Data Curation at Scale: The Data Tamer System

Stonebraker et.al., CIDR 2013

Presented by Zhou Lu, Junyang Gao

Some contents are based on slides of M Stonebraker and Kevin Chang.
Outline

- Introduction
- Example Applications
- System Architecture
- Experimental Validation
- Discussion
Traditional Wisdom - ETL

- Human defines a global schema
- Assign a programmer to each data source to
  - Understand it
  - Write local to global mapping (in a scripting language)
  - Write cleaning routine
  - Run the ETL
- Scales to (maybe) 25 data sources
Introduction

● Data Curation
  ○ Discovering some data source(s) of interest
  ○ Cleaning, Semantically integrating the new data with local data sources
  ○ Deduplicating the result

● Data Tamer, an end-to-end curation system
  ○ Collecting all of the curation components
Four Characteristics

- **Scalability through automation**
  - Automated algorithms, reduce human involvement
- **Data cleaning**
- **Non-programmer orientation**
  - Current ETL system requires programming knowledge
- **Incremental**
  - New data sources must be integrated incrementally as they are uncovered.
Outline

- Introduction
- Example Applications
- System Architecture
- Experimental Validation
- Discussion
Example: A Web Aggregator

- Problem: extract data about “things to do” from the URL into a collection of key-value pair
  - (key-name, value)
- Challenge: the source data might have various data format
Goby Problem

The Same?
Example: A Biology Application

- Problem: integrating the same biology experiment results from various scientists
- Challenge: without a standard for attribute names or languages
Example: A Health Services Application

- Problem: aggregate all of the claims records of health insurance, group by provider and dedup their database
- Challenge: replace manual processes with more automated ones
In all cases:

- Multi source
- Large dataset
- Lack unified standard

- lower cost
- better performance
- automated solution
Outline

- Introduction
- Example Applications
- System Architecture
- Experimental Validation
- Discussion
System Architecture

- ingest
- Schema integration
- crowd
- dedup
- vis/xform

Postgres

- Management Console
  - Schema Integration
  - Entity Consolidation Component
  - Data Visualization Component
  - DE Crowd Sourcing Component
System Architecture - Ingest

- Data Tamer Administrator (DTA) specifies sites - URL or file name
- Assume each site is a collections of records, each contains one or more key-value pair
- Upstream wrapper
- Other information: Schemas, Dictionaries, Templates, Authoritative Tables, etc.
- Loaded into Postgres
System Architecture - Schema Integration

Introduction: Schema Integration

- the activity of integrating the schemas of existing or proposed databases into a global, unified schema.
System Architecture - Schema Integration

Introduction: Schema Integration (cont.)

Key challenge: Semantic Heterogeneity

- same concept, but different names for tables and attributes
  - Rating vs Classification
- Different units
  - Ft vs meter vs inch
- multiple attributes in 1 schema relate to 1 attribute in the other
  - Total price = base price + tax rate * base price
- And more...
System Architecture - Schema Integration

Introduction: Schema Integration (cont.)

Matching system architecture:
System Architecture - Schema Integration

● Level of knowledge
  ○ Level 3: Complete Knowledge on global schema
  ○ Level 1: No knowledge available
  ○ Level 2: Partial information available

● Integrate data sources based on specified schemas
  ○ Use dictionaries, synonyms, templates and authoritative tables for help
  ○ Domain Expert (DE) involved for review
  ○ System gets better and better over time - less human effort
System Architecture - Schema Integration

- Compares an attribute from a data source to a collection of other attributes
- Inner loop is a collection of algorithms (experts):
  - 1. Fuzzy string comparisons over attr. name using trigram cosine similarity
  - 2. Treat each column of data as a document, measure TF-IDF cosine similarity
  - 3. Minimum description length (MDL), similar to Jaccard similarity. Suited for categorial and finite domain data.
  - 4. Welch’s t-test - probability the columns were drawn from the same distribution
- Scores from each experts combined heuristically
System Architecture - Schema Integration
System Architecture - Schema Integration
### System Architecture - Schema Integration

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>2</td>
<td>LOC1.STATE</td>
<td></td>
</tr>
<tr>
<td>phones</td>
<td>2</td>
<td>PHONE</td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>2</td>
<td>EMAIL</td>
<td></td>
</tr>
<tr>
<td>address</td>
<td>2</td>
<td>LOC1 ADDRESS</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>2</td>
<td>TITLE</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>2</td>
<td>DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>images</td>
<td>2</td>
<td>IMAGE1</td>
<td></td>
</tr>
<tr>
<td>moreinfoLink</td>
<td>2</td>
<td>WEBSITE</td>
<td></td>
</tr>
<tr>
<td>city</td>
<td>2</td>
<td>CITY</td>
<td></td>
</tr>
<tr>
<td>siteURL</td>
<td>2</td>
<td>WEBSITE</td>
<td></td>
</tr>
<tr>
<td>zipcode</td>
<td>0.7</td>
<td>ZIP</td>
<td></td>
</tr>
<tr>
<td>lat</td>
<td>0.7</td>
<td>LATITUDE</td>
<td></td>
</tr>
<tr>
<td>amenities</td>
<td>0.6</td>
<td>DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>lon</td>
<td>0.5</td>
<td>LONGITUDE</td>
<td></td>
</tr>
<tr>
<td>activities</td>
<td>0.5</td>
<td>DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td>fax</td>
<td>0.1</td>
<td>DESCRIPTION</td>
<td></td>
</tr>
</tbody>
</table>

**Threshold**
System Architecture - Schema Integration
System Architecture - Schema Integration

Improvements:

- **Now:** Pair-wise comparison. Quadratic in complexity
- **Better:** Two-pass approach
  - Cheap first pass as high-pass filter (No need to look into data, like expert 1)
  - More detailed comparisons in second pass on a subset of attributes
  - Save wasted effort comparing between fields with little in common
System Architecture - Entity Consolidation

- Bootstrapping Training Data
- Records Categorization
- Learning Deduplication Rules
- Clustering and Consolidation
System Architecture - Entity Consolidation

Bootstrapping Training Data:

- Get training set of known duplicates and non-duplicates
- Domain experts involved
- Sort tuples pairs by their similarities, partition the range of sim values into a number of equal-width bins, sample from each bin and let DE label
- Directly get known duplicates from domain knowledge or domain-specific handcrafted rules
System Architecture - Entity Consolidation

Two-Phase Records Categorization:

- Same idea as blocking techniques
- Learn deduplication rules that specific to each category
- First-Phase: clustering sampled tuples using \texttt{k-means++}, obtain a set of representative features that characterize each category
- Second-Phase: assign all other tuples to the closest category
- Categorization is dynamic to new data sets, merge/split when needed
- Using samples, not the entire data set. Avoid expensive operations
What is k-means++?

- Choosing good initial seeds for k-means clustering
- Basic idea: spread out the k initial centers
- After first center is chosen (randomly), after which each subsequent center is chosen from the remaining points with prob. proportional to its squared distance from its closest existing cluster center
- Then run k-means
System Architecture - Entity Consolidation

Learning Deduplication Rules:

- Learn cut-off threshold on attribute similarities
- Learn probability dist. Of attribute similarities for duplicates/non-duplicates
- Naive Bayes Classifier (Maximize A Posteriori)
Clustering and Consolidation:

- Clustering on a list of tuple pairs that are learned to be duplicate
- Ensure the final deduplication results are transitive
- Modified version of correlated clustering algorithm
  - Starts with all singleton cluster
  - Repeatedly merges randomly-selected cluster that have a “connection strength” above a certain threshold
  - Connection strength: the number of edges across two clusters over the total number of possible edges between two clusters (i.e., the Cartesian production of the two clusters)
- Dynamic to the changes of the underlying similarity graph
- Clusters are the consolidated using user-defined rules
Entity Clusters

Different records

Entity identified: **Craft Farm** (3519)

Data source: www_alabarma_travel_3588/0
System Architecture - DE CrowdSourcing

- Deal with multiple DEs’ response
- To determine the confidence of Response Quality
- To increase Response Rate
- To better manage DE workloads
System Architecture - DE CrowdSourcing

Confidence-Based Metrics for DEs and Responses

- DE: Ratings \([0,1]\) => prob. the DE produces a correct response
- Responses: Bayesian Evidence Gathering based on DEs’ rating

\[
P(\bar{R}|B) \cdot P(B) \\ P(\bar{R}|B) \cdot P(B) + P(\bar{R}|\bar{B}) \cdot P(\bar{B})
\]

\[
P(\bar{R}|X) = \prod_{R_i=X} E_i \cdot \prod_{R_i\neq X} (1 - E_i)
\]
System Architecture - DE CrowdSourcing

● Expert Classes
  ○ Classify experts into different class based on their rating
  ○ Assign tasks based on difficulty of tasks, expert classes, budget, etc.

● Economic Model
  ○ Economic incentives for good citizenship. Encourage DE to produce high quality response
  ○ Dynamic pricing to manage workload
System Architecture - Visualization Component

System Architecture - Summary

- Ingest
- Schema integration
- Crowd
- Dedup
- Vis/xform

Postgres
Outline

- Introduction
- Example Applications
- System Architecture
- **Experimental Validation**
- Discussion
Experimental Validation

<table>
<thead>
<tr>
<th></th>
<th>Aggregator</th>
<th>Data Tamer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total records</td>
<td>146690</td>
<td></td>
</tr>
<tr>
<td>Pairs reported as duplicates</td>
<td>7668</td>
<td>180445</td>
</tr>
<tr>
<td>Common reported pairs</td>
<td></td>
<td>5437</td>
</tr>
<tr>
<td>Total number of true duplicates (estimated)</td>
<td>182453</td>
<td></td>
</tr>
<tr>
<td>Reported true duplicates (estimated)</td>
<td>7444</td>
<td>180445</td>
</tr>
<tr>
<td>Precision</td>
<td>97%</td>
<td>100%</td>
</tr>
<tr>
<td>Recall</td>
<td>4%</td>
<td>98.9%</td>
</tr>
</tbody>
</table>

Figure 2: Quality results of entity consolidation for the web aggregator data
Experimental Validation

Figure 3: Quality results of entity consolidation for Verisk data
Outline

● Introduction
● Example Applications
● System Architecture
● Experimental Validation
● Discussion
Discussion

- Transforming raw data into a key-value pair format that is readable for Data Tamer might be very costly. Optimization needed on their “upstream wrapper”
- General data cleaning (other than deduplication) procedure is not included
- Data transformation issues, i.e., ft/meter/inch, price with/without tax
- Accept a live data feed
Thanks!

Questions?