CSE 232A
Graduate Database Systems

Arun Kumar

Topic 5: Data Integration and Cleaning

Slide ACKs: Phil Bernstein, Luna Dong, and Theo Rekatsinas
Outline

❖ Motivation and Definition
❖ Tasks and Challenges
   ❖ Information Extraction
   ❖ Schema Alignment
   ❖ Entity Linkage
   ❖ Data Fusion/Cleaning
Data Integration: Motivation

❖ Unification of large databases across organizations:

Walmart

Target

Each org. might have 1000s of tables with details of products, suppliers, stores, customers, employees, transactions, etc.!

Q: How can merged org. get a uniform view of all data in the org., both schemas and instances? “One version of truth”
Data Integration: Motivation

- Entity search (books, products, etc.)

Such e-retail platforms support millions of third party vendors, keep adding new kinds of products, etc.

**Q:** How can old database schemas be reconciled with new vendor requirements, evolving product catalogs, etc.
Data Integration: Motivation

❖ Unified Web search over text, structured, and other data

Q: How to extract structured data from text and match the “entities” in the infoboxes with search results?
Data Integration: Motivation

- AI services (conversational assistants, chat bots, etc.)

They answer questions by querying multiple data sources.

**Q:** How to enable a uniform view of the backend databases of facts and resolve conflicting entities, facts, etc.?
The Grand Goal of Data Integration

- Provide uniform access to data from multiple autonomous and heterogeneous data sources
  - **Data sources**: Databases/websites/text corpora/etc.
  - **Multiple**: 2 or more data sources (even 2 is hard!)
  - **Heterogeneous**: Source data models may be different
  - **Autonomous**: Data sources not controlled by you
  - **Access**: Ability to query and/or update/maintain data
  - **Uniform**: Same/similar interfaces to reason about data

*Achieve the above with minimal human curation effort!*
Why is Data Integration Hard?

❖ Heterogeneity:
❖ Different ways to capture same entities/attributes/concepts. E.g., “Full Name” vs “First Name; Last Name; MI”, etc.
❖ Different attributes captured in different sources
❖ Different value representations for same entity. E.g., “CA” vs “California” vs “Cal”, etc.
❖ Sources could be in different data models (relational, text, graphs, etc.); may need conversion to one model
Why is Data Integration Hard?

❖ **Ambiguity, Inconsistencies, and Errors:**
❖ Different semantics for same concept. E.g., Does “Salary” mean gross pay or net pay, post tax or pre tax, etc.?
❖ Different concepts with same name. E.g., Does “Apple” refer to a tech company or a fruit?
❖ Manual data entry mistakes, inconsistent naming, etc.

❖ **Scale and Evolution:**
❖ Real-world database schemas/instances large and evolving
❖ Number of data sources can also be large; can change

*Domain-specific human intervention may still be necessary, but automate DI as much as possible.*
More Data Integration Terminology

From the DB community:

❖ **Data Warehouse**: Create a materialized hand-defined single store to pull and unify all relevant data from sources

❖ **Virtual Integration**: Support queries over a “mediated schema” that *reformulates* queries over the sources

From the AI community:

❖ **Knowledge Graph**: Fancier name for a data warehouse! :)

❖ **Linked Data**: Analogous to virtual integration
Outline

❖ Motivation and Definition
❖ Tasks and Challenges
❖ Information Extraction
❖ Schema Alignment
❖ Entity Matching
❖ Data Fusion and Cleaning
Information Extraction (IE)

Extract data with given relation schema (e.g., entity-attribute-value triples) from semi-structured or unstructured data

<table>
<thead>
<tr>
<th>Company</th>
<th>Group</th>
<th>Prev Close (Rs)</th>
<th>Current Price (Rs)</th>
<th>% Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Great Eastern</td>
<td>A</td>
<td>389.60</td>
<td>418.40</td>
<td>+ 7.39</td>
</tr>
<tr>
<td>JSW Energy Ltd.</td>
<td>A</td>
<td>67.25</td>
<td>72.15</td>
<td>+ 7.29</td>
</tr>
<tr>
<td>SREI Infrastructure</td>
<td>A</td>
<td>110.65</td>
<td>117.25</td>
<td>+ 5.96</td>
</tr>
<tr>
<td>Eide Inds.</td>
<td>A</td>
<td>207.35</td>
<td>219.50</td>
<td>+ 5.86</td>
</tr>
<tr>
<td>Indraprastha Gas Ltd</td>
<td>A</td>
<td>1,312.00</td>
<td>1,362.00</td>
<td>+ 3.34</td>
</tr>
<tr>
<td>Kaveri Seed Company</td>
<td>A</td>
<td>501.85</td>
<td>579.30</td>
<td>+ 8.97</td>
</tr>
<tr>
<td>Religare Enterprises</td>
<td>A</td>
<td>48.50</td>
<td>40.40</td>
<td>- 17.65</td>
</tr>
<tr>
<td>Tata Sponge Iron</td>
<td>A</td>
<td>854.15</td>
<td>883.80</td>
<td>+ 3.44</td>
</tr>
<tr>
<td>Dilip Buildcon Ltd.</td>
<td>A</td>
<td>567.65</td>
<td>593.90</td>
<td>+ 4.62</td>
</tr>
<tr>
<td>L&amp;T</td>
<td>A</td>
<td>1,226.50</td>
<td>1,172.30</td>
<td>- 4.34</td>
</tr>
<tr>
<td>Gujarat Pipavav Port</td>
<td>A</td>
<td>129.00</td>
<td>133.95</td>
<td>+ 3.78</td>
</tr>
<tr>
<td>IIFL Holdings</td>
<td>A</td>
<td>622.45</td>
<td>646.00</td>
<td>+ 3.90</td>
</tr>
</tbody>
</table>

Knowledge Base Construction (KBC): Generalization of IE; extract multiple relations (a database) in one go!
Wrapper Induction vs IE

- Extracting structure from HTML or XML is a bit easier; search for relevant paths (e.g., XPaths) to convert to tuple.

<table>
<thead>
<tr>
<th>Name</th>
<th>Year</th>
<th>Director</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Avengers: Infinity War”</td>
<td>2018</td>
<td>“Anthony Russo, Joe Russo”</td>
</tr>
</tbody>
</table>

- Typically done in a “semi-supervised manner” with minimal human annotation of path to trace on a page for a website.
Entity and Relation Extraction from Text

- Extraction structure from free text is far more challenging!
- 3 main types of IE from text:
  - **Closed-world IE**: Entities & attributes known; extract values
    
    | MovieID | Name     | Year | Director       |
    |---------|----------|------|----------------|
    | ID_Avatar | “Avatar” | 2009 | “Jim Cameron” |

  - **Closed IE**: Attributes known; extract entities and values
    
    | MovieID | Name     | Year | Director       |
    |---------|----------|------|----------------|
    | ID_Avatar | “Avatar” | 2009 | “Jim Cameron” |

  - **Open IE**: Extract all of entities, attributes, and values
    
    | Entity         | Attribute | Value  |
    |----------------|-----------|--------|
    | ID_Avatar      | “Name”    | “Avatar” |
Approaches for IE/KBC

- 3 main kinds: Rule-based; statistical NLP; deep learning

- **Rule-based IE:**
  - Developer writes domain-specific rules for matching patterns and extracting entities/attributes/values
  - Gets very tedious; reasonable precision but poor recall

- **Statistical NLP for IE:**
  - Hand-designed NLP “features”; Named Entity Recognition (NER), POS tags, bag of words, syntactic and dictionary-based features, etc. + classical ML model (logistic regression, SVM, etc.)
  - Still a bit tedious for “feature engineering”; slightly better recall but still poor; precision is also poor
SOTA Approach for IE/KBC

❖ **Deep learning for IE:**
❖ Current state of the art (SOTA) methods for IE/KBC use deep learning to automate feature engineering
❖ Word/phrase/sentence embeddings combined with long short-term memory recurrent neural networks (LSTMs), convolutional neural networks (CNNs), and/or recursive neural networks
❖ “Attention” mechanisms helpful for relational extraction
❖ High precision and high recall in many cases
❖ But needs tons of labeled data! :)

16
Outline

❖ Motivation and Definition
❖ Tasks and Challenges
  ❖ Data Extraction
  ❖ Schema Alignment
  ❖ Entity Matching
  ❖ Data Fusion and Cleaning
An old and fundamental problem in DI: Which attributes correspond to which when querying over all sources?

Arises in classical scenario of org. mergers; also arises when consuming KBC outputs with other structured data.

**Source 1**

<table>
<thead>
<tr>
<th>FullName</th>
<th>Age</th>
<th>GrossSalary</th>
<th>NetSalary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Liddell</td>
<td>27</td>
<td>115,000</td>
<td>80,000</td>
</tr>
</tbody>
</table>

**Source 2**

<table>
<thead>
<tr>
<th>LastName</th>
<th>FirstName</th>
<th>MI</th>
<th>Age</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Williams</td>
<td>Aisha</td>
<td>R</td>
<td>30</td>
<td>120,000</td>
</tr>
</tbody>
</table>

**Query:** Get the average salary of all employees
Virtual Databases

- Construct an “intermediary” schema between user-facing queries and data sources
- Acts as a “virtual database” that reformulates queries

Slow, tedious, costly, and error-prone manual process

Reduced human effort to align schemas and add wrappers to sources
Mediated Schema

- Mediated schema must be hand designed up front
- Sources must provide “Source Catalog” with metadata about their local schema and semantics
- Schema design, query optimization, and query execution all faces unique challenges
- **Query reformulation**: Translate queries over mediated schema into queries over source schemas
- 2 main approaches: **Global-as-View** vs **Local-as-View**
Global-As-View (GAV)

❖ **Basic idea**: Mediated schema is treated as a “view” (query) over the set of all source schemas

❖ **Query answering automatically operates over sources**

<table>
<thead>
<tr>
<th>FullName</th>
<th>Age</th>
<th>GrossSalary</th>
<th>NetSalary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice Liddell</td>
<td>27</td>
<td>115,000</td>
<td>80,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LastName</th>
<th>FirstName</th>
<th>MI</th>
<th>Age</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Williams</td>
<td>Aisha</td>
<td>R</td>
<td>30</td>
<td>120,000</td>
</tr>
</tbody>
</table>

Create View Mediated (FullName, Age, Salary)
As Select FullName, Age, GrossSalary From S1 Union Select FirstName||“ ”||MI||“ ”||LastName, Age, Salary From S2

❖ **Issues**: Granularity of information may be lost; not flexible for adding/removing sources;
Local-As-View (LAV)

- **Basic idea**: Each source ("local") schema is treated as a "view" (query) over the mediated schema.
- Need a new query rewriting engine to convert queries over mediated schema to queries over sources.

<table>
<thead>
<tr>
<th>Mediated</th>
<th>LastName</th>
<th>FirstName</th>
<th>MI</th>
<th>Age</th>
<th>GrossSalary</th>
<th>NetSalary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liddell</td>
<td>Alice</td>
<td>NULL</td>
<td>27</td>
<td>115,000</td>
<td>80,000</td>
<td></td>
</tr>
<tr>
<td>Williams</td>
<td>Aisha</td>
<td>R</td>
<td>30</td>
<td>120,000</td>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>

- **Issues**: Query rewriting engine becomes complex; needs new kinds of statistics; new query optimization issues.

Select FirstName||" "||MI||" "||LastName, Age, GrossSalary, NetSalary from Mediated
Q: Can we automate the creation of mediated schema?!

- **Schema Matching**: Algorithmically detect which attributes in the sources are semantically the same/related
  - E.g., S1.\{FullName\} matches S2.\{FirstName, LastName, MI\}

- **Schema Mapping**: Algorithmically construct *transformation functions* between the matches attributes sets! Strictly more general and difficult than schema matching
  - E.g., S1.FullName maps to S2.FirstName||“ ”||S2.MI||“ ”||S2.LastName

- As with IE, 3 main kinds of approaches: rule-based, statistical NLP-based, and deep learning-based (details skipped)
Outline

❖ Motivation and Definition
❖ Tasks and Challenges
❖ Data Extraction
❖ Schema Alignment
❖ Entity Matching
❖ Data Fusion and Cleaning
Entity Matching

- After data extraction and/or schema alignment, tuples have to be integrated into the unified schema
- Alas, **duplications** of entities might exist among sources
- Might need to match and deduplicate entities in unified data; otherwise, query answers will be wrong/low quality
- Aka entity deduplication/record linkage/entity linkage

<table>
<thead>
<tr>
<th>Customers1</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FullName</strong></td>
<td><strong>Age</strong></td>
<td><strong>City</strong></td>
<td><strong>State</strong></td>
</tr>
<tr>
<td>Aisha Williams</td>
<td>27</td>
<td>San Diego</td>
<td>CA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customers2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LastName</strong></td>
<td><strong>FirstName</strong></td>
<td><strong>MI</strong></td>
<td><strong>Age</strong></td>
<td><strong>Zipcode</strong></td>
<td></td>
</tr>
<tr>
<td>Williams</td>
<td>Aisha</td>
<td>R</td>
<td>27</td>
<td>92122</td>
<td></td>
</tr>
</tbody>
</table>

**Q:** Are these the same person (“entity”)?
General Workflow of Entity Matching

❖ 3 main stages: Blocking -> Pairwise check -> Clustering

❖ **Pairwise check:**
  ❖ Given 2 records, how likely is it that they are the same entity? SOTA: “Entity embeddings” + deep learning!

❖ **Blocking:**
  ❖ Pairwise check cost for a whole table is too high: $O(n^2)$
  ❖ Create “blocks”/subsets of records; pairwise only within
  ❖ Domain-specific heuristics for “obvious” non-matches using similarity/distance metrics (e.g., Jaccard on Name)

❖ **Clustering:**
  ❖ Given pairwise scores, consolidate records into entities
Outline

- Motivation and Definition
- Tasks and Challenges
  - Data Extraction
  - Schema Alignment
  - Entity Matching
  - Data Fusion and Cleaning
Data Fusion and Data Cleaning

❖ Instance values from sources might have conflicts!
❖ Can arise due to various reasons: IE errors, schema alignment errors, entity matching errors, or even plain old mismatched semantics!

Q: What is Aisha’s true age?!

<table>
<thead>
<tr>
<th>Customers1</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FullName</strong></td>
<td><strong>Age</strong></td>
<td><strong>City</strong></td>
<td><strong>State</strong></td>
<td><strong>Zipcode</strong></td>
</tr>
<tr>
<td>Aisha Williams</td>
<td>27</td>
<td>San Diego</td>
<td>CA</td>
<td>92122</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customers2</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LastName</strong></td>
<td><strong>FirstName</strong></td>
<td><strong>MI</strong></td>
<td><strong>Age</strong></td>
<td><strong>Zipcode</strong></td>
</tr>
<tr>
<td>Williams</td>
<td>Aisha</td>
<td>R</td>
<td>37</td>
<td>92122</td>
</tr>
</tbody>
</table>

❖ Data Fusion: Resolve conflicts and verify the facts
❖ Data Cleaning: Same issues with quality of data but in a non-DI or single-source setting; often used synonymously
Data Error Detection and Repairs

- If a fact is available from only one source, typically human interventions with domain intuitions are the only way!
- But often, multiple sources report on the same “fact”
- Various techniques to leverage this observation to detect and repair errors in the data
  - Simple majority voting
  - Probabilistic ML models that estimate source accuracy
  - Emerging work on using deep representation learning
- Getting labeled/training data is a key bottleneck; notion of data errors is often too dataset-specific/use case-specific!

*Data cleaning in practice is “death by a thousand cuts”!*
Algorithmic Cleaning Approaches

❖ To reduce human effort, can ask them to codify domain-specific rules/constraints reg. data quality instead of verifying all facts
❖ Can also exploit database dependencies, other integrity constraints, etc. in a unified logical reasoning engine
❖ External knowledge sources (e.g., Freebase) can help
❖ HoloClean is a recent approach that integrates all these possibilities into a unified probabilistic inference framework
  ❖ “Hard constraints” and “soft constraints” for flexibility
  ❖ Labeled data for training weights; repairs as inference
  ❖ Achieves better precision and recall that prior art
Outline

❖ Motivation and Definition
❖ Tasks and Challenges
  ❖ Data Extraction
  ❖ Schema Alignment
  ❖ Entity Matching
  ❖ Data Fusion and Cleaning

References:
“Data Integration and ML: A Natural Synergy” (http://dataintegration.ai)
“Big Data Integration” by Xin Luna Dong and Divesh Srivastava