Text and Structured Data Fusion in Data Tamer at Scale

Michael Gubanov
MIT CSAIL
32 Vassar Street
Cambridge, MA 02139
michaelgubanov@csail.mit.edu

Michael Stonebraker
MIT CSAIL
32 Vassar Street
Cambridge, MA 02139
stonebraker@csail.mit.edu

Daniel Bruckner
UC Berkeley EECS
465 Soda Hall
Berkeley, CA 94720
bruckner@cs.berkeley.edu

***Abstract*—Large-scale text data research has recently started to regain momentum [1]–[10], because of the wealth of up to date information communicated in unstructured format. For example, new information in online media (e.g. Web blogs, Twitter, Facebook, news feeds, etc) becomes instantly available and is refreshed regularly, has very broad coverage and other valuable properties unusual for other data sources and formats. Therefore, many enterprises and individuals are interested in integrating and using unstructured text in addition to their structured data.**

DATA TAMER, introduced in [11] is a new data integration system for structured data sources. Its features include a schema integration facility, an entity consolidation module and a unique expert-sourcing mechanism for obtaining human guidance. Also, included are a capability for data cleaning and transformations.

Here we describe a new scalable architecture and extensions enabling DATA TAMER to integrate text with structured data.

I. INTRODUCTION

Large-scale unstructured data is currently becoming one of the major focal points of data management and information retrieval research [2]–[7], [12]–[15], because of its many attractive properties. For example, online media (e.g. Web blogs, Twitter, Facebook, news feeds, etc) has very broad coverage, is instantly updated and therefore is an attractive large-scale dataset containing a wealth of information not immediately available from other sources. Many web sources export only text, even if they store data internally as something else. Lastly, much enterprise information, such as employee evaluations, internal documents, powerpoint presentations, etc., are primarily text.

DATA TAMER, described in [11] is an end-to-end data curation and integration system for structured data. In that paper, we indicated the main modules in DATA TAMER, including a schema integration facility, a data cleaning module, a primitive transformation engine and an entity consolidation system. We also presented the results from three different pilot use cases. One was the integration of 80,000 URLs from a web aggregator, the second was the integration of 8000 spreadsheets from scientists at a large drug company, and the third was the integration of medical insurance records from an insurance aggregator. All of these use cases entailed integration and consolidation of structured data sources.

However, it is clear that text is an important data source in many environments as noted above. As such it is important to extend DATA TAMER to be able to integrate text with structured and semi-structured data sources. This paper presents the architecture of the extended version of DATA TAMER and a scenario integrating a very large text data source with a collection of structured data sources. We can see in Table I that WEBINSTANCE consists of 242 distributed 2GB extents and has more than 17 million entries (refer to Section 3 for a more detailed explanation of Table I and the dataset).

TABLE I. SEMI-STRUCTURED SHARDED WEB-INSTANCE COLLECTION STATISTICS

```
> db.instance.stats();
{
        "ns" : "dt.instance",
        "count" : 17731744,
        "numExtents" : 242,
        "nindexes" : 1,
        "lastExtentSize" : 1903786752,
        "totalIndexSize" : 733651904,
        ...
}
```

We begin in Section 2 with the extended architecture for DATA TAMER. Then, in Section 3 we continue with a description of the data sets we used. In Section 4 we describe scalable data ingestion, schema integration, and data fusion in DATA TAMER. Also included are performance results of the machine learning text data cleaning and pre-processing extension. Section 5 indicates the demo we plan to run at the conference in more detail. Lastly, Section 6 concludes with related work and suggestions for future research.

TABLE II. SEMI-STRUCTURED SHARDED WEB-INSTANCE AND WEB-ENTITIES COLLECTION STATISTICS

```
> db.entity.stats();
{
        "ns" : "dt.entity",
        "count" : 173451529,
        "numExtents" : 56,
        "nindexes" : 8,
        "lastExtentSize" : 2042834432,
        "totalIndexSize" : 59123168800,
        ...
}
```

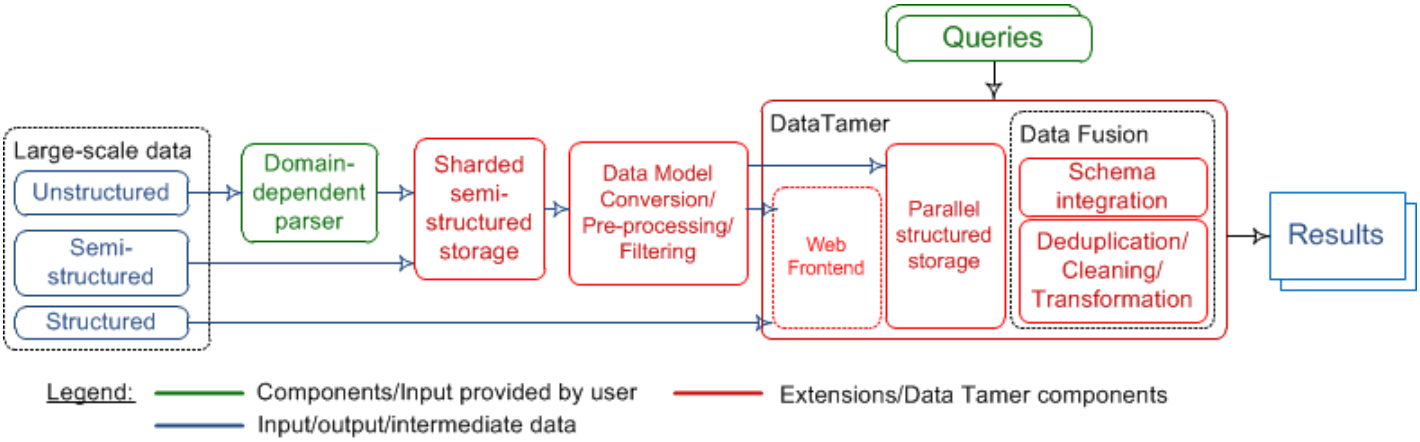

Fig. 1. Extended Data Tamer Large-scale Data Fusion Architecture

II. Architecture

Figure 1 shows our extended architecture for DATA TAMER. In this figure, we show modules for data ingest (accepting data from a data source and storing it in our internal RDBMS), schema integration (matching up attribute names for entities to form the composite entities in the global schema), entity consolidation (finding records from different data sources which describe the same entity and then consolidating these records into a composite entity record), data cleaning (to correct erroneous data) and data transformation (for example to translate euros into dollars).

To extend this system to deal with text we are cooperating with Recorded Future, Inc., a Web aggregator of textual information [16]. They have more than 1 TByte of text they have ingested from the Web. Like other text applications, they are interested in specific kinds of information. Hence, they have a domain-specific parser which looks through the text for information of interest. Other text aggregators we have talked to, indicate the need for a domain-specific parser. This module is shown in Figure 1 as a user-defined module. The result of this parse is large-scale semi-structured/hierarchical data, which after flattening can be processed by DATA TAMER. Hierarchical data model is often used by Web-scale distributed semi-structured storage engines used to manage Web-crawls or other datasets having large amounts of unstructured data. By flattening here we mean the process of converting hierarchical data into flat records before processing by DATA TAMER. However, the characteristics of this data are quite different from structured data sources. For example, the data is usually much dirtier than typical structured data. Also, structured data tends to have many attributes, while text usually has only a few. Since text and structured data have very dissimilar characteristics it is a challenging undertaking to fuse both in DATA TAMER.

III. Datasets

Large-scale data either coming from distributed or local data sources supported by the extended DATA TAMER architecture can be either *structured, semi-structured*, or *unstructured* (see Figure 1).

Large-scale Web-text: Here for demonstration purposes we used ≈ 1 TByte of Web-text primarily from Recorded Future augmented by the fragments of news-feeds, blogs, Twitter, etc processed by a domain-dependent parser [16]. The output of the parse is entity data along with the text fragments where the data came from. Table I illustrates the statistics computed for the WEBINSTANCE dataset containing the fragments. We can see it consists of 242 distributed 2GB extents and has more than 17 million entries. In more detail *ns* in Tables I and II refers to namespace, *count* to the total number of entries in this collection, *numExtents* to the number of 2GB extents used to store the collection, *nindexes* to the number of indexes, and *lastExtentSize* to the size of the last extent on disk in bytes, *lastIndexSize* to the index size created for this collection.

TABLE III. STATISTICS BY ENTITY TYPE IN WEB-ENTITIES

type	cnt
Person	38867351
OrgEntity	33529169
GeoEntity	11964810
URL	11194592
IndustryTerm	9101781
Position	8938934
Company	8846692
Product	8800019
Organization	6301459
Facility	4081458
City	3621317
MedicalCondition	1313487
Technology	940349
Movie	260230
ProvinceOrState	223243

...

Table II has statistics for the WEBENTITIES dataset, which is the output of the domain-specific parser [16], consisting of the entity instances with their attributes. We can see it has more than 173 million entries and consisting of 56 2GB distributed extents. Table III has the statistics by type of entity available for the WEBENTITIES dataset.

Google Fusion tables: In addition to the web-scale text dataset we used 20 structured data sources found using Google Fusion Tables having Broadway shows schedules, theater locations, and discounts. The structured sources on average have

5-20 different attributes and 10-100 rows. We refer to this dataset as FTABLES in the following sections.

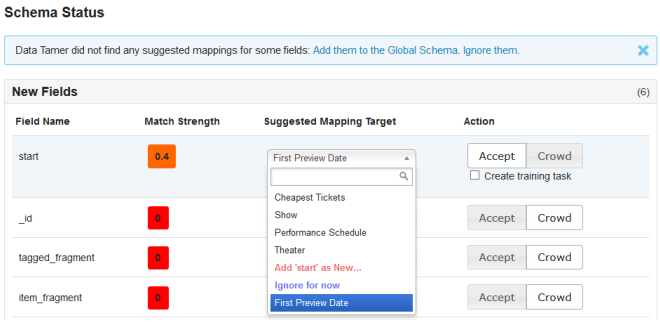

Fig. 2. Schema Integration - Global Schema Initialization

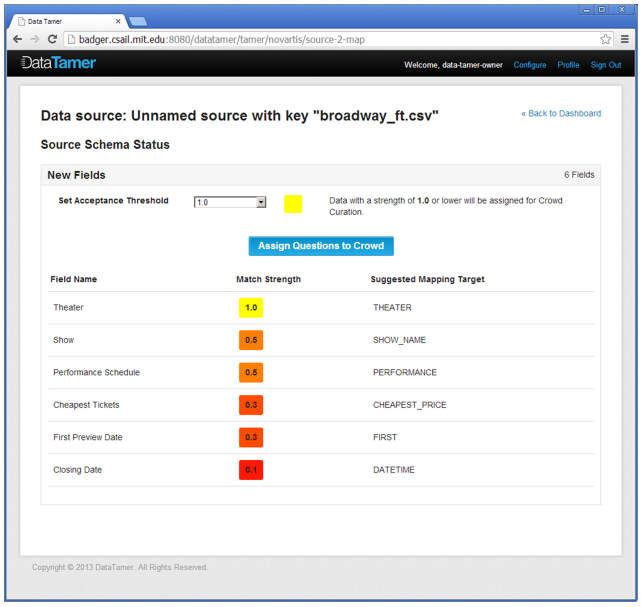

Fig. 3. Schema Integration - Structured Data (Broadway shows)

IV. TEXT AND STRUCTURED DATA FUSION

Here we illustrate the DATA TAMER ability to integrate text and structured data sources. First, the structured data from FTABLES with Broadway show schedules, theater locations, prices, and discounts are imported and used to initialize a global integrated schema. This schema is created from scratch by using metadata from the incoming sources - i.e. in a *bottom-up* fashion. Second, unstructured data containing Web-page text fragments about recent movies and Broadway shows from the WEBINSTANCE is pre-processed, filtered, imported into the DATA TAMER, and used to further populate the global schema. Next, schema matching of these datasource against the global schema is performed to integrate the data. Figure 2 illustrates the early stages of the *bottom-up* process of global schema building, i.e. the stage when the global schema does not have many attributes yet, and the schema matching process may require more human intervention than it will later on. We can see the suggested matching targets from the global schema in the drop-down menu, the matching scores, and the alert indicating that there are some fields that do not have any counterpart in the global schema yet, and the suggested actions (add to the global schema, ignore). Figure 3 illustrates matching the schema of the FTABLES data to the global integrated schema. We can see the attributes on the left from the incoming datasource, the attributes on the right from the global schema and the heuristic matching scores. The user can pick the acceptance threshold by looking at the quality of matches and selecting the threshold value below which the suggested matching targets require expert assessment. Last but not least, we trained a machine-learning classifier on a large-scale web-text and used it for deduplication and data cleaning [6]. It demonstrated 89/90% precision/recall by 10-fold crossvalidation on several different types of entities from the web-text dataset, described in Section 3.

TABLE IV. TOP 10 MOST DISCUSSED AWARD-WINNING MOVIES/SHOWS FROM WEB-TEXT

MOVIE/SHOW
"The Walking Dead"
"Written"
"Mean Streets"
"Goodfellas"
"Matilda"
"The Wolverine"
"Trees Lounge"
"Raging Bull"
"Berkeley in the Sixties"
"Never Should Have"
..

V. DEMO DETAILS

Consider someone, who is interested in watching a recent popular award-winning movie or a Broadway show for the best price possible. The most popular movies and shows are heavily discussed on the Web, in online social media and other information resources, so the user decides to find the top 10 by querying the WEBINSTANCE dataset. The result is illustrated in Table IV. Next, the user picks the *Matilda* show from the list and is interested to find out more about the show, the theaters, schedules, and the best available price.

We demonstrate how DATA TAMER can help find all needed information for this user. We import the first part of the data (the post-processed Web-text fragments discussing the movies) and demonstrate the query results that we can get by using just this dataset. In Table V we can see the information about *Matilda* from the Web text (WEBINSTANCE dataset) - there are no theaters, pricing or schedules. Next, we import the second dataset (FTABLES), perform schema matching, and fusion of these datasets. Table VI illustrates the results of the query on the integrated global schema in DATA TAMER. As a result of the fusion, the user is given an enriched query result and does not need to perform other manual searches for information. Without DATA TAMER, the user would need to spent considerable effort searching and manually putting together pieces of needed information from different data sources. With DATA TAMER, the user can apply the customizable, self-enriching data fusion architecture to help with a *needle-in-a-haystack* search and the consolidation of disparate pieces of needed information.

TABLE V. QUERY RESULTS FOR THE "MATILDA" BROADWAY SHOW FROM WEB-TEXT

SHOW_NAME	"Matilda"
TEXT_FEED	"..which began previews on Tuesday, grossed 659,391, or...And Matilda an award-winning import from London, grossed 960,998, or 93 percent of the maximum."

TABLE VI. ENRICHED QUERY RESULTS FROM WEB-TEXT AND FUSION TABLES

SHOW_NAME	"Matilda"
THEATER	"Shubert 225 W. 44th St between 7th and 8th"
PERFORMANCE	"Tues at 7pm Wed at 8pm Thurs at 7pm Fri-Sat at 8pm Wed, Sat at 2pm Sun at 3pm"
TEXT_FEED	"..which began previews on Tuesday, grossed 659,391, or...And Matilda an award-winning import from London, grossed 960,998, or 93 percent of the maximum."
CHEAPEST_PRICE	"$27"
FIRST	"3/4/2013"

VI. RELATED AND FUTURE WORK

Much of the current research in data management, information retrieval, and search is devoted to the Web, social networks, personal resources, and unfortunately does not apply directly to text. The most recent relevant research by Lu et al proposes an efficient algorithm called *selective-expansion* and *SI-tree* a new indexing structure designed for efficient string similarity joins with synonyms [7]. In [17] Gao et al propose two types of similarities between two probabilistic sets and design an efficient dynamic programming-based algorithm to calculate both types of similarities.

Dong et al in [18] describes generic selective information integration critical in search. Another significant work in [19] sheds light on controversial decision making process in large-scale data fusion. Halevy in [20] describes research efforts in a structural realm of large-scale information fusion. Gupta in [21] gives a partial overview of recent structured data research related to Web search. Many recent venues highlight text as an area of growing interest to Data Management communities [2]–[6], [8]–[10], [15], [17], [18], [22]–[35].

None of these efforts, to the best of our knowledge, systematically address the problem of automatic fusion of text, structured, and semi-structured data at scale. Here we demonstrated a new scalable data integration architecture and a system capable of integrating unstructured, semi-structured, and structured data. We also showed the added value of data integration using the Broadway shows scenario.

REFERENCES

[1] M. Gubanov and L. Shapiro, "Using unified famous objects (ufo) to automate alzheimer's disease diagnostics," in *BIBM*, 2012.

[2] A. Singhal, "Introducing the knowledge graph: Things, not strings," in *Google Blog*, 2012.

[3] M. Gubanov, L. Popa, H. Ho, H. Pirahesh, P. Chang, and L. Chen, "Ibm ufo repository." in *VLDB*, 2009.

[4] L. Chilton, G. Little, D. Edge, D. Weld, and J. Landay, "Cascade: Crowdsourcing taxonomy creation," in *CHI*, 2013.

[5] C. Zhang, R. Hoffmann, and D. Weld, "Ontological smoothing for relation extraction with minimal supervision," in *AAAI*, 2012.

[6] M. Gubanov and M. Stonebraker, "Bootstraping synonym resolution at web scale," in *DIMACS*, 2013.

[7] J. Lu, C. Lin, W. Wang, C. Li, and H. Wang, "String similarity measures and joins with synonyms," in *SIGMOD*, 2013.

[8] Y. Cai, X. L. Dong, A. Halevy, J. M. Liu, and J. Madhavan, "Personal information management with semex," in *SIGMOD*, 2005.

[9] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *SIGMOD*, 2008.

[10] E. Agichtein, E. Brill, and S. Dumais, "Improving web search ranking by incorporating user behavior information," in *SIGIR*, 2006.

[11] M. Stonebraker, D. Bruckner, I. Ilyas, G. Beskales, M. Cherniack, S. Zdonik, A. Pagan, and S. Xu, "Data curation at scale: The data tamer system," in *CIDR*, 2013.

[12] M. Gubanov, A. Pyayt, and L. Shapiro, "Readfast: Browsing large documents through unified famous objects (ufo)," in *IRI*, 2011.

[13] M. Gubanov and A. Pyayt, "Readfast: High-relevance search-engine for big text," in *ACM CIKM*, 2013.

[14] M. Gubanov and M. Stonebraker, "Large-scale semantic profile extraction," in *EDBT*, 2014.

[15] R. Helaoui, D. Riboni, M. Niepert, C. Bettini, and H. Stuckenschmidt, "Towards activity recognition using probabilistic description logics," in *AAAI*, 2012.

[16] "Recorded future inc." [Online]. Available: http://www.recordedfuture.com

[17] M. Gao, C. Jin, W. Wang, X. Lin, and A. Zhou, "Similarity query processing for probabilistic sets," in *ICDE*, 2013.

[18] X. L. Dong, B. Saha, and D. Srivastava, "Less is more: Selecting sources wisely for integration," in *VLDB*, 2013.

[19] ——, "Explaining data fusion decisions," in *WWW*, 2013.

[20] A. Halevy, "Data publishing and sharing using fusion tables," in *CIDR*, 2013.

[21] N. Gupta, A. Halevy, B. Harb, H. Lam, H. Lee, J. Madhavan, F. Wu, and C. Yu, "Recent progress towards an ecosystem of structured data on the web," in *ICDE*, 2013.

[22] M. Gubanov and A. Pyayt, "Medreadfast: Structural information retrieval engine for big clinical text," in *IRI*, 2012.

[23] M. Gubanov, L. Shapiro, and A. Pyayt, "Learning unified famous objects (ufo) to bootstrap information integration," in *IRI*, 2011.

[24] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," in *IJCAI*, 2007.

[25] J. Diederich, W.-T. Balke, and U. Thaden, "Demonstrating the semantic growbag: automatically creating topic facets for faceteddblp," in *JCDL*, 2007.

[26] K. R. Venkatesh Ganti, Surajit Chaudhuri, "A primitive operator for similarity joins in data cleaning," in *ICDE*, 2006.

[27] S. Sekine, "On-demand information extraction," in *COLING/ACL*, 2006.

[28] O. Udrea, L. Getoor, and R. J. Miller, "Leveraging data and structure in ontology integration," in *SIGMOD*, 2007.

[29] S. Amer-Yahia, L. V. S. Lakshmanan, and S. Pandit, "Flexpath: flexible structure and full-text querying for xml," in *SIGMOD*, 2004.

[30] N. Polyzotis, M. Garofalakis, and Y. Ioannidis, "Approximate xml query answers," 2004.

[31] Y. Li, C. Yu, and H. V. Jagadish, "Schema-free xquery," in *VLDB*, 2004.

[32] X. Zhou, J. Gaugaz, W.-T. Balke, and W. Nejdl, "Query relaxation using malleable schemas," in *SIGMOD*, 2007.

[33] X. Dong and A. Y. Halevy, "Malleable schemas: A preliminary report." in *WebDB '05*, 2005.

[34] X. Dong and A. Halevy, "Indexing dataspaces," in *SIGMOD*, 2007.

[35] J. Park and D. Barbosa, "Adaptive record extraction from web pages," in *WWW*, 2007.