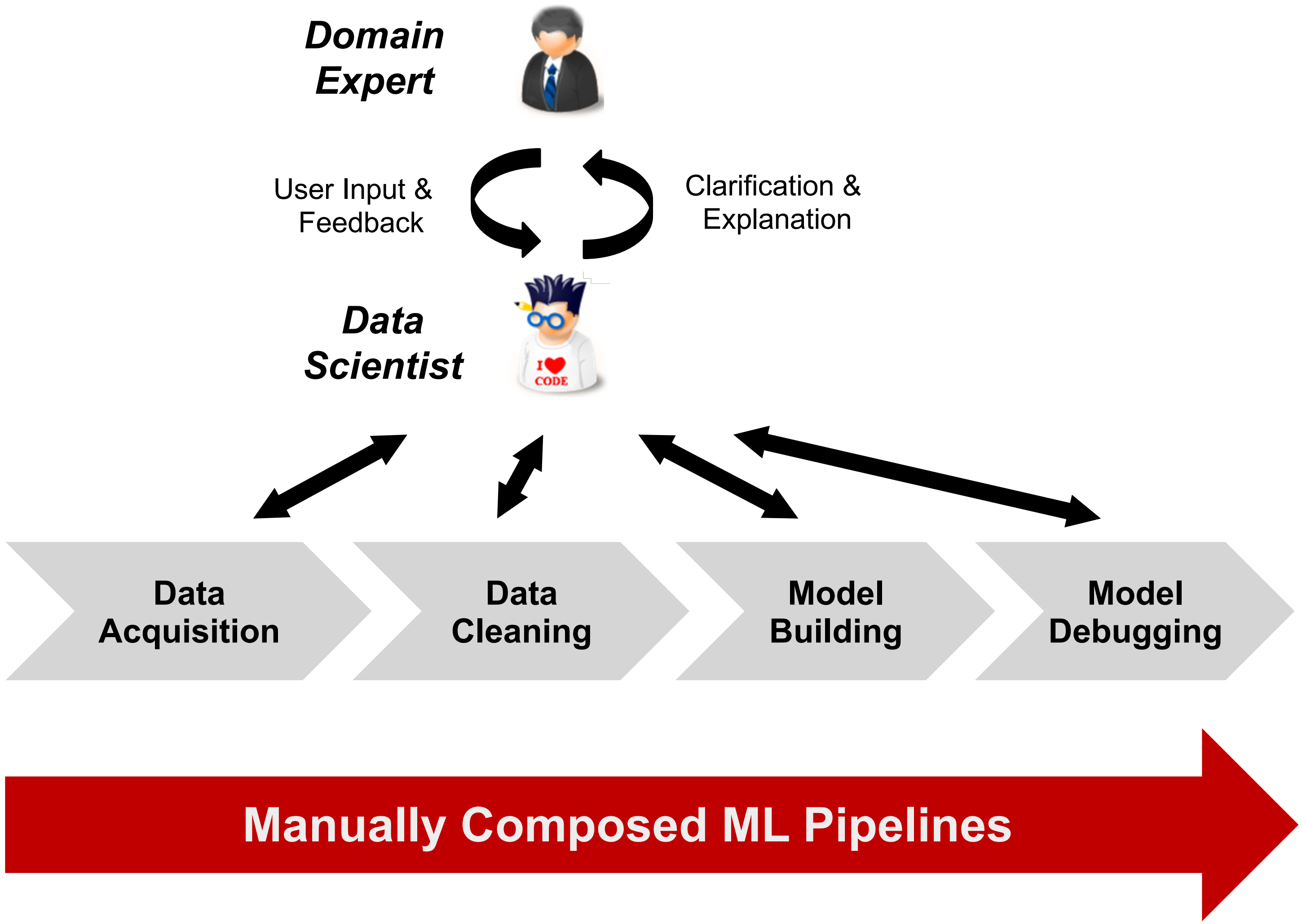
```
nathaniel@titanx:~$ ./interactive.sh
```

```
Loading model...
indexing database...
select distinct first_name from patients
select distinct last_name from patients
select distinct gender from patients
select distinct diagnosis from patients
preparing lemmatizer...
type ":q" to exit
nl query:
```

OUR DATA SCIENCE STACK



HOW ARE ML PIPELINES BUILD TODAY?



WHAT IS THE VISION OF MLPAL?

Domain Expert

User Input & Feedback

Clarification & Explanation

MLPal = Use AI to automate AI

Data Acquisition | **Data Cleaning** | **Model Building** | **Model Debugging**

ReStore

Automation of ML Pipeline Construction

DECISION MAKING ON INCOMPLETE DATA

Motivation: Many data-driven decisions in organizations are based OLAP and Data Warehouses (e.g., total revenue of last year)

amazon REDSHIFT snowflake Google BigQuery …

Central Assumption for OLAP: Data is Complete

- **Traditionally:** Data comes from Internal (curated) Sources
  (data is complete → holds true)
- **Today:** Data Lakes, Integration with External Data, ...
  (data is often incomplete → missing rows of a table)

INCOMPLETE DATA IS EVERYWHERE

Example: Housing Price Dataset in US

- Neighborhoods are complete
- Apartments incomplete: only publicly available in some states

Neighborhood [Complete]		
id	state	pop_density
1	NYC	27,000
2	CA	254
...		

Apartment [Incomplete]	
neighborhood_id	rent
1	2000$
1	3000$
...	...

Complete Apartments for Some States

Sources of Incompleteness

- Systematically Missing Data (e.g., Data only availed in some states)
- Integration of Independent Databases / External Data
- Expensive Data Collection (e.g., Survey Data)

CHALLENGES OF INCOMPLETENESS

Problem: Incompleteness might lead to highly inaccurate results for aggregate queries → erroneous decisions

Challenges:

- Bias in the data (e.g., more apartments from states with dense population and higher rents)
- Correlations across tables (e.g., Higher population density → higher apartment prices)

Strategies today:

- **Ignore Problems** → Assume Sample is representative
- **Manual Cleaning / Completion** → Expensive cleaning

OVERVIEW OF RESTORE (PART OF MLPAL)

Idea:

- Use <u>available data as evidence</u> to synthesize missing data
- Exploits various <u>signals</u> in existing data (e.g., correlations, distributions)

Neighborhood [Complete]		
id	state	pop_density
1	NYC	27,000
2	CA	254
...		

Apartment [Incomplete]	
neighborhood_id	rent
1	2000$
1	3000$
...	...

Complete Apartments for Some States

Main steps:

1. **Offline:** Learn <u>neural completion models</u> from incomplete database
2. **Online:** Generate missing data for aggregate-join queries

```
SELECT AVG(rent) FROM neighborhood
            NATURAL JOIN apartment
            GROUP BY state;
```

Neighborhood ⋈ Apartment [Completed]				
neighborhood_id	state	pop_density	apartment_id	rent
1	NYC	27,000	1	2000$
1	NYC	27,000	2	3000$
2	CA	254	3	3200$
2	CA	254	4	2000$
2	CA	254	5	1000$

Benjamin Hilprecht et al.: ReStore - Neural Data Completion for Relational Databases. SIGMOD 2021

RESTORE: OFFLINE AND ONLINE STEPS

Offline: Schema Annotation by User + Learn Neural Completion Models (both steps are query-independent)

annotated

Landlord [Complete]		
id	age	$TF_{Apartments}$
1	50	1
2	60	?
...		

Apartment [Incomplete]		
neighborhood_id	landlord_id	rent
1	1	2000$
1	2	3000$

Neighborhood [Complete]			
id	state	pop_density	$TF_{Apartments}$
1	NYC	27,000	2
2	CA	254	?
...			

Systematically missing: All apartments available for NYC

generated

Completion Model (Landlord → Apartment)

Input: Evidence Tuple → *Output: Missing Tuple*

Landlord Tuple		
id	age	$TF_{Apartments}$
2	60	3

Apartment Tuple	
landlord_id	rent
2	2000$

Completion Model (Neighborhood → Apartment)

Input: Evidence Tuple → *Output: Missing Tuple*

Neighborhood Tuple			
id	state	pop_density	$TF_{Apartments}$
2	CA	254	3

Apartment Tuple	
neighborhood_id	rent
2	3200$

Online: Use models at runtime to complete missing data for given query

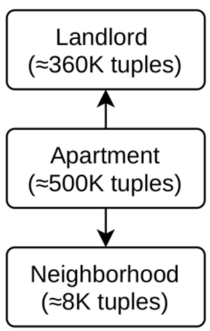
```sql
SELECT AVG(rent)
FROM neighborhood
NATURAL JOIN apartment
GROUP BY state;
```

Neighborhood ⋈ Apartment [Completed]				
neighborhood_id	state	pop_density	apartment_id	rent
1	NYC	27,000	1	2000$
1	NYC	27,000	2	3000$
2	CA	254	3	3200$
2	CA	254	4	2000$
2	CA	254	5	1000$

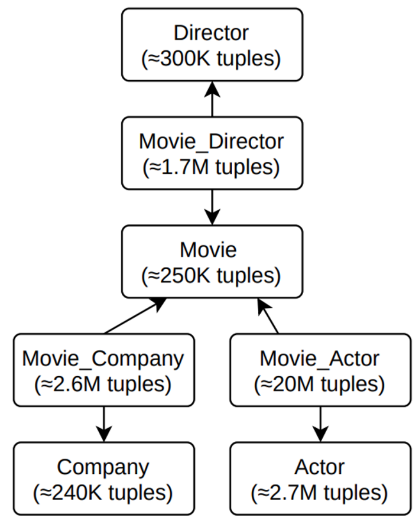
RESTORE: EXPERIMENTAL EVALUATION

Two Real-World Datasets (Airbnb, IMDB/Movies)

- Biased removal of tuples from data sets
- Five different setups per dataset (H1-H5, M1-M5) t + varying keep rate / removal correlation

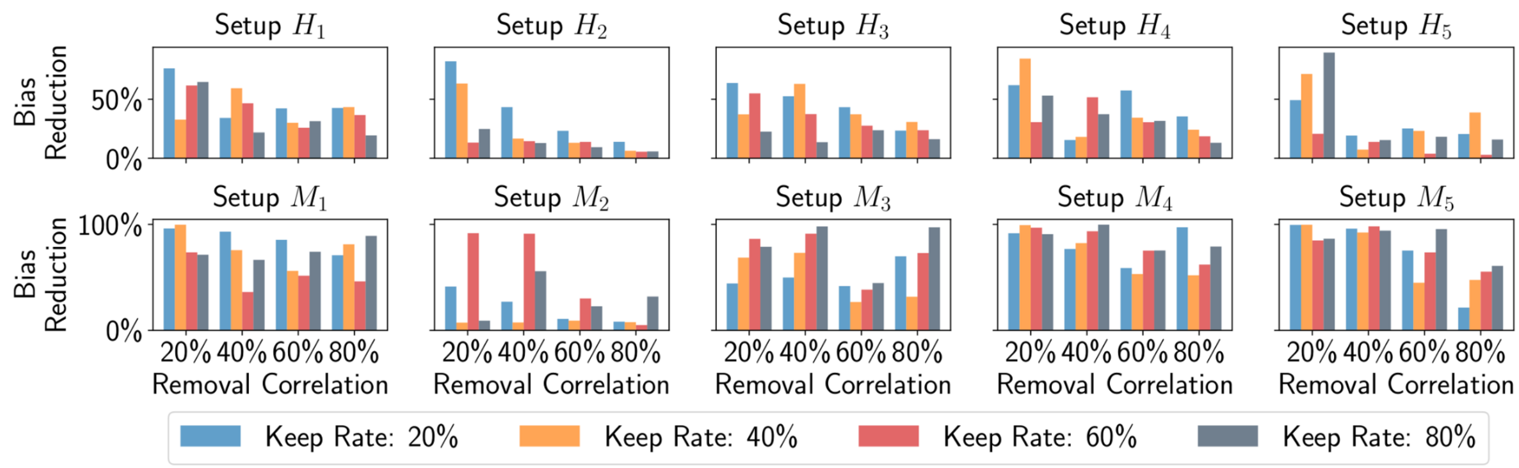


Airbnb Dataset (3 Tables)



IMDB Dataset (7 Tables)

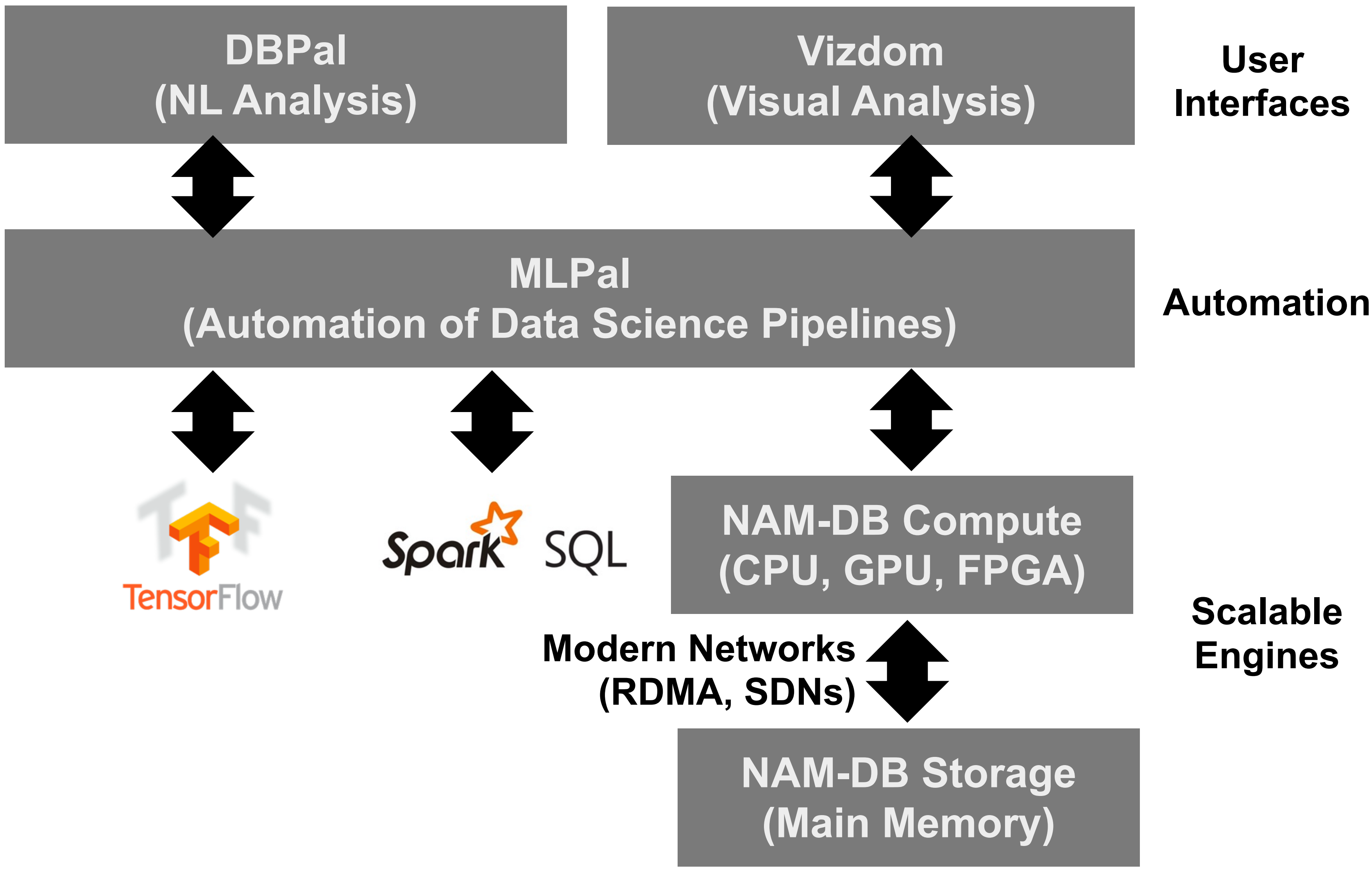
RESTORE: EXPERIMENTAL EVALUATION



Main Findings:

- Bias Reduction up to ~100%
- Varying Accuracy (since predictability varies - in the paper: confidence bounds)
- High removal correlation still good results

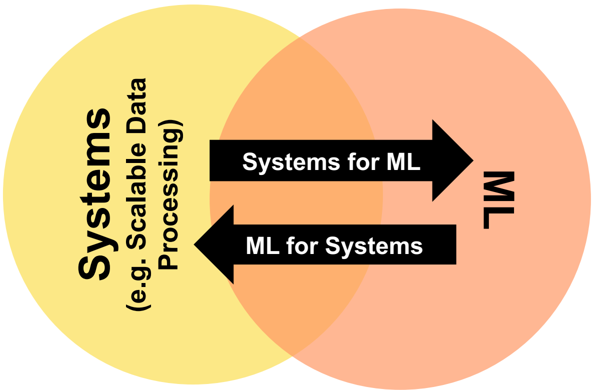
OUR DATA SCIENCE STACK



FUTURE DIRECTIONS

Systems for Machine Learning

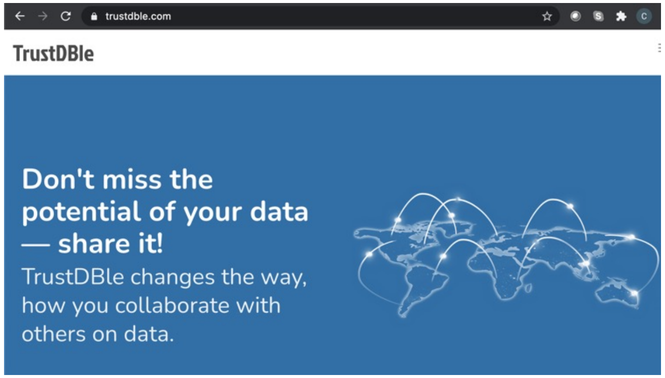
- Automation of Data Science
- Scalable Heterogeneous Systems
- …



Machine Learning for Systems

- Learned Data Partitioning
- Learned Optimizers
- …

Other directions: Trustworthy Data Sharing (TrustDBle)



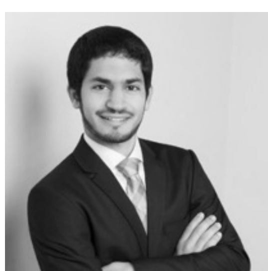
COLLABORATORS AND STUDENTS










THANK YOU FOR YOUR ATTENTION!