ABSTAT 1.0: Compute, Manage and Share Semantic Profiles of RDF Knowledge Graphs

Renzo Arturo Alva Principe, Blerina Spahiu(✉), Matteo Palmonari, Anisa Rula, Flavio De Paoli, and Andrea Maurino

University of Milano-Bicocca, Milano, Italy
{renzo.alvaprincipe,blerina.spahiu,matteo.palmonari,anisa.rula, flavio.depaoli,andrea.maurino}@unimib.it

Abstract. As Linked Data available on the Web continue to grow, understanding their structure and content remains a challenging task making such the bottleneck for their reuse. ABSTAT is an online profiling tool which helps data consumers in better understanding the data by extracting ontology-driven patterns and statistics about the data. This demo paper presents the capabilities of the new added feature of ABSTAT.

1 Introduction

Knowledge Graphs (KGs) in the Linked Open Data cloud[1] define possible classes and relations in a schema or ontology, and mainly describe instances and interlink entities through relations. KGs cover different domains and are widespread, for example, in the EuBusinessGraph project[2], several parties contribute their data into the KG of the company. Despite the gross amount of data available on the Web, the selection of the data suitable for a given task is not straightforward as many data discovery steps have to be performed in order to understand data set's content and their characteristics. Thus, in order to use a data set, one needs to know which classes and properties are most commonly used, which predicates are generally associated with an instance of a given class, the potential domain and range of a given predicate, the cardinality of a predicate, etc. ABSTAT is an ontology-driven linked data summarization model which helps users in an effortless understanding of the data [5]. Given a RDF data set and, optionally, an ontology (used in the data set), ABSTAT computes a semantic profile which consists of a summary and statistics. ABSTAT's summary is a collection of patterns known as Abstract Knowledge Patterns (AKPs) of the form <`subjectType, pred, objectType`>, which represent the occurrence of triples <`sub, pred, obj`> in the data, such that `subjectType` is a minimal type of the subject and `objectType` is a minimal type of the object. With the

[1] http://lod-cloud.net/.
[2] http://eubusinessgraph.eu/.

A. Gangemi et al. (Eds.): ESWC 2018 Satellite Events, LNCS 11155, pp. 170–175, 2018.
https://doi.org/10.1007/978-3-319-98192-5_32

term type we refer to either an ontology class (e.g., foaf:Person) or a datatype (e.g., xsd:DateTime). By considering only minimal types of resources, computed with the help of the data ontology, we exclude several redundant AKPs from the summary making them compact and complete. Summaries are published and made accessible via web interfaces, in such a way that the information that they contain can be consumed by users and machines (via APIs). The user interface is available and can be used to explore summarized datasets[3]. Several approaches to profile RDF data have been proposed, we refer to our research papers [1,5] for a detailed discussion of state-of-the-art. While many of these approaches publish and make accessible the computed profiles, only a few are open source and, to the best of our knowledge, none of them provide support for the summarization process to the user. Based on requirements collected in the two industry-driven innovation projects EW-Shopp[4] and EuBusinessGraph we have built ABSTAT 1.0, a tool to compute, manage and make accessible to humans and machines semantic profiles of RDF graphs. Compared to the ABSTAT research prototype [2], ABSTAT 1.0 not only provides more features, which are used in different applications scenarios [1,3,5] but it has also developed into a tool that lays on a more scalable modular and effective architecture, and is endowed with a user interface to help the management of the profiling process. ABSTAT 1.0 is released as open source[5] under the GNU Affero General Public License v3.0[6].

In this paper, we make the following contributions: (i) Minimalization over properties; (ii) AKPs inference and instance count; (iii) Cardinality extraction; (iv) Configuration and launch of the summarization via GUI; (v) Indexing of summaries via GUI; (vi) Browsing and full-text search; (vii) Access to summaries via APIs (viii) Autocomplete service over arbitrary strings.

2 Exploring and Understanding a Data Set with ABSTAT

ABSTAT controller[7] is designed to be modular and decoupled as in Fig. 1. The modules of ABSTAT 1.0 are the following:

– **ABSTAT Viewer** provides a graphic user interface to serve different types of tasks such as summary exploration, execution of the summarization process using a wizard and summaries indexing. Summary exploration can be performed using constrained queries (a desired subject and/or predicate and/or object) and full-text search. The summarization wizard provides a GUI to let users select datasets/ontologies from a populated list or using an upload module, configure and execute the summarization process. After the semantic profile is computed, the user can load/index it on a persistent storage/search engine in order to support its access through APIs or GUI.

[3] http://abstat.disco.unimib.it.
[4] http://www.ew-shopp.eu.
[5] https://bitbucket.org/disco_unimib/abstat.
[6] https://www.gnu.org/licenses/.
[7] http://backend.abstat.disco.unimib.it.

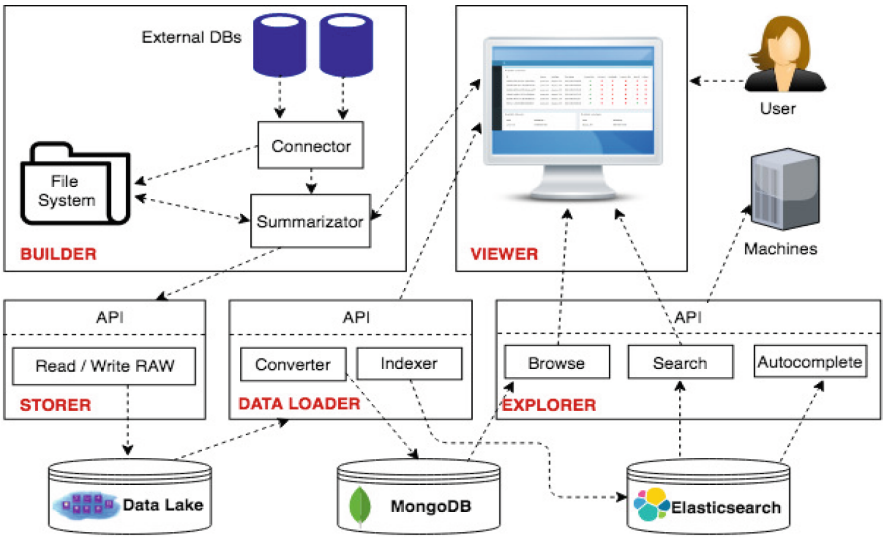

Fig. 1. ABSTAT architecture

- **ABSTAT Builder** is the module that executes the summarization algorithms and produces the profiles. The Summarizator component requires as input a dataset (in N3 format) and an ontology (in OWL format) along with the configuration chosen by the user. If the data are in an external DB, the Connector component allows extracting a dump and storing it in the correct file to serve as input to the Summarizator.
- **ABSTAT Storer** component feeds a data lake storage with the raw data produced by the Builder. It also receives download requests from users who want to get raw summaries.
- **ABSTAT Loader** contains the Converter component, which converts the data formats in the Data Lake in a format suitable for the Explorer module. The Indexer component indexes summaries in a search engine. Note that the Loader component receives the control input from the Viewer.
- **ABSTAT Explorer** is organized as a set of APIs to satisfy profile exploration requests from Viewer or users who want to use them directly.

3 Demonstration

ABSTAT is a framework that computes and provides access to semantic profiles that consist in an RDF summary and statistics. The summary of a data set describes its content by listing every schema-level pattern that occur in the data. In addition, semantic profiles provide several statistics about the occurrence of patterns, types and properties and cardinality statistics. During the summarization process if the user specifies the main pay-level domain of the data set (e.g., dbpedia.org for DBpedia), ABSTAT can distinguish between resources

(patterns, types and properties) that are internal (resources having the specified pay-level domain) and external (resources having a pay-level domain different from the one specified by the user). This distinction has the only purpose of letting users filter out patterns that include some external resource (e.g., hide all patterns that contain the type foaf:Person when looking at patterns extracted from DBpedia).

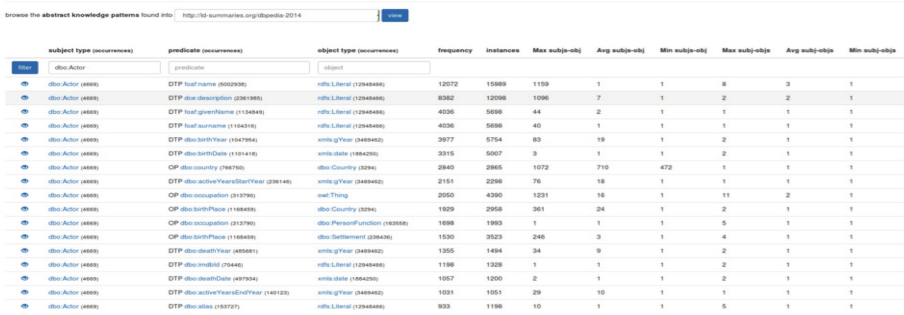

Fig. 2. ABSTAT browse GUI

Figure 2 shows the home page of ABSTAT. The menu on the left side can be used to explore semantic profiles. The **Overview** page gives an overview of the uploaded data sets, ontologies and computed profiles. **Summarize** page gives a configuration interface for custom summarizations including data sets and ontologies uploading. **Consolidate** allows to persist and index the computed profiles into the search engine. **Browse** is the GUI for constraint-based pattern exploration. **Search** is the GUI for full-text searching. Patterns, predicates and types that match the keyword will be returned. Search can be processed over the whole set of indexed profiles or on those originated from a specif data sets. Statistics, data set names and pattern symbols will be shown in the results of the query. **Manage** allows to remove data sets, ontologies and profiles. **APIs** lists the available APIs for machine-friendly profile exploration.

Patterns of the semantic profile are sorted by frequency in descendant order. The user can also put constraints on subjects and/or predicates and/or objects. In every text box a simple suggestion menu will recommend types/predicates that occur in the patterns. Then patterns are filtered in order to match the user constraints. Figure 3 shows the patterns that match the predicate `dbo:knownFor` and the object type `dbo:Film`. For each pattern several statistics are returned. Considering the one in the black box, the frequency of the pattern shows how many times does this pattern occur in the data set. The number of instances shows how many instances have this pattern including those for which the types `Person` and `Film` and the predicate `knownFor` can be inferred. Max (Min, Avg) subjs-obj cardinality is the maximal (minimal, average) number of distinct entities of type `Person` linked to a single entity of type `Film` through the predicate

`knownFor`. Max (Min, Avg) subj-objs is the maximal (minimal, average) number of distinct entities of type `Film` linked to a single entity of type `Person` through the predicate `knownFor`. Frequency is given also for types and predicates.

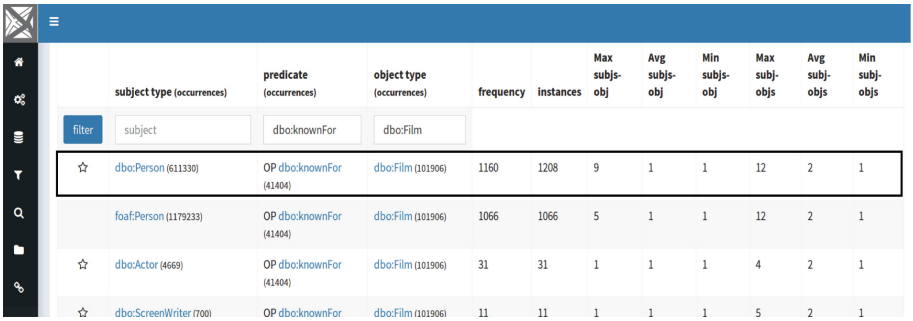

Fig. 3. Semantic profile of DBpedia 2014 data set

Previous experiments suggest that ABSTAT summaries help users in understanding a data set, e.g., by facilitating query formulation, and provide support to the assessment of data quality by finding outliers in the vocabulary usage [5]. In addition, we have recently found that rich profiles as the ones computed in ABSTAT 1.0 support automatic feature selection for semantic recommender systems, outperforming other purely statistical measures like Information Gain [1,3]. Finally, ABSTAT 1.0 supports vocabulary suggestions, similarly to [4]. In the future, ABSTAT will provide more significant statistics such statistics about class hierarchy depth, classes and properties per entity, etc.

Acknowledgements. This research has been supported in part by EU H2020 projects EW-Shopp - Grant n. 732590, and EuBusinessGraph - Grant n. 732003.

References

1. Di Noia, T., Magarelli, C., Maurino, A., Palmonari, M., Rula, A.: Using ontology-based data summarization to develop semantics-aware recommender systems. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 128–144. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_9
2. Palmonari, M., Rula, A., Porrini, R., Maurino, A., Spahiu, B., Ferme, V.: ABSTAT: linked data summaries with ABstraction and STATistics. In: Gandon, F., Guéret, C., Villata, S., Breslin, J., Faron-Zucker, C., Zimmermann, A. (eds.) ESWC 2015. LNCS, vol. 9341, pp. 128–132. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25639-9_25
3. Ragone, A.: Schema-summarization in linked-data-based feature selection for recommender systems. In: Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, 3–7 April 2017, pp. 330–335 (2017)

4. Schaible, J., Gottron, T., Scherp, A.: *TermPicker*: enabling the reuse of vocabulary terms by exploiting data from the linked open data cloud. In: Sack, H., Blomqvist, E., d'Aquin, M., Ghidini, C., Ponzetto, S.P., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9678, pp. 101–117. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-34129-3_7
5. Spahiu, B., Porrini, R., Palmonari, M., Rula, A., Maurino, A.: ABSTAT: ontology-driven linked data summaries with pattern minimalization. In: Sack, H., Rizzo, G., Steinmetz, N., Mladenić, D., Auer, S., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9989, pp. 381–395. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47602-5_51