Generating Unified Famous Objects (UFOs) from the Classified Object Tables

Anusha Kola, Harshal More, Sean Soderman, Michael Gubanov
Department of Computer Science
University of Texas at San Antonio

Abstract—It is difficult to access data generated by different data sources due to the representation and format differences. ETL, KETL, Jedox, Apatar are some examples of data translation and fusion packages that can be used to resolve representation differences of data coming from different data sources have been favored. However, most tools require significant manual effort to map the data sources. Structural mismatch of data between objects with the same semantics reduces the accessibility of data.

Here we discuss our initial efforts toward a scalable unsupervised system and algorithms to generate Unified Famous Objects (UFO) - the self-learning “intelligent” data structures that help automate data fusion at scale [Gubanov et al., 2009], [Gubanov et al., 2011]. UFO is a data structure encapsulating different representations of the same data object (e.g. Songs), hence capable of automatically recognizing and mapping such object in different data sources, and significantly reducing manual effort during data integration process.

We evaluate our algorithms on a large-scale Web tables corpus having $\approx 64$ million of tuples.

Keywords—Web-search; Large-scale Data Management; Cloud Computing; Data Fusion and Cleaning; Summarization; Human-Computer Interaction.

I. INTRODUCTION

Big data variety is one of the most challenging problems in Big data research agenda [Stonebraker, 2012], [Gubanov, 2017], [M.Gubanov et al., 2017], [M.Podkorytov et al., 2017]. A variety of data sources, produce valuable data, but also inevitably introduce information representation differences that represent a significant impediment for someone who wants to gain access to all relevant data sources. Here we describe our initial efforts to automate data fusion on a large-scale corpus of Web tables containing $\approx 64$ million tuples.

Solution: To resolve the information representation mismatch between the data sources, there are many solutions including [Bernstein, 2003], [Gubanov et al., 2009], [Gubanov et al., 2011], [Gubanov et al., 2014], [Gubanov and Stonebraker, 2014], [Gubanov and Payt, 2013], [Gubanov and Shapiro, 2012], [Gubanov et al., 2008], [Gubanov and Bernstein, 2006]. Among these solutions, the IBM UFO Repository is one of the attempts to scale data fusion up. It introduces an initial notion of UFO - an object that creates an abstraction over different data representations that have the same semantics. For example, “preis” is German for “price”, and both should be treated the same way when considering data stored either of the attributes (except the currency is different). Here, we specifically describe our initial efforts to generate UFOs automatically from tables of classified objects in a large-scale structured dataset.

II. ARCHITECTURE

Algorithm: We generate UFOs in three steps. First, we calculate the weights of table attributes within a chosen classified subset of our dataset. Table I shows the attributes sorted by weight for the Songs object. Name, price, album, time, and artist attributes have the most weight.

Next, we group similar attributes together by walking down the set of discovered attributes, generating related words such as synonyms, hyponyms, lemmas and plurals for each attribute. We group attribute $a$ with any attribute $b$ (where $a \neq b$ and is the attribute having a set of words generated) if a word belonging to $a$’s set of words matches $b$.

Lastly, in order to recognize the data without a metadata field, or in recognizing the more related attributes that did not have proper metadata fields, we applied our method to
train a classifier identifying price attribute. This was done by feeding in the data from different price attributes as identified by our generated UFO.

### III. EVALUATION

**Evaluation:** With respect to the above experiment of identifying price attributes, we observe precision of 91.6% and recall of 89.7% on 10-fold cross-validation. In Listing 2, we can see an example of Songs UFO, in Listing 3, we can see an example of Posts UFO.

To evaluate object recognition performance of the generated UFOs, we use these UFOs to classify the Web tables:

- Extract the metadata rows from the Web tables corpus.
- Match the metadata field names using the UFO.
- Return the best matching tables.

Stored UFOs can be used in identifying the objects from any data source. This process is illustrated in Figure 2.

### Table I

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>32781</td>
</tr>
<tr>
<td>price</td>
<td>31123</td>
</tr>
<tr>
<td>time</td>
<td>31020</td>
</tr>
<tr>
<td>album</td>
<td>21374</td>
</tr>
<tr>
<td>länge</td>
<td>1658</td>
</tr>
<tr>
<td>nom</td>
<td>1658</td>
</tr>
<tr>
<td>prix</td>
<td>729</td>
</tr>
<tr>
<td>durée</td>
<td>729</td>
</tr>
<tr>
<td>interpret</td>
<td>591</td>
</tr>
<tr>
<td>格</td>
<td>478</td>
</tr>
<tr>
<td>タイトル</td>
<td>478</td>
</tr>
<tr>
<td>nome</td>
<td>459</td>
</tr>
<tr>
<td>prezzo</td>
<td>415</td>
</tr>
<tr>
<td>durata</td>
<td>415</td>
</tr>
<tr>
<td>naam</td>
<td>375</td>
</tr>
<tr>
<td>アルバム</td>
<td>375</td>
</tr>
<tr>
<td>artiste</td>
<td>285</td>
</tr>
<tr>
<td>precio</td>
<td>250</td>
</tr>
<tr>
<td>álbum</td>
<td>224</td>
</tr>
<tr>
<td>título</td>
<td>216</td>
</tr>
<tr>
<td>artist</td>
<td>208</td>
</tr>
<tr>
<td>アーティスト</td>
<td>158</td>
</tr>
<tr>
<td>description</td>
<td>87</td>
</tr>
<tr>
<td>released</td>
<td>81</td>
</tr>
<tr>
<td>artiest</td>
<td>71</td>
</tr>
<tr>
<td>itunes</td>
<td>47</td>
</tr>
<tr>
<td>duración</td>
<td>44</td>
</tr>
<tr>
<td>preço</td>
<td>44</td>
</tr>
<tr>
<td>nombre</td>
<td>34</td>
</tr>
</tbody>
</table>

| "download": ["download", "search"]

**Listing 2. UFO Songs in JSON**

```json
{"Songs": [
    "name": ["name", "nom", "nome", "タイトル", "naam", "título", "lyrics", "nombre", "song"],
    "price": ["price", "preis", "prix", "prezzo", "preis", "precio", "precio", "perhour"],
    "artist": ["artist", "artista", "artiest", "interpre"],
    "time": ["time", "length", "länge", "lengte", "durée", "durata", "duração", "duração"],
    "album": ["album", "Álbum", "movie"],
    "description": ["description", "descripción"],
    "music": ["music", "type"],
    "date": ["date", "datum"],
    "show": ["show"],
    "type": ["type", "all styles"]
]
```

**Listing 3. UFO Posts in JSON**

```json
{"Posts": [
    "date": ["date", "date & time", "date:", "date —", "issue / date", "date m/d/y", "date d’inscription", "date", "date posted", "date posted", ‘(date’, "date", "date of findings","date and time", "! dates", "thedate", "title/date", "(‘(date’", "‘time’",
    "post": ["latest post", "last post", "posts", "post", "recent post", "last posts", "blog post", "latest post info", "last poster", "poster", "lastpost"],
    "replies/comments": ["replies", "total replies", "no replies yet", "{replies|views}", "no replies", "replies", "comment", "comments", "last comment", "no comment", "special comment", "regulatory comment", "comment", "commented", "# comments", "comments(!)", "issuer comment", "site comments", "comment"],
    "topic": ["topics", "similar topics", "latest topics", "main topics", "general topics", "agenda topics", "photography topics", "forum topics", "active topics", "thesis topics", "topics/messages", "title", "topic title", "title", "item title", "titles", "title", "title"],
    "update": ["last updated", "updated", "date created", "last update", "lastupdated", "modified date", "updates", "updated &lt;&lt", "date modified"],
    "views": ["views", "view", "total views", "{replies|views}", "views", "{view}", "# views"]
]
```
Figure 2. Identify Songs instances using the Songs UFO

IV. RELATED WORK

[Gubanov et al., 2009], [Gubanov and Pyayt, 2012], [Gubanov and Shapiro, 2012] Proposed and evaluated the concept of a Unified Famous Object (UFO) - a unified, standardized interface to heterogeneous data sources. Our constructed UFOs can be used to recognize and query data objects from different data sources.

[Gubanov et al., 2011], [Gubanov and Pyayt, 2013], [Gubanov et al., 2014], [Gubanov and Stonebraker, 2014], [Gubanov et al., 2013] Used UFOs for representing entities in an index of topics extracted from a reference book. Our framework also concentrates on creating the sets of attributes with straight matches then identifying and merging similar attributes into a single UFO attributes.

[Bernstein, 2003] describes Model Management (MM), a framework to manipulate schemas a mappings with many useful common operators. However, MM’s operators work with schemas as a whole, hindering its scalability compared to UFOs.

V. CONCLUSION

Our generated UFOs can be used for:

- Identifying similar data objects in a large-scale structured dataset.
- Fusing/Mapping/Transforming the data instances from different data sources having different schemas.

Our future focus is on training a classifier or algorithm to generate formulas/transformations for the data instances under a particular column for each attribute of an identified UFO. This can be extended further:

1) To form more accurate clusters based on UFOs.
2) To identify similar objects that have similar metadata.
3) To amend objects having none or incorrect metadata with automatically generated metadata.

Finally, the tables can have none or incorrect metadata. Machine learning models can be trained to identify correct metadata if it is present.

REFERENCES


