Findings from Two Decades of Research on Schema Discovery using a Systematic Literature Review

Silvio Normey[1,3], Lorena Etcheverry[1], Adriana Marotta[1], and Mariano P. Consens[2]

[1] Universidad de la República, Uruguay
[2] University of Toronto
[3] Instituto Federal de Educação Ciência e Tecnologia Sul-Rio-Grandense

Abstract. We present a systematic literature review applied to the last twenty years of research in the area of schema discovery (also known as schema inference, or schema extraction) applied to semistructured data. Our survey characterizes the different objectives, methodologies, and evaluations that are described in the literature. We present the preliminary findings of our analysis and make observations that can benefit future research and development efforts in the area.

1 Introduction

Semistructured data formats have enjoyed increased adoption for more than two decades. These flexible formats have several practical applications, including data exchange, Web APIs, and data storage. While the XML standard quickly become the dominant format two decades ago, recent years have seen JSON emerge as a popular API and storage format. Key characteristics of semistructured data are the co-existence of data and meta-data, and the flexibility to support schema-less data, as well as schema validation in a variety of languages. This meta-data flexibility has motivated multiple research efforts aimed at discovering (or extracting, or inferring) schema from semistructured data instances. Schema discovery has also been incorporated in a variety of semistructured data management tools, and new scenarios and applications continue to emerge.

We believe that a systematic literature review of this area can contribute to future research efforts, while also helping to inform data management tool developers. Our survey characterizes the different objectives, methodologies, and evaluations present in the literature of the last twenty years. This short paper briefly describes the survey methodology in the next Section. The final Section discusses the preliminary results of our systematic literature review.

2 Survey Methodology

Our approach follows the systematic survey methodology described in [1, 2]. This Section describes the first two phases of the process; planning the review, and conducting the review. The next Section reports the results.

CEUR-WS.org/Vol-2100/paper25.pdf

2.1 Planning the Review

The first phase, review planning, consists of the following three activities.

Identifying the need for the review. As far as we know, there is no comprehensive literature survey that synthesizes the knowledge developed over the last two decades to address schema discovery in semi-structured data. We believe that a systematic literature review shall shed light over a variety of issues relevant to future schema discovery research and development efforts.

Formulating the research questions. Formulating one or more research questions (abbreviated RQ) is a critical step in the systematic literature review methodology we follow. Our study starts by focusing in the following research question.

RQ: *What are the objectives, methodologies, and evaluations that are present in the schema discovery literature, applied to semistructured data formats (excluding schema discovery from web pages)?*

Developing the review protocol. The review protocol defines the methods used during the execution of the systematic review (described in the next Section).

2.2 Conducting the review

The second phase, conducting the review, is composed of two steps (search strategy and study selection), described below.

Search strategy The search strategy objective is to find publications strongly related to the RQ, while completing and capturing potentially reproducible bibliographic searches. The procedure consists of the following three steps.

Identify the search terms Search terms are formulated from the RQ, and synonyms are incorporated (using the boolean OR connector). In our study, the search expression corresponds to "schema discovery OR schema extraction OR schema inference".

Identify the literature resources The authors judgment selected five electronic bibliographic databases; ACM Digital Library, IEEE Xplore, SpringerLink, Science Direct, Scopus. The authors consider that ACM (Digital Library), IEEE (Xplore), Springer (Link), and Elsevier (ScienceDirect) are the main publishers (and corresponding bibliographic portals) of highly ranked journals and conferences in the computer science area. The authors also consider that Scopus, an abstract and citation database that indexes a broad set of sources, can contribute by expanding the search space.

Conduct the search process The search process consists in submitting the search expressions in each one of the five selected libraries, and storing all the results obtained. This requires adapting the search expression (and choosing appropriate advanced search options) for each portal interface.

Study selection The set of references obtained from the searches conducted in all the libraries is filtered in various steps; duplicates are removed, the title and the abstract of each paper is judged in order to discard out-of-topic papers, and then inclusion and exclusion criteria is applied to obtain a refined set of papers. The initial search returned 412 pertinent papers, of which 107 papers were identified as duplicates, and therefore excluded, resulting in a set of 305 papers. Then, out-of-topic papers were discarded after reading their title and abstract. Finally, inclusion and exclusion criteria were applied to further filter the set of papers. The inclusion criteria consisted in only keeping computer science papers related to the research question, which have been published between 1997 and 2017. Exclusion criteria consisted in filtering papers that are not writen in english, or focused on HTML based sources or Deep Web. We excluded works that deal with schema discovery from structured web pages since they have been already reviewed in extent in the context of web mining tasks [3]. The outcome of this selection process was 76 selected papers, and 229 excluded.

3 Review results and discussion

In this section we first define the criteria used to analyze the selected papers. Then, we present the results of a preliminary analysis, which consists in applying these criteria to a subset of 31 of the selected papers. Table 1 summarizes the results of this analysis. Finally, we discuss on some interesting aspects observed.

The analysis criteria is organized in three aspects: the objectives of the paper, the methodology outlined in the paper, and the evaluation strategy. We further refine these aspects as follows:

- **Objectives**. We identify the problems and contexts addressed by the work. We define four categories: concrete motivation and applications (OM), semistructured data formats supported (OF), schema languages supported for the input (OSI) and the output (OSO). For example, observing the row corresponding to [4] in Table 1 we see that the motivation for extracting the schema is to obtain a schema description in order to query data (OM), while the addressed data format is JSON (OF), and JSON appears as the output format used in the proposal (OSO).
- **Methodology**. This criterion focuses on the main characteristics of the proposed solutions. The defined categories are: internal data representation (MD), inferring attributes, related-entities, constraints, types (MI), software environment and availability of an implementation (MS). Continuing with the previous example, in Table 1, row [4], we find that the proposed solution uses a graph as internal representation (MD), it infers attributes and data types (MI), and the paper presents information about the implementation (MS).
- **Evaluation**. This analysis aspect aims to answer how experiments were carried out and how their results were studied and validated. For this purpose the following categories were defined: quality measures for the result schema (EQ), experimental input data (ED), experimental measures (EM), comparison with alternative solutions (EC), support for updates, appends, streaming (EU), support for schema evolution (EE), and scalability of the solution and parallelization (ES). Returning to our example in Table 1, in the row corresponding to [4] we observe that the authors do not present quality measures

for the obtained schema (EQ), that they use real data in the experiments (ED), that they measure the execution time of their process (EM), and that they present a comparison with other solutions (EC). However, they do not show experimentation about updates, appends, streaming or evolution in schemas (EU and EE) and neither they carry out experiments on scalability or parallelization (ES).

3.1 Discussion

Most of the selected works do not present a motivation for schema extraction, they are only focused on the methodology. In some cases the motivation is the need of an schema to improve data querying, to implement query verification, or to manipulate data. Few works emphasize on the need for schema extraction to check constraints.

Regarding data formats, most of the works use either XML, JSON, or RDF. We observe that oldest data formats, such as OEM and XML, were object of investigation in the 90s and the beginning of the past decade. In the current decade JSON and RDF are the main objects of study. Most of the reviewed solutions receive raw data as input (e.g., XML or JSON documents), while the output format varies. In the case of XML data, the extracted schemas are often presented as DTDs and XML schemas. In the cases of RDF and JSON, the extracted schema often consists of a class structure.

Most of the reviewed works on JSON and XML use trees to internally represent the inferred schema, and also as output. In the case of RDF data tuples, classes, and graphs are used, and there is not a clear preference.

Regarding on what the reviewed works produce, we observe that all the proposals infer the structure of the schema, while 39% of them also infer data types and 26% also infer related-entities.

In regard to the experimentation, we observe that most of the papers measure the quality of the extracted schema. These evaluation is often carried out on real data, while few works use synthetic data. Two metrics are frequently used to evaluate the solutions: the effectiveness of the schema to evaluate the accuracy of the proposed methodology, and the execution time to test its efficiency. Most of the reviewed works (62%) do not compare their approach with others, and in most of the cases scalability tests are omitted. A small portion of the literature reviewed addresses evaluation. A similar comment applies to the availability of tools and implementations.

Another significant point of analysis is the shortage of solutions that support schema evolution, updates, appends or stream. This means that in most of the algorithms proposed it is necessary to re-process all the database and infer a new schema in order to keep it updated.

References

1. Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University **33**(2004) (2004) 1–26

2. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. Journal of systems and software **80**(4) (2007) 571–583
3. Kosala, R., Blockeel, H.: Web mining research: A survey. SIGKDD Explor. Newsl. **2**(1) (June 2000) 1–15
4. Störl, U., Darmstadt, H., Scherzinger Othregensburg, S.: Schema Extraction and Structural Outlier Detection for JSON-based NoSQL Data Stores. (2015)
5. Cánovas Izquierdo, J.L., Cabot, J.: JSONDiscoverer: Visualizing the schema lurking behind JSON documents. Knowledge-Based Systems (2016)
6. Baazizi, M.A., Colazzo, D., Ghelli, G., Sartiani, C.: Counting types for massive JSON datasets. In: Proceedings of The 16th International Symposium on Database Programming Languages - DBPL '17. (2017)
7. Gallinucci, E., Golfarelli, M., Rizzi, S.: Schema Profiling of Document Stores. (2017)
8. Ruiz, D.S., Morales, S.F., Molina, J.G.: Inferring versioned schemas from NoSQL databases and its applications. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). (2015)
9. Wang, L., Zhang, S., Shi, J., Jiao, L., Hassanzadeh, O., Zou, J., Wangz, C.: Schema management for document stores. Proc. VLDB Endow. **8**(9) (May 2015) 922–933
10. Cánovas Izquierdo, J.L., Cabot, J.: Discovering implicit schemas in JSON data. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). (2013)
11. Baazizi, M.A., Lahmar, H.B., Ben, H., Colazzo, D., Ghelli, G., Sartiani, C.: Schema Inference for Massive JSON Datasets
12. DiScala, M., Abadi, D.J.: Automatic Generation of Normalized Relational Schemas from Nested Key-Value Data. In: Proceedings of the 2016 International Conference on Management of Data - SIGMOD '16. (2016)
13. Christodoulou, K., Paton, N.W., Fernandes, A.A.A.: Structure inference for linked data sources using clustering. In: Transactions on Large-Scale Data- and Knowledge-Centered Systems XIX: Special Issue on Big Data and Open Data. (2015) 1–25
14. Kellou-Menouer, K., Kedad, Z.: On-line Versioned Schema Inference for Large Semantic Web Data Sources. (2017)
15. Abedjan, Z., Gruetze, T., Jentzsch, A., Naumann, F.: Profiling and mining RDF data with ProLOD++. In: Proceedings - International Conference on Data Engineering. (2014)
16. Weise, M., Lohmann, S., Haag, F.: LD-VOWL: Extracting and visualizing schema information for linked data. In: CEUR Workshop Proceedings. (2016)
17. Konrath, M., Gottron, T., Staab, S., Scherp, A.: SchemEX - Efficient construction of a data catalogue by stream-based indexing of linked data. In: Journal of Web Semantics. (2012)
18. Matono, A., Kojima, I.: Paragraph tables: A storage scheme based on RDF document structure. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). (2012)
19. Kellou-Menouer, K., Kedad, Z.: Schema Discovery in RDF Data Sources. In: ER. (2015)
20. Mlýnková, I., Nečaský, M.: Towards Inference of More Realistic XSDs. (2009)
21. Marciniak, J.: XML schema and data summarization. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). (2010)

22. Mlýnková, I., Nečaský, M.: Heuristic Methods for Inference of XML Schemas: Lessons Learned and Open Issues. **24**(4) (2013) 577–602
23. Guen-Hae, K., Sang-Ki, K., Yo-Sub, H.: Inferring a Relax NG Schema from XML Documents. (2016)
24. Xing, G., Parthepan, V.: Efficient schema extraction from a large collection of XML documents. (2011)
25. Klempa, M., Kozak, M., Mikula, M., Smetana, R., Starka, J., Švirec, M., Vitásek, M., Necaský, M., Holubova, I.: JInfer: A framework for XML schema inference. Computer Journal (2013)
26. Janga, P., Davis, K.C.: Mapping Heterogeneous XML Document Collections to Relational Databases. LNCS **8824** (2014) 86–99
27. Peng, F., Chen, H.: Discovering restricted regular expressions with interleaving. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). (2015)
28. Klempa, M., Stárka, J., Mlnková, I.: Optimization and Refinement of XML Schema Inference Approaches. Procedia Computer Science **10** (2012) 120–127
29. Cao, H., Qi, Y., Selçuk, K., #3, C., Sapino, M.L.: XML Data Integration: Schema Extraction and Mapping. (2010)
30. Janga, P., Davis, K.C.: Schema extraction and integration of heterogeneous XML document collections. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). (2013)
31. Garofalakis, M., Gionis, A., Rastogi, R., Seshadri, S., Shim, K., Kaist, A.: XTRACT: A System for Extracting Document Type Descriptors from XML Documents. (2000)
32. Bex, G.J., Gelade, W., Neven, F., Vansummeren, S.: Learning Deterministic Regular Expressions for the Inference of Schemas from XML Data. ACM Transactions on the Web (2010)
33. Hegewald, J., Naumann, F., Weis, M.: XStruct: Efficient schema extraction from multiple and large XML documents. In: ICDEW 2006 - Proceedings of the 22nd International Conference on Data Engineering Workshops. (2006)
34. Nestorov, S., Ullman, J., Wiener, J., Chawathe, S.: Representative Objects: Concise Representations of Semistructured, Hierarchical Data. (1997)

A Appendix

Table 1. Sample application of the analysis criteria

	Criteria													
	Objectives				Methodology			Evaluation						
References	OM	OF	OSI	OSO	MD	MI	MS	ED	EQ	EM	EC	EU	EE	ES
[5]	Schema Description	JSON	JSON	Class	Class	Attributes, Related-entities	✓	-	-	-	-	-	-	-
[6]	Schema Description	JSON	JSON	Schema, Type	Tree	Attributes, Type	-	Real	✓	Succinctness, Execution Time	-	-	-	✓
[7]	Schema Description, Query	JSON	JSON	Tree	Tree	Attributes	-	Real, Synthetic	✓	Effectiveness, Execution Time	-	-	-	✓
[8]	Schema Description, Query	JSON	JSON	Class	Class	Attributes, Type, Related-entities	-	-	-	-	-	-	-	-
[9]	Schema Description, Query	JSON	-	-	Tree	Attributes	-	Real	✓	Effectiveness	-	-	✓	-
[4]	Schema Description, Query	JSON	-	JSON	Graph	Attributes, Type	✓	Real	-	Execution Time	✓	-	-	-
[10]	Schema Description, Query	JSON	-	Class	Tree	Attributes, Type, Related-entities	✓	-	-	-	-	-	-	-
[11]	Schema Description, Query Verification, Check Constraints	JSON	-	Schema, Type	Tree	Attributes, Type	✓	Real	✓	Execution time	-	✓	✓	✓
[12]	Schema Description, Query, Data Manipulation	JSON	JSON	Schema	Graph	Attributes, Related-entities	-	Real	✓	Effectiveness	✓	-	-	-
[13]	Schema Description, Query	RDF	RDF	Class	Tuple	Attributes, Related-entities	-	Real, Synthetic	✓	-	-	-	-	-
[14]	Schema Description	RDF	RDF	Class	Class	Attributes	✓	Real	✓	Execution Time, Effectiveness	✓	-	-	✓
[15]	Schema Description	RDF	RDF	Class	Tree	Attributes, Constraints	-	Real	-	-	-	-	-	-
[16]	Schema Description	RDF	RDF	Class	Class	Attributes, Type	✓	Real	-	-	-	-	-	✓
[17]	Schema Description, Query	RDF	RDF	RDF	Graph	Attributes	-	Real	-	Effectiveness, Execution Time	-	✓	-	✓
[18]	Schema Description, Query	RDF	RDF	Table	Table, Graph	Attributes	-	Real, Synthetic	✓	Execution Time	✓	-	-	✓
[19]	Schema Description, Query Verification, Check Constraints	RDF	RDF	Class	Tuple	Attributes, Related-entities	-	Real	✓	-	-	-	-	-
[20]	Schema Description	XML	XML	XSD	Automaton	Attributes, Type	-	-	-	-	-	-	-	-
[21]	Schema Description, Data Manipulation	XML	XML	XML	Graph	Attributes, Related-entities, Data	✓	Real	-	-	-	-	-	-
[22]	Schema Description	XML	XML	DTD, XML Schema	Tree	Attributes, Type, Constraints	-	-	-	-	-	-	-	-
[23]	Schema Description	XML	XML	Tree	Tree	Attributes	-	Synthetic	✓	Effectiveness	-	-	-	-
[24]	Schema Description	XML	XML	DTD	Automaton	Attributes	-	Synthetic	✓	Effectiveness, Succinctness, Execution Time	✓	-	-	✓
[25]	Schema Description	XML	XML	DTD, XSD, XML Schema	Tree, Grammar	Attributes, Type	✓	Synthetic	✓	Succinctness	-	-	-	-
[26]	Schema Description	XML	XML	Relational Model	Grammar	Attributes, Related-entities, Constraints	-	Real	✓	Effectiveness, Execution Time	✓	-	-	✓
[27]	Schema Description	XML	XML	Regular Expression	Regular Expression	Attributes	-	Real	-	-	✓	-	-	-
[28]	Schema Description	XML	XML	DTD, XSD, XML Schema	Automaton, Tree	Attributes	-	Synthetic	✓	Succinctness	✓	-	-	-
[29]	Schema Description, Data Manipulation	XML	XML	DTD, XML Schema	Tree, Graph	Attributes, Type	-	-	-	-	-	-	-	-
[30]	Schema Description, Data Manipulation	XML	XML	DTD, XSD	Tree, Grammar	Attributes, Type, Constraints	-	Real, Synthetic	✓	Execution Time	✓	-	-	-
[31]	Schema Description, Query, Query Verification	XML	XML	DTD	Regular Expression	Attributes	-	Real, Synthetic	✓	Effectiveness	✓	-	-	-
[32]	Schema Description	XML	Automaton	Expression	Regular Expression	Attributes	✓	Real, Syntethic	✓	Effectiveness, Execution time	✓	-	-	-
[33]	Schema Description, Query, Query Verification, Data Manipulation	XML	XML	XSD	Structure Model	Attributes, Type	-	Synthetic	✓	Schema Description, Execution Time	-	-	✓	✓
[34]	Schema Description, Query	OEM	OEM	Representative Objects	Graph	Attributes	-	-	-	-	✓	-	-	-