Data Cleaning in the Wild: Reusable Curation Idioms from a Multi-Year SQL Workload

Shrainik Jain and Bill Howe
University of Washington
Outline

• Motivation

• SQLShare System
  • Database as a cloud service
  • Multi year SQL workload

• SQL idioms for Data Cleaning

• Automatically identifying the idioms
  • Using word vectors and LSTM models

• Future work
Typical Data Processing Pipeline

Data Cleaning/Wrangling/Munging

Actual Work

80% of the time Via Scripts and external tools

20% of the time In the database
Imagine you’re a scientist….

```
> ./run-experiment-X
Running Experiment X ...
3GB written to Output.csv

> python my_fav_script.py Output.csv
Error: Out of Memory 😞
```

You should use a database!

Friend in the CS dept.
Imagine you’re a scientist....

```
> brew install TheirSQL
... wait some time.
... wait some time.
... wait some time.
Dependency missing. 😞

> brew install dependency
> brew install TheirSQL
```
Imagine you’re a scientist....

> ./TheirSQL.exe
>> Create database XYZ
>> Create TABLE X (, , ,)
>> Insert into X From Output.csv
... wait some time.
Column type mismatch. 😞
>> exit

> vim my_fix_script.py
> python my_fix_script.py Output.csv
Imagine you’re a scientist....

> ./TheirSQL.exe
>> Insert into X From Output.csv
... wait some time.

>> Select * from X

Time to first query: Too Long!
Why not just scripts and files?

Hypothesis: Databases aren’t the problem, it’s how we tell people to use them: No messy data allowed.

But, “clean data is like clean money – it doesn’t exist”

Key Idea: Embrace messy data; clean it up on the go
Outline

• Motivation
• SQLShare System
  • Database as a cloud service
  • Multi year SQL workload
• SQL idioms for Data Cleaning
• Automatically identifying the idioms
  • Using word vectors and LSTM models
• Future work
Solution: SQLShare Database-as-a-Service \cite{1}

- SQLShare Design Principles:
  - Upload should never fail
    - Relaxed schemas
  - Minimal database jargon
    - Unify views and tables
  - Data sharing should be a first-class operation
  - Full SQL support

\cite{1} Shrainik Jain et al., SQLShare: Results from a Multi-Year SQL-as-a-Service Experiment. In proceedings of the 2016 ACM SIGMOD International Conference on Management of Data
Datasets in SQLShare

- Internal dataset structure
- SQL view metadata preview
- Newly uploaded dataset
- Derived views
- Wrapper views
- Base tables
Summary: One system for all of data lifecycle

SQLShare, empowers novice users by providing a system which handles use-cases across the data lifecycle.
A query workload to inform database research

- A dataset of real handwritten queries.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries</td>
<td>24275</td>
</tr>
<tr>
<td>Views</td>
<td>4535</td>
</tr>
<tr>
<td>Tables</td>
<td>3891</td>
</tr>
<tr>
<td>Columns/Table</td>
<td>19</td>
</tr>
<tr>
<td>Users</td>
<td>591</td>
</tr>
</tbody>
</table>
Where does data cleaning come into the picture?

Our goal: SQL recommendation to assist with in-database cleaning.

Current progress: Extract cleaning idioms from the corpus to measure their frequency.
Outline

• Motivation
• SQLShare System
  • Database as a cloud service
  • Multi year SQL workload
• SQL idioms for Data Cleaning
  • Automatically identifying the idioms
    • Using word vectors and LSTM models
• Future work
Idioms from the workload

<table>
<thead>
<tr>
<th>Idiom</th>
<th>Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal recompositioning</td>
<td>210</td>
</tr>
<tr>
<td>Vertical recompositioning</td>
<td>100</td>
</tr>
<tr>
<td>Column Rename</td>
<td>720</td>
</tr>
<tr>
<td>NULL Injection and Type Coercion</td>
<td>420</td>
</tr>
</tbody>
</table>

Total Datasets: 4535
Horizontal recompositioning

Example:

```
SELECT *
FROM [che].[m1]
FULL OUTER JOIN [che].[m3]
  ON [che].[m1].m1_loci_id = [che].[m3].m3_loci_id
```

Curation on Ingest:

- Automatic join finding using measures like jaccard similarity

---

Vertical recompositioning

Example:

```sql
SELECT *
FROM [gbc3].[sqlshare-exp.txt]
UNION ALL
SELECT *
FROM [gbc3].[gen_sqlshare.txt]
```

Curation on Ingest:

- Learn schema alignment heuristics from the data,
- Applying schema matching methods, UNION ALL queries can be automatically identified

Column rename

Example:
SELECT
    column2 AS sp,
    column3 AS SPID,
    column4 AS Prot
FROM
[userX].[uniprotolyblastx2.tab]

Curation on Ingest:
• Non-Trivial.
• Identifying is easy, suggesting valid renames can be ambiguous.
  • One possible way could be to calculate the earth mover distance between the histograms of column values and suggest rename to column with which this distance is least.
NULL injection and Type Coercion

Example:
SELECT
  CASE
    WHEN [400 avg NSAF] = ’N/A’ THEN NULL
    ELSE [2800 avg NSAF]/[400 avg NSAF]
  END
FROM [emma].[NSAFwithAve]

Curation on Ingest:
• Infer data types based on a prefix of rows, and create two table. The first table corresponds to the predicted type, and the second table holds non-conforming rows and has every column typed as a string. Finally, create a view to union the 2 tables and is presented to the user, along with the information about the 2 base tables.
Outline

• Motivation
• SQLShare System
  • Database as a cloud service
  • Multi year SQL workload
• SQL idioms for Data Cleaning
• Automatically identifying the idioms
  • Using word vectors and LSTM models
• Future work
Identifying Query Idioms

• Stack overflow questions can be used to train a neural encoder-decoder model using LSTM networks [1]
  • Embedding SQL queries in n-dimensional vector spaces based on query semantics (description).
  • Use a clustering algorithm to find similar queries.

Identifying Query Idioms

Range Queries

- select species, subspec, name, bodymass from [user450].[birds.csv] where id >= 1 and id <= 20
- select ID, Strain, sex, age, brainwt, bodywt, Res1_sex FROM [user319].[Lincoln University Sample Data-2.csv] where sex = 'F' Or brainwt < 300 and bodywt > 530
- select Time, Mode, Count, Total, S41, S42, S43 FROM [user250].[Old SFR Data] WHERE S41>0 and S41<1000 and Count = 300

Rename Columns

- select gig1 as GigSeq, gig2 as OlySeq, gig3 as PercID, gig4 as alignlength, gig5 as mismatches FROM [user10].[gigastolyblast.tab]
- select [entry no.] as [d1 entry no.], [protein] as [d1 protein], [protein probability] as [d1 protein probability] from [user212].[table_interact-2015_may_6_bacteria_detection17.prot.xls]
- Select [protein description] as [i2.2 protein description], [percent coverage] as [i2.2 percent coverage], [tot indep spectra] as [i2.2 tot indep spectra] from [user212].[table_interact-2015_may_6_bacteria_detection66.prot.xls]
Outline

• Motivation
• SQLShare System
  • Database as a cloud service
  • Multi year SQL workload
• SQL idioms for Data Cleaning
• Automatically identifying the idioms
  • Using word vectors and LSTM models
• Future work
Auto-generating cleaning queries

• What makes a query a "cleaning query"?
• One model: The ones near the root of a deep tree of views.
• Another factor: Cleaning queries are easier to generalize and reuse across users/domains. They involve fewer domain-specific literals, query structures, etc.
• "We're not sure yet"
Auto-generating generic queries

• Using metadata (inferred schema, other tables, past queries) as features, find the right query idiom.
• For a newly uploaded dataset: use metadata to find the class of queries which fit this dataset.
• Synthesize query.
Overall Vision

Ingest Dirty Data
Generate Queries from Idioms
User Does Actual Science
Learn Newer Idioms
Summary

• Relaxed Schemas afford cleaning via SQL.
• Data cleaning can be pushed to Databases, rather than being a prerequisite.
• Automating some cleanup operations within Database seems possible.