Nested Dolls: Towards Unsupervised Clustering of Web Tables

Rituparna Khan, Michael Gubanov
Department of Computer Science
Florida State University

***Abstract*—Here we discuss our initial efforts towards unsupervised clustering of a large-scale Web tables dataset. We improve our previous approach of weakly-supervised clustering, where an operator would provide a few descriptive keywords to generate an entity-identifying classifier, which is applied to the corpora to form a cohesive entity-centric cluster [1]. Here, we make a next step towards *fully unsupervised* algorithm by automatically generating these descriptive keywords. These keywords then can be used to generate high-precision training data and train a classifier to form a cluster. Here, we describe and evaluate this new unsupervised keyword generation algorithm and apply it to a large-scale Web tables corpus to form initial small high-precision clusters.**

***Keywords*-Web-search; Large-scale Data Management;Big Data; Data Fusion; Data Integration; Data Cleaning; Summarization; Human-Computer Interaction; Machine Learning; Natural Language Processing (NLP)**

I. Introduction

There is a variety of structured data on the Web, in enterprise data lakes, hospitals and other data sources. This variety poses a significant impediment on the way to understanding any large-scale structured corpus [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Here we describe our initial efforts to design and evaluate an unsupervised scalable clustering algorithm and apply it to a large-scale Web tables corpus [1]. It forms high-precision entity-centric clusters of Web tables, each corresponding to an entity represented in a corpus by many tables. The clusters are based on attributes, characterizing an entity, for example *artist, title, album* for Songs.

In our previous work on weakly supervised clustering applied to keyword-search over Web tables, we observe that keyword-search over the entire corpus is inferior to the same keyword-search (and ranking function), but over a smaller entity-centric cluster relevant to the query [16]. For example, if a user is looking to buy a song from iTunes, routing the query to the Songs cluster will return more relevant search results.

There are other approaches to structured data classification and clustering. For example, [17] proposed a solution of stitching tables on the same site into meaningful union table by considering contexts which are web page title and text surrounding the table. They considered each context as a sequence labeling problem and identified useful attribute values. By contrast, our approach is fully unsupervised. We, first, generate high precision table clusters based on calculated weights of the attributes. Calculating weights was challenging, because of the spam present in the Web tables corpus and a variety of ways to name even the same attribute. We performed spam filtering and NLP parsing to clean the corpus and normalize the attributes to their root form. Table I depicts the list of attributes sorted by their weights (normalized frequency). Our algorithm generates these lists and forms the attribute sets that are used to generate clusters in a "nested doll" fashion (see Section 3).

Attribute (Root Form)	Weight
date	10051
name	5532
time	3730
categori	3649
titl	3262
comment	3227
descript	3225
type	3221
post	3084
price	3040
size	2916
last	2806
team	2659
total	2508
valu	2237
rank	2133
score	1953

Table I
Attribute weight list

We describe the architecture and our "nested doll" clustering algorithm in Section 2. Section 3 has the detailed performance evaluation. Section 4 reviews the related work. We conclude in Section 5 with related work and forward looking statements.

II. Architecture

We use a parallel columnar storage engine [18] and created a wide table with 20 columns to ingest and store our corpus from [1]. We set an attribute called metaFlag to *true* for the rows containing metadata. After fetching the attribute names, we cleaned the attributes of Spam and stemmed using NLP to get the root forms of those attributes. Based on the cleaned root forms, we calculated attribute weights.

"Nested Dolls" clustering: We form clusters based on the attribute lists sorted by weight in a "nested doll" fashion,

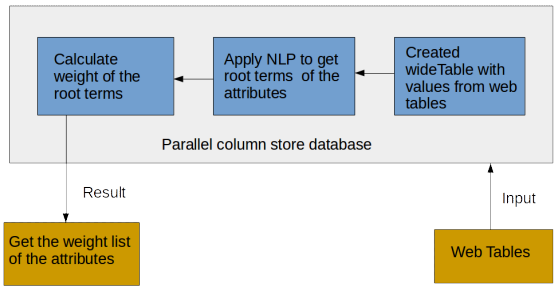

Figure 1. Architecture

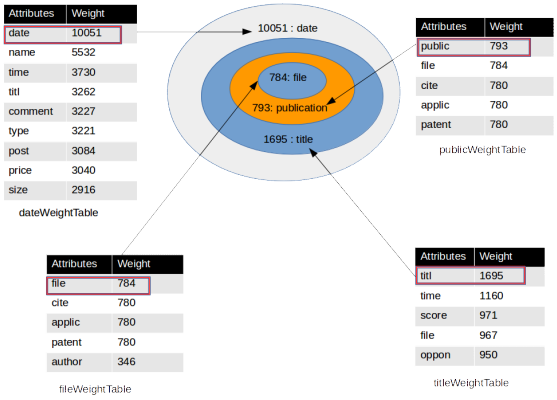

Figure 2. "Nested Dolls" Clustering

hence the algorithm name. Table I shows the attributes sorted by weight for a *random* sample.

We refer to Figure 2 to explain the attribute set and further the cluster formation. First, we take the top attribute from the attribute list calculated on the random sample from the entire corpus (i.e. *date* in this case) and calculate a derived attribute list just on those tables that have that attribute. We depict schematically those table sets having the attributes as nested ovals in Figure 2. We continue calculating the derived attribute lists, taking the top attribute, filtering the table set by this attribute in similar fashion to step 1. This results in smaller table sets based on *title, publication, file* attributes nested within the outer *date* table set.

The query below generates a "nested doll" based on an attribute from the previous round. The sub-query output are the rowIDs of tables having a specific attribute (*topSubAttribute*). The outer query selects the attribute list of such tables, which forms the next level "nested doll".

Cluster	Attributes	Precision	Recall
Songs	Name,Time,Price	100%	4.16%
Sports	Date,Score,Opponent	99%	33%
Patents	Date,Title,Publication,File	97%	33%
Swimming	Name,Time,LMSC	100%	6.67%
Comments	Categories,Comments	96%	15.38%

Table II
REPRESENTATIVE HIGH-PRECISION CLUSTERS

Listing 1. Nested Doll Generation.

```
SELECT table_attributes,rowId,col1,col2
FROM wideTable WHERE
rowId IN (SELECT rowId
FROM attributes WHERE
table_attribute LIKE topSubAttribute);
```

We observed that slight deviation of attribute selection in radius of 10 within a top attribute sometimes results in a better quality cluster. This phenomena is a subject of our current research.

III. EVALUATION

We demonstrate several top entity clusters generated by our algorithm in Table 2. To calculate precision and recall, we draw a *random* samples of size 100 and manually label the entities by using independent annotators. We also make the following observations:

- Normalized weighted sorted root-form attribute lists are quite effective in approximately summarizing the content of a large-scale structured corpus. For example, in Table I we can find top attributes *team*, *score*, *total*, *rank*, which means Sports is one of the main themes of the corpus.
- The smaller gets the "nested doll", the more specific are the tables. For example, *date* "doll" has a variety of tables. *date* and *title* "doll" has tables with various documents and patents. A smaller "doll" with *date*, *title*, *publication* and *file* attributes has tables only with patents.

IV. RELATED WORK

[17] proposed a solution of "stitching" tables from the same Web site into a "union" table. For stitching the tables, they extracted additional information from HTML surrounding the table such as page title and text surrounding the table. [19] discusses challenges of identifying titles of web tables. The authors trained a sequence-to-sequence model to generate titles from flattened key-value pairs having a web table metadata.

[20] discussed knowledge base exploration and observed that existing knowledge exploration relies on combination of knowledge bases and query logs. They proposed *Knowledge Carousels* (sideways and downwards) for simplifying traversal of large corpus of Web tables. Downwards approach

provides information on entities while sideway approach results in useful associations. [21] discussed challenges of extracting high-quality Web table corpus from HTML and getting semantic clues from the extraction. They worked to extract high quality Web tables by training two high-accuracy classification models using multi-kernel SVM to get both *horizontal* and *vertical* tables. They report precision and recall of 96.6% and 85.1% respective, which indicates a high-precision model.

All of the above approaches are supervised either requiring expensive training data to train the models or another form of supervision. We aim here exactly at the opposite - avoid any supervision, hence making first steps to generate high-precision training data using fully unsupervised "nested-doll" algorithm.

V. Conclusion

We described a fully unsupervised scalable "nested doll" algorithm to generate high-precision entity clusters in a large-scale Web table corpus. It is a step forward compared to our previous work on weakly supervised Web table clustering that required a few descriptive keywords from the user to generate an entity cluster. Here we completely eliminated this requirement. Clustering of Web tables can be valuable tool for:

- *Data fusion* as the formed clusters contain all relevant tables pertaining certain entity from the corpus.
- *Search*, as keyword-search over a more focused cluster provides more relevant search results compared to searching over the entire corpus. The query can be routed to a most relevant cluster and return better search results. More detailed analysis and evaluation of relevance gain can be found here [16].

Our current research focus includes expanding the clusters (i.e. increasing recall), while maintaining scalability and unsupervised properties of the algorithm.

Acknowledgments: We would like to thank anonymous reviewers for their feedback on earlier drafts of this paper. This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 1701081.

References

[1] M. Gubanov, M. Priya, and M. Podkorytov, "Cognitivedb: An intelligent navigator for large-scale dark structured data," in *WWW*, 2017.

[2] M. N. Gubanov, P. A. Bernstein, and A. Moshchuk, "Model management engine for data integration with reverse-engineering support," in *ICDE*, 2008.

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

[4] M. Gubanov, L. Shapiro, and A. Pyayt, "Readfast: Structural information retrieval from biomedical big text by natural language processing," in *Information Reuse and Integration in Academia and Industry*. Springer, 2013.

[5] B. Alexe, M. Gubanov, M. Hernandez, H. Ho, J.-W. Huang, Y. Katsis, L. Popa, B. Saha, and I. Stanoi, "Simplifying information integration: Object-based flow-of-mappings framework for integration," in *BIRTE*, 2009.

[6] M. N. Gubanov, L. Popa, H. Ho, H. Pirahesh, J.-Y. Chang, and S.-C. Chen, "Ibm ufo repository: Object-oriented data integration," *VLDB*, 2009.

[7] M. Gubanov and A. Pyayt, "Type-aware web search," in *EDBT*, 2014.

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

[9] ——, "Text and structured data fusion in data tamer at scale," in *ICDE*, 2014.

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

[11] A. P. Michael Gubanov and L. Shapiro, "Readfast: Browsing large documents through ufo," in *IRI*, 2011.

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

[13] S. Ortiz, C. Enbatan, M. Podkorytov, D. Soderman, and M. Gubanov, "Hybrid.json: High-velocity parallel in-memory polystore JSON ingest," in *IEEE BigData*, 2017.

[14] M. Simmons, D. Armstrong, D. Soderman, and M. N. Gubanov, "Hybrid.media: High velocity video ingestion in an in-memory scalable analytical polystore," in *IEEE BigData*, 2017.

[15] M. N. Gubanov, "Polyfuse: A large-scale hybrid data fusion system," in *ICDE*, 2017.

[16] S. Soderman, A. Kola, M. Podkorytov, M. Geyer, and M. Gubanov, "Hybrid.ai: A learning search engine for large-scale structured data," in *WWW*, 2018.

[17] F. W. Xiao Ling, Alon Halevy and C. Yu, "Synthesizing union tables from the web," in *IJCAI*, 2013.

[18] M. Stonebraker, D. J. Abadi, A. Batkin, X. Chen, M. Cherniack, M. Ferreira, E. Lau, A. Lin, S. Madden, E. J. O'Neil, P. E. O'Neil, A. Rasin, N. Tran, and S. B. Zdonik, "C-store: A column-oriented DBMS," in *VLDB*, 2005.

[19] H. L. Braden Hancock and C. Yu, "Title generation for web tables," in *ArXiv*, 2018.

[20] F. K. Y. W. C. Y. Fernando Chirigati, Jialu Liu and H. Zhang, "Knowledge exploration using tables on the web," in *VLDB*, 2016.

[21] B. H. H. L. J. M. A. R. W. S. K. W. F. W. Sreeram Balakrishnan, Alon Halevy and C. Yu, "Appling webtables in practice," in *CIDR*, 2015.