Semantic Web 1 (0) –
IOS Press

Understanding the Structure of Knowledge Graphs with ABSTAT Profiles

Blerina Spahiu [a,*], Matteo Palmonari [a], Renzo Arturo Alva Principe [a] and Anisa Rula [b]
[a] *Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milano (Italy)*
E-mail: {blerina.spahiu, matteo.palmonari, renzo.alvaprincipe}@unimib.it
[b] *Department of Information Engineering, University of Brescia (Italy)*
E-mail: anisa.rula@unibs.it

Editors: First Editor, University or Company name, Country; Second Editor, University or Company name, Country
Solicited reviews: First Solicited Reviewer, University or Company name, Country; Second Solicited Reviewer, University or Company name, Country
Open reviews: First Open Reviewer, University or Company name, Country; Second Open Reviewer, University or Company name, Country

Abstract.
While there has been a trend in the last decades for publishing large-scale and highly-interconnected Knowledge Graphs (KGs), their users often get overwhelmed by the daunting task of understanding their content as a result of their size and complexity. Data profiling approaches have been proposed to summarize large KGs into concise and meaningful representations, so that they can be better explored, processed, and managed. Profiles based on schema patterns represent each triple in a KG with its schema-level counterpart, thus covering the entire KG with profiles of considerable size. In this paper, we provide empirical evidence that profiles based on schema patterns, if explored with suitable mechanisms, can be useful to help users understand the content of big and complex KGs. We consider the ABSTAT framework, which provides concise pattern-based profiles and comes with faceted interfaces for profile exploration. Using this tool we present a user study based on query completion tasks, where we demonstrate that users who look at ABSTAT profiles formulate their queries better and faster than users browsing the ontology of the KGs, a pretty strong baseline considered that many KGs do not even come with a specific ontology that can be explored by the users. To the best of our knowledge, this is the first attempt to investigate the impact of profiling techniques on tasks related to a content understanding with a user study.

Keywords: data understanding, data profiling, summarization, rdf, knowledge graph

1. Introduction

Knowledge Graphs (KGs), such as DBpedia[1], Wikidata[2], Google's Knowledge Graph and Microsoft's Satori, contain facts about a variety of different entities such as places, people, organizations, events, art works, and so on [28]. To support reuse and interoperability, hundreds of publicly available KGs are published using the RDF[3] data model, which is based on triples having the form <*subject*,*predicate*,*object*> [28]. In RDF graphs, nodes are connected by directed edges labeled with (RDF) *properties* and represent entities, literals (e.g., strings, numbers, etc.), or entity and data types[4]. Types and properties provide the *vocabulary* used to organize the KG, which may be also formally specified using

*Corresponding author. E-mail: blerina.spahiu@unimib.it.
[1] http://dbpedia.org/
[2] https://www.wikidata.org/
[3] https://www.w3.org/RDF/
[4] Entity types are usually referred to as *classes*; in this paper we prefer to use the term "entity type"to refer to classes, so that we can use the broader term "type"to refer to entity and data types

ontologies [21]. RDF-Schema[5] (RDFS), the simplest ontology language, supports the specification of subclass relations (e.g., *City* is subclass of *Place*), subproperty relations (e.g., *capitalOf* is subproperty of *locatedIn*), and domain and range restrictions, which constrain the usage of properties (e.g., subjects and objects of triples using the *capitalOf* predicate must be, respectively, of type *City* and *Country*). More expressive languages, e.g., OWL [6], allow to specify more fine-grained constraints [21].

Ontologies inform users about the structure of a KG by describing dependencies among types and predicates, and, especially, the types of entities that are expected with specific predicates. These dependencies can be inspected using ontology editing tools such as Protégé [46, 60] and are very important to effectively consume the data contained in a KG. Let us consider query answering, a quintessential data consumption task: to formulate a proper query, e.g., using the SPARQL[7] language, the user needs some prior knowledge about how the KG is structured. For example, consider the small subset of DBpedia represented in Fig. 1 a) and the following target query: *"In which films directed by Garry Marshall was Julia Roberts starring?"* To formulate this query, the user needs to know that actors and films are connected with the property starring, while films are also described with the property director.

Ontologies are helpful but often insufficient to fully understand the structure of a KG for a variety of reasons: (i) a KG may use a vocabulary that is not explicitly associated with a reference formal ontology (e.g., Linked Geo Data[8], but consider also Schema.org[9], which does not come with a specification in a formal language), or pick terms from multiple ontologies, (ii) the ontology may be underspecified, i.e., specifications cover only a few dependencies (e.g., in DBpedia several properties do not have domain and/or range restrictions [10]), (iii) KGs may not only be very large (e.g., Microsoft Academic Knowledge Graph[11] (makg)) but also have a complex ontology (e.g., the 2016-10 DBpedia ontology contains 467 types and 1.446 properties), (iv) KGs may use terms in a way that is not fully compliant with the ontology specifications, (v) some users may find it difficult to master ontological languages [53].

Some of these limitations can be mitigated by exploratory search [34], i.e., by formulating queries and retrieving results iteratively. However, this method is effort-consuming and can even fail when some of the exploratory queries match too many results (e.g., *find all the properties where the triple subject is a movie*) [59]. Several approaches have been proposed to help users overcome these challenges. Approaches based on *faceted search* [24, 45] and *query-by-example* [38, 39] allow users to query a KG without any knowledge about its structure. Others propose to exploit *data visualization* [7, 42] (including ontology visualization functionalities [30]) to understand structural properties. *Profiling and summarization* approaches [1, 2, 9, 13, 19, 22, 29, 31, 36, 37, 44, 47, 52, 57] are more specifically targeted to help users (and/or machines) understand the structure of a KG, its properties, and its more salient content. While "understanding" intuitively refers to the ability to know how something works or what something means, we could not find any agreed definition about what *understanding the structure of a KG* actually means. Following some previous work [33], in this paper we refer to knowledge graphs *understanding* as the process of gaining insights by accessing and exploring a *set of simple structures* that are easily *understood and meaningful*.

Pattern-based profiling and summarization approaches [5, 11, 32] such as ABSTAT[12] [4, 59] and Loupe [44] use vocabulary usage patterns - referred to as *schema-level patterns* in the following - as primitives to represent these simple structures, under the assumption that they can be easily understood by users. These schema-level patterns have the form $(Type, Predicate, Type)$, e.g., $(\mathsf{Actor}, \mathsf{starring}, \mathsf{Film})$, and can be associated with different numerical attributes, e.g., frequency. A profiles consists in a set of patterns extracted from a KG with the help, if available, of an ontology. In the example graph depicted in Fig. 1, a user would inspect the patterns represented in Fig. 1 b) to understand the structure of the KG and the usage of types and properties in the KG. Such patterns might be considered as "views" that allow to speed up knowledge discovery. These profiles have been proved to support different downstream computational tasks [16, 18, 51, 58]. ABSTAT profiles cover

[5] https://www.w3.org/TR/rdf-schema/
[6] https://www.w3.org/TR/owl2-primer/
[7] https://www.w3.org/TR/sparql11-query/
[8] http://linkedgeodata.org/
[9] https://schema.org/
[10] Underspecification is often well justified and not imputable to quality issues
[11] http://ma-graph.org/
[12] http://abstat.disco.unimib.it/

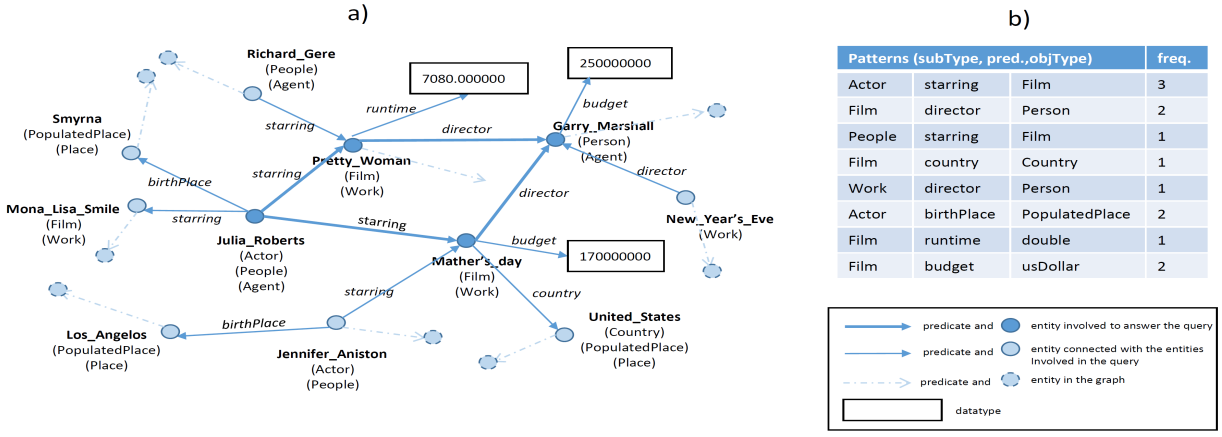

Fig. 1. Knowledge graph and extracted patterns

all the triples in the KG, which means that each triple is represented by at least one schema-level pattern. The number of patterns depends therefore on the number of different usage patterns that occur in a dataset and may be quite large for large KGs using complex ontologies, e.g., DBpedia. To reduce the number of patterns ABSTAT uses a minimalization mechanism to retain only the most specific patterns for each triple. In addition, it comes with user interfaces that support search, filter and exploration over profiles, thus helping understand also complex KGs with large amount of patterns [50].

The main objective of this paper is to provide empirical evidence that pattern-based profiles, explored from suitable user interfaces, can be useful to help users understand the structure of large and complex KGs, and, especially, when compared to ontology exploration frameworks. To evaluate our hypothesis, we present a user study based on query completion tasks over DBpedia: the users' ability to understand the KG structure is measured downstream by their ability to fill in schema-level elements in SPARQL queries using ABSTAT or Protégé as supporting tools. Protégé is a well-known, highly adopted and mature tool that provides several functionalities to explore ontologies. DBpedia is a large and complex KG that has its own specific OWL ontology that can be explored in Protégé (we remark that this condition does not apply to all KGs, which can still be explored through their ABSTAT-based profiles). Our findings suggest that users supported by ABSTAT formulate their queries better and faster than users supported by Protégé. In the paper we report a detailed analysis of our findings and the feedback collected during the study, which provides insights into the difficulties that users encounter when they have to choose the vocabulary to formulate queries. We release all the data collected in our study for future research.

To the best of our knowledge, this is the first attempt to investigate the impact of KG profiling and summarization approaches on abilities that are related to content understanding. While our study focuses on pattern-based approaches, we believe that the proposed evaluation methodology can be adapted to evaluate future KG profiling and summarization approaches based on different principles.

In this paper we make the following contributions:

- Provide an extended analysis of the conciseness of the summary computed by ABSTAT.
- Propose a methodology to evaluate how a profiling tool helps users in understanding the data through the assignment of cognitive tasks.
- Construct and present a user study based on query completion tasks where users make use of profiles to complete their queries.
- Apply the proposed methodology to evaluate ABSTAT profiles from a user understanding prospective and demonstrate empirically its superiority over a baseline approach.
- Make publicly available all questionnaires and their results so further research and investigation can be made.

This paper is structured as follows: Section 2 introduces the ABSTAT summarization framework and presents ABSTAT profiles, the Web application and provides an analysis of the conciseness of the profiles.

The design of the user study is presented in Section 3 while the empirical evaluation of the user study and the quality analysis of the results is discussed in Section 4. Related work of approaches and tools that support data understanding is reviewed in Section 5 while Section 6 summarises some reflection and lessons learned. Finally, conclusions end the paper in Section 7.

2. ABSTAT Profiler

In this section we present ABSTAT framework that computes and provides access to semantic profiles. First, we present and describe profiles content in Section 2.1 and then in Section 2.2 we present ABSTAT Web application. Finally, in Section 2.3 we discuss how minimalization, a distinctive feature of ABSTAT, allows the creation of profiles that are compact and concise with respect to the content of the KG.

2.1. ABSTAT Profiler

ABSTAT is a data profiling framework aiming to help users understanding the content of big data set by exploring its semantic profile [4]. It takes as input a data set and an ontology (used by the data set) and returns a semantic profile (Fig.2). Thanks to the highly distributed architecture, ABSTAT is able to profile very big KGs [4]. The semantic profile produced by ABSTAT consists of a summary of patterns and several statistics. The key aspect of the summary is the use of minimal type patterns that represent an abstraction of the data set. A minimal type pattern is a triple (C, P, D) that represents the occurrence of triples $< a,P,b >$ in the RDF data, such that C is the minimal type (most specific type among all the types) of the subject a and D is the minimal type (most specific type among all the types) of the object b according to a terminology graph, which is introduced to represent the data ontology. For example, consider the pattern *(Book, publisher, Publisher)* shown with the black box in Fig. 2. This patterns indicates that there are triples with the predicate *publisher* in the data set, that have *Book* as the most specific type among the types of the subject and *Publisher* as the most specific type among the types of the object. Finally, the semantic profile also includes several statistics. In the following we provide an overview of the statistics produced by ABSTAT considering the highlighted pattern in Fig. 2:

- **occurrence** (in orange) for types, predicates and datatypes. This statistics gives the occurrence of the respective types, predicates and datatypes in the data set. For example, in the sample of DBpedia, the types Book and Publisher occur 31029 and 1141 times respectively, while the predicate publisher occurs 70339 times.
- **frequency** (in green) of the pattern shows how many times the pattern (minimal) occurs in the data set.
- **instances** (in green) for patterns shows how many instances have this pattern including those for which the types *Book* and *Publisher* and the predicate *publisher* can be inferred.
- **cardinality statistics** (in turquoise) provide information about cardinality estimation for a given predicate. Max (Min, Avg) subjs-obj cardinality is the maximal (minimal, average) number of distinct entities of type Book linked to a single entity of type Publisher through the predicate publisher.

For a formal and complete definition of the profiling model of ABSTAT please refer to [4].

2.2. Accessing Profiles with the ABSTAT Tool

An ABSTAT profile provides a good abstraction over a data set but it would not be much helpful without a proper access, navigation and presentation of results. ABSTAT tool fulfills this need through a Web application that computes and provides access to profiles[13]. To help users explore the information represented in the profiles, ABSTAT provides two graphic user interfaces (GUIs).

Browse. Suppose the user wants to explore a data set that has been profiled with ABSTAT. Since the user often does not know much about the data, she/he probably does not have a clear idea about what to search. First of all, the user selects the profile to inspect as shown in top-most part of Fig. 3. Patterns are sorted by frequency in order to present a small snapshot about the most frequently used patterns in the data. Subsequently the user can decide to filter the data using the three available text boxes for adding constraints on the subject type, predicate, object type and any combination of these. While the input is typed inside a text box the autocompletion feature will recommend types/predicates that occur in the patterns (the bottom part of Fig. 3 shows how predicate text box

[13] http://abstat.disco.unimib.it/

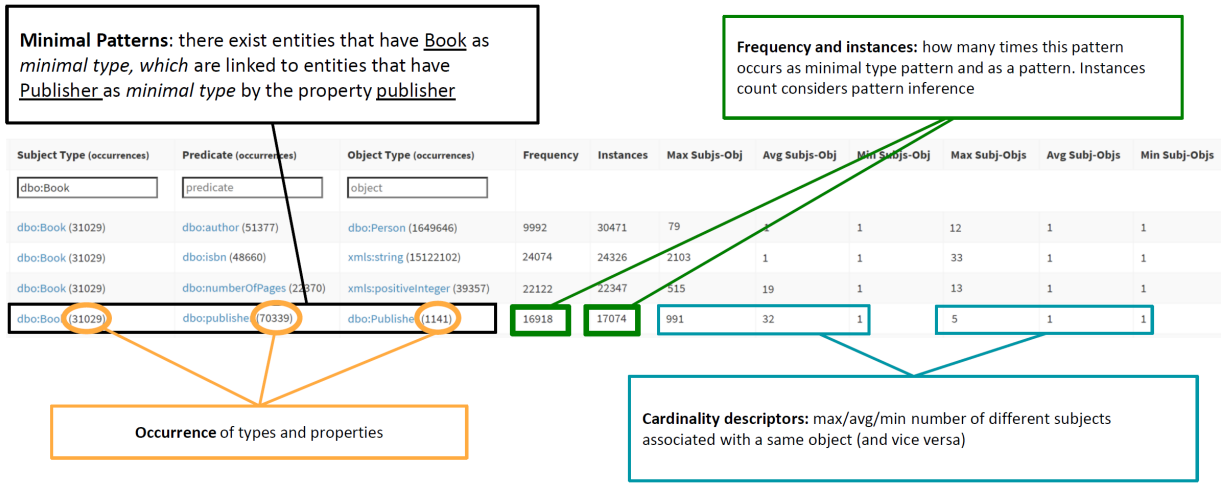

Fig. 2. A sample of the Semantic Profile of the DBpedia 2015-10 data set

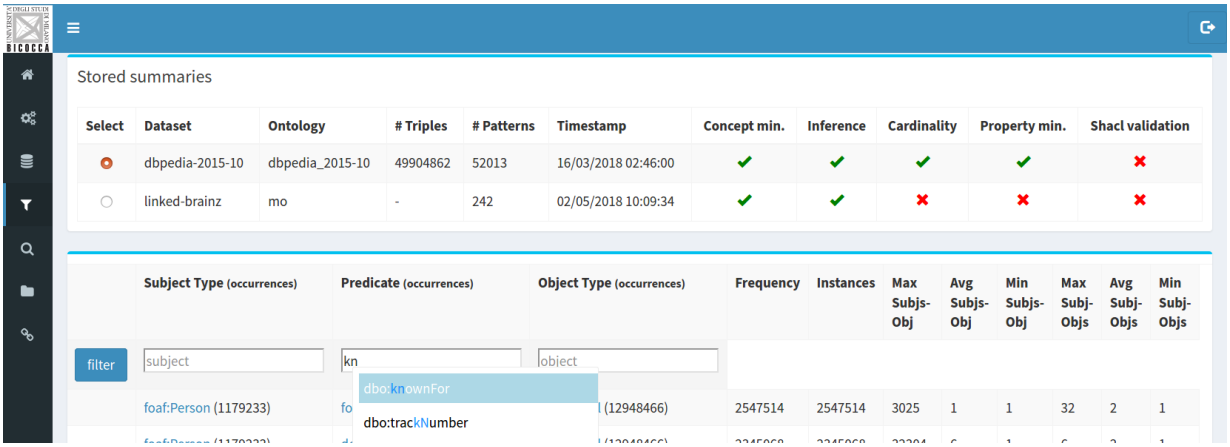

Fig. 3. Semantic Profile exploration of the DBpedia 2015-10 data set (Browse GUI)

suggest `dbo:knownFor` and `dbo:trackNumber` for the input string "kn"). For example, by filtering patterns in the subject box with the constraint `dbo:Film`, patterns with interesting predicates such as `dbo:starring`, `dbo:producer`, `dbo:musicComposer`, `dbo:budget` will be returned, which explain what kind of information we can retrieve from the data set with respect to movies. Therefore, we can see how ABSTAT guides the user to understand a data set through Browse GUI.

In addition, the GUI shows statistics associated with the patterns. Fig. 3 shows the patterns that match the predicate `dbo:knownFor` and the object type `dbo:Film`. Considering the one in the black box, in the sequence order we have; occurrence of subject type, predicate and, object type, frequency of the patterns (number of *minimally* represented assertions) and additional statistics. More in details, statistics for this pattern tell that there are 1.160 relational assertions $P(a,b)$ such that $(\texttt{dbo:Person}, \texttt{dbo:knownFor}, \texttt{dbo:Film})$ is a minimal type pattern for $P(a,b)$. Moreover, there exist 611.330 individuals of type `dbo:Person`, 101.906 individuals of type `dbo:Film` and 41.404 relation assertion $P(a,b)$ such that P is `dbo:knownFor`. In addition, *instances statistic* shows that there are 1.208 relational assertions represented by this pattern (including those minimally represented by more specific

patterns). Finally, *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`. For more details about these additional statistics we refer to [50].

Search. This interface is the GUI for full-text search, where the user can insert any keyword and get results that match the input (patterns, types and predicates) for all the profiles or some specific profile. Statistics, data set names and patterns will be shown in the results of the query. Fig. 4 shows the results for the input "influenced" over the dbpedia-2015-10 data set. Notice that for each result a colored label indicates if it is a predicate, a type or a pattern. The first two results show information about properties, followed by two patterns. Frequency is shown for patterns and occurrence for types and properties. For example, `dbo:influencedBy` is an object property that occurs 10.676 times in the dbpedia-2015-10 data set.

GUI-based access to ABSTAT profile is more relevant to the scope of this paper. However, it is worth noting that ABSTAT supports API-based access to control the profile process (e.g., launch a summarization, store profiles, etc.) and to make third-party applications access the profiles. APIs that provide access to the profiles support all the functionalities that are available with the Browse and Search GUIs. In addition, vocabulary suggestion APIs have been developed. They serve vocabulary suggestions and pattern search (**Suggestions-APIs**) in semantic table interpretation and annotation tools [14–17] and, with more sophisticated ranking functions, feature selection in KB-based recommender systems [18, 51].

2.3. Conciseness of ABSTAT Profiles

Understanding data sets with large numbers of concepts, relationships, and entities is a hard problem, since their presentation could easily overload the user with information and prevent them to reach an overall understanding or to find particular information [49]. In this section we measure the conciseness of the summary produced by ABSTAT and compare it with the conciseness achieved by Loupe [44] an approach similar to ours that does not use minimalization. From all the available works on profiling we compare with Loupe for two reasons (see Section 5): (i) many of the works on profiling do not provide an instrument to explore profiles, and (ii) from all the available tools, Loupe profiles are most similar to ABSTAT, with the difference that ABSTAT adopts the minimalization technique.

For this comparison we use summaries extracted from different RDF data sets: different versions of DBpedia (2014-en[14], 2015-10[15], 2016-10[16]), LinkedBrainz[17], DrugBank[18], pharmagkb[19], linkedgeodata[20], geonames[21], geonames-mod (a version of geonames where codes[22] were substituted with their specific label for two reasons: (i) to have a better compression rate, and (ii) to have a summary that is better understandable by humans who can easily read labels instead of codes) and makg[23].

Table 1 provides a quantitative overview of data sets and their summaries. To assess the conciseness of a summary we measure the *compression rate* (C.R.), defined as the ratio between the number of patterns in a summary and the number of assertions from which the summary has been extracted.

We compare the compression rate (C.R.) achieved by our model with respect to Loupe. ABSTAT achieves a *compression rate* of ∼0,003 for **dbp2014-en**, ∼0,009 for **dbp2015-10-en**, ∼0,0027 for **dbpedia2016-10-en** and ∼4,1e-7 for **linkedbrianz**. Comparing the compression rate obtained by our approach with the one obtained by Loupe (∼0,05 for dbp2014-en, ∼0,114 for dbp2015-10-en, ∼0,0033 for **dbpedia-2016-10-en** and ∼4,1e-7 for linkedbrianz) we observe that the summaries computed by ABSTAT are more concise, as we only include minimal type patterns. Loupe instead, does not apply any minimalization technique thus its summaries are less concise. For instance, a user that explores the profiles of **dbpedia-2016-10-en** who is particularly interested in exploring all patterns that have `dbo:birthPlace` as predicate and `dbo:Place` as their object type, has to explore 111 patterns using ABSTAT and 395 using Loupe (almost $3,5$ times more patterns). Although the compression

[14] http://downloads.dbpedia.org/2014/en/
[15] http://downloads.dbpedia.org/2015-10/core-i18n/en/
[16] http://downloads.dbpedia.org/2016-10/core-i18n/en/
[17] http://www.linkedbrainz.org/LinkedBrainz201712.tgz
[18] https://www.drugbank.ca/releases/latest
[19] https://www.pharmgkb.org/
[20] https://hobbitdata.informatik.uni-leipzig.de/LinkedGeoData/downloads.linkedgeodata.org/releases/
[21] https://download.geonames.org/export/dump/
[22] https://www.geonames.org/export/codes.html
[23] http://ma-graph.org/

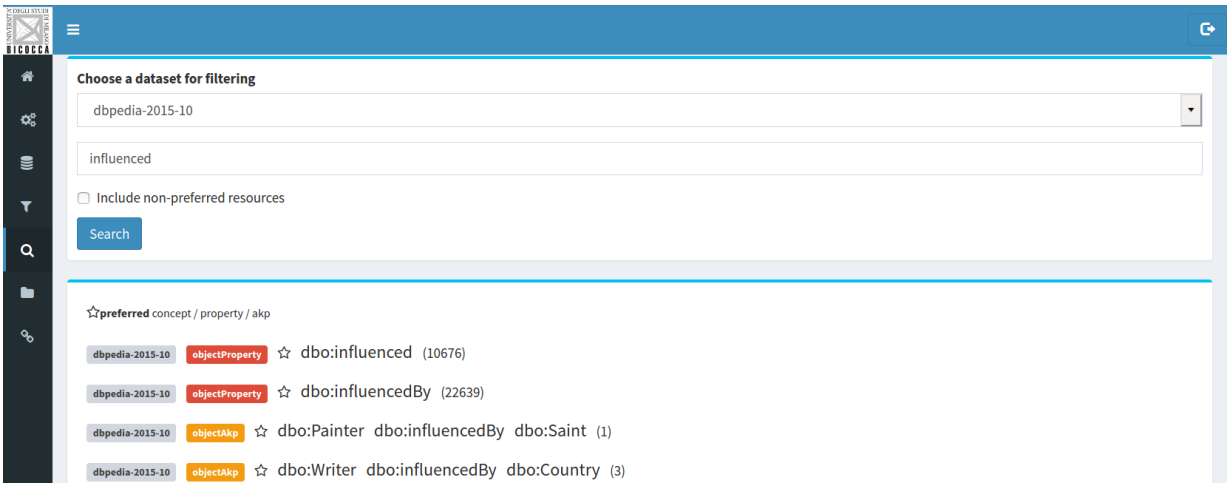

Fig. 4. Semantic Profile search for DBpedia 2015-10 data set (Search GUI)

Table 1
Statistics about relations, assertions, types, properties and patterns used in the data set

	Relational	Typing	Assertions	Types* (Ext.)	Properties* (Ext.)	Patterns	No minim. / (Loupe)	C.R. ABSTAT / Loupe
linkedbrainz	∼208.9M	∼29.7M	∼238.6M	12 (9)	29 (10)	98	99	4,1e-7/4,1e-7
dbp2014-en	∼521.6M	∼44,4M	∼566M	528 (79)	57451 (56015)	1636629	2919869	0,003/0,005
dbp2015-10-en	∼606.4M	∼75.2M	∼221.7M	2424 (1918)	62556 (61121)	21298654	25195035	0,09/0,114
dbp2016-10-en	∼ 2.4B	∼324.8M	∼2.7B	1220 (753)	122973 (121527)	7324742	8922021	2,7e-3/3,3e-3
drugbank	∼3.9M	∼773.6K	∼4.7M	102 (91)	102 (59)	1403	1403	3e-4/3e-4
pharmgkb	∼2.7M	∼40.6K	∼2.8M	57 (51)	75 (67)	644	644	2e-4/2e-4
linkedgeodata	∼818.8M	∼397.7M	∼1.2B	1143 (1143)	33347 (33347)	398297	398297	3e-4/3e-4
geonames	∼164.9M	∼11.7M	∼176.6M	1 (0)	24 (6)	27	27	1e-7/1e-7
geonames-mod	∼141.6M	∼11.7M	∼153.3M	682 (681)	26 (6)	4691	15760	3e-5/1e-4
makg	∼7.4B	∼744.1M	∼8.1B	13 (5)	50 (3)	233	233	2e-7/2e-7

* The number of types and properties refer to the total number of types and properties used in the data set (internal and external to the ontology). In brackets it is given only the number of external types and properties with respect to the ontology.

rate, even for ABSTAT profiles, humans still have to explore manually or by eye-balling many patterns. Producing even more compact summaries by reducing the number of patterns to explore i.e., patterns might be grouped in super patterns that the user might afterward zoom in for more specific patterns, is a task that we consider for future work. However we should mention that, having a low compression rate is not our goal; it is only our means to find redundant patterns. Low compression ratios are exactly a sign that many redundancies are discovered (i.e., patterns) which can be inferred and thus, users explore a smaller summary and understand the input graph better.

The effect of minimalization is more observable on DBpedia data sets, since the DBpedia terminology specifies a richer terminology graph and has more typing assertions. The same effect is evident also for geonames-mod, for which ABSTAT achieves a compression rate smaller than Loupe. This happens because once we substitute in geonames the codes with their label, we obtain a richer terminology graph. Observe also that, data set such as dbp-2015-10-en, dbp-2016-10-en, linkedgeodata and genoames-mod are the ones with more external types and properties. Such external types and properties are added to their terminology graph during the minimal types computation phase

as they were not part of the original terminology, and thus are considered by default as minimal types.

ABSTAT summaries are more concise with respect to the size of the data set itself. Although ABSTAT gives its best in terms of compression rate when summarization is applied over a data set with a terminology graph characterized by a rich type hierarchy, summaries remain powerful also with a poor terminology or even when it is missing. In case of a missing ontology the minimal type calculation is skipped and every type that describes an individual is included in the profile. In this case, the pattern frequency is equal to the pattern occurrence since there is no minimalization.

3. Design of the experimental study

In this section we introduce the design of the experimental settings to evaluate how profiling tools help users in knowledge graph understanding. We must observe that defining what precisely means, for a user, to understand a KG (or, in general, a data set) is not trivial. We are not aware, for example, of previous work providing a conclusive definition of *data understanding*, or a set of agreed-upon methods to measure understanding by humans. As we stated in Section 1, we measure data understanding by setting up a user study where users are assigned some cognitive tasks [54]. Users performance can be quantitatively measured and is used as proxy for evaluating a cognitive process [56]. Our proposed experimental settings encompasses five sequential steps, which need to be executed in the specified order, since the outputs of previous steps feed the tasks of the next steps. In the following we provide details for each step:

1. **Context definition:** The first step of the experimentation regards the definition of the context i.e., information gathered about the data set and the tested systems. First, we need to specify the domain of interest and the related data sets for which a summary needs to be executed. Second, we need to identify the systems to be compared. They should: (i) have similar aim, (2) be available, and (3) be comparable on the output that they provide to the user.
2. **Tasks definition:** The second step of the experimentation refers to the definition of tasks. There might be different dimensions to measure data understanding e.g., quantifying query completion, data exploration for natural language question answering, etc. In this paper we consider query completion as a component of data understanding. Different approaches might use different tasks to evaluate data understanding from their perspective, however, one of the challenges when designing user study experiments is the number of tasks that users have to complete as they are time-consuming and fatigue bias need to be reduced [48].
3. **User profiling and recruitment:** This step regards the definition of a user model to assess the KG understanding through the tasks defined in step 2. A user can be someone who might have some knowledge and/or is familiar with the technologies related to the defined tasks. Once the user profile is defined, the next step is to determine the recruiting process. For the recruitment one might consider: public mailing lists, social communities (Facebook, Tweeters, etc.) or people from laboratories or department working on similar technologies related to the defined tasks. However, for the new system which is being tested, supporting material that present the aim and the functionalities of the system should be provided to the users at the beginning of the experiments.
4. **Questionnaire composition:** This step regards, the design of the questionnaire that, on the basis of the data set chosen, the user profiling and task defined in the previous steps, can include several blocks. For example, one might consider to dedicate a first block of questions to gather background information about users participating in the survey; a second block regards questions to complete the defined tasks; and a third block comprises a set of feedback questions for the task and general comments about the survey at the end. Moreover, the environment where the survey is being executed should be the same for all users such that the results are comparable.
5. **Evaluation metrics:** The final step considers the design of the evaluation process. In order to compare the performance of users that complete their tasks for both systems a set of metrics should be set. The performance might be compared in terms of precision, recall, f-measure, accuracy and time. Moreover, a set of questions for the qualitative analysis should be set in order to gain a deep understanding of users behaviour and the characteristics of the systems being evaluated.

4. Evaluation

In this section we present and discuss the practical application of the user study. First, we present in Section 4.1 how we applied each phase of the experimental settings in a real user study. Second, in Section 4.2 we analyse the results of the user study, and third, in Section 4.3 we provide a thorough qualitative analysis of the results of the experiment.

4.1. User Study

In the following we describe the user study:

Context definition: Firstly, we choose the two systems to be compared and the data set on which to apply the user study. We choose to use Web Protégé as the baseline tool. There are three main reasons for such choice: (1) There are no other up and running tools for data profiling (except LOUPE); (2) Ontologies help users understand the data at hand as they describe the data by making semantics explicit; and (3) Both tools, ABSTAT and Protégé, give information about schema (the ontology in Protégé and schema-patterns in ABSTAT) not about instances or entities. DBpedia is the data set that users needed to explore and understand. DBpedia is one of the most important data sets of the Semantic Web community as it contains real, large scale data and is complex enough with 449 classes and 1436 properties. It has a documented schema which might be downloaded easily[24]. All the above reasons make DBpedia challenging enough to assess the abilities of users to understand the data by exploring ABSTAT profiles or the ontology in Web Protégé.

Tasks definition: We designed a user study based on the assignment of cognitive tasks related to query completion. We selected a set of queries from the Questions and Answering in Linked Open Data benchmark[25]. Such queries are believed to be representative of realistic information needs [40], although we cannot guarantee that they cover every possible information need. To gather evidence and evaluate data understanding, we ask users to complete and answer to three queries from DBpedia 2014, using either ABSTAT or Protégé. In order to answer to the selected queries we require users to explore the profile of DBpedia provided by ABSTAT or by exploring the ontology in Web Protégé. For each query we provide a "template" of the corresponding SPARQL query, with spaces intentionally left blank for properties and/or concepts. For example, given the natural language specification "Which is the second highest mountain on Earth?", we ask participants to fill in the blank spaces:

```
SELECT DISTINCT ?uri WHERE
?uri rdf:type .... .
?uri .... ?elevation .
ORDER BY DESC(?elevation)
OFFSET 1 LIMIT 1
```

The three queries are of different length, defined in terms of the number of triple patterns within the WHERE clause; one query of length one, one of length two and one of length four. Using 3 queries is coherent with other related work which suggest that the user study would have 20-60 participants, who are given 10-30 minutes of training, followed by all participants doing the same 2-20 tasks, during a 1-3 hour session [48]. Both groups execute SPARQL queries against the DBpedia 2014 data set through the same interface and were asked to submit the results they considered correct for each query.

User profiling and recruitment: For the aim of our user study, we designed an online survey, which was distributed in the semantic-web and public-lod public mailing lists of the W3C[26]; in the internal mailing lists of the affiliation lab of the authors; and social communities in Facebook and Twitter. As the survey was distributed online, no special recruitment is required. In the introductory page of the survey we explain for all users the scope of the survey and give instructions on how to complete the survey. For users who choose to answer queries using ABSTAT, we strongly recommended to watch an explanatory video about how to use ABSTAT[27]. For users who choose to answer the queries using Protégé, we did not give any training as we assume that users who choose to take such survey are familiar with SPARQL and ontologies.

Questionnaire composition: The survey is designed in three parts and would take in average around 30 minutes to be completed. In the first part we ask users 6 background questions in the form of choosing the best answer which describe them. The second part is about query completion, while in the third part of the survey, we ask all users 7 feedback questions for each query and at the end general comments about the survey.

[24] https://wiki.dbpedia.org/services-resources/ontology
[25] http://qald.aksw.org/index.php?x=home&q=home
[26] https://lists.w3.org/Archives/Public/public-lod/2016Dec/0003.html
[27] https://www.youtube.com/watch?v=Gn_-WLM1utU

Evaluation metrics: To evaluate the performance of users in completing the survey, we measure the time spent to complete each query and the correctness of the answers. The correctness of the answers is calculated as the ratio between the number of correctly answered types and/or properties to the given query against the total number of query variables. We also run different statistical tests in order to evaluate the significance of the obtained results. Moreover, for the qualitative analysis we defined a set of questions to understand the users behavior and systems characteristics.

Once we specify and apply all steps of the methodology we define a set of questions to answer with our user study:

Q1. Does ABSTAT help users to complete queries more accurately with respect to the baseline?

Q2. Does ABSTAT help users to complete queries faster than using the baseline?

Q3. Which is the target group that would take more advantage of ABSTAT profiles and for what kind of queries?

Q4. How intuitive is ABSTAT?

4.2. Quantitative Analysis

In total 117 users completed the survey. Before analysing the results we performed the data cleaning process. Data cleaning is performed in order to remove incorrect information or information that could skew the data. We removed from our data three cases:

- the data from two users who opened and left the survey in stand by (the time spent to answer one single query was ≈2 hours.
- the data from one user who completed only the first part of the survey (background questions)
- the data from a user who took the experiment twice (duplicated mail address) with one having a bias toward the other as the second experiment had a minor time in completing the whole survey.

After cleaning the responses from dirty or useless data, we considered 113 answers; 59 users answered the queries using ABSTAT and 54 using Protégé. Not all the users completed the whole survey: 113 (54 ABSTAT and 49 Protégé) completed the first query, 105 (55 ABSTAT and 50 Protégé) the first two queries and only 103 (54 ABSTAT and 49 Protégé) answered to all the three queries.

In the following we analyse and respond to each of the questions introduced at the beginning of this section.

Q1. Does ABSTAT help users to complete queries more accurately with respect to the baseline?

Table 2 provides the distribution of the number of users and their percentage in answering the survey for each group. The number and the percentage are given for users who correctly answer to each query and those who did not.

Response to Q1: We can observe that for each query, users who choose to answer queries with ABSTAT achieve a higher accuracy (the ratio between the number of correctly answered types and/or properties to the given query against the total number of query variables). There is a notable difference between the number of users who correctly answer to all queries using ABSTAT and those who used Protégé. This effect is notable for all queries despite their difficulties.

Q2. Does ABSTAT help users to complete queries faster with respect to the baseline?

In order to answer to Q2, we performed the T-test to understand if there is a statistical difference between the time needed for users from both groups to correctly answer to queries in the survey. A T-test's statistical significance indicates whether or not the difference between two groups' averages time in answering to the queries, likely reflects a "real" difference in the population from which the groups were sampled[28]. ABSTAT users performed the task of query completion faster for the most difficult queries still assuring a high accuracy. We interpret the latter trend as a classical cognitive pattern, as the participants became more familiar with ABSTAT interface.

Response to Q2: For our experiments the independent T-test, showed that the time needed to correctly answer to the first two queries, the most difficult ones, was statistically significant between two groups. Moreover, the average time to answer to the easiest one is almost in the same range, even though smaller for ABSTAT users (-66,4sec). Table 3 reports the value of T-test and the average time for correctly answering to each query from both groups.

Q3. Which is the target group that would take more advantage of ABSTAT profiles and for what kind of queries?

In order to answer to the above question we profiled all the participants in terms of knowledge about SPARQL, data modelling, DBpedia data set and ontology. As the aim of the experiment is to evaluate query

[28] A statistically significant T-test is when the difference between two groups is unlikely to have occurred because the sample happened to be atypical.

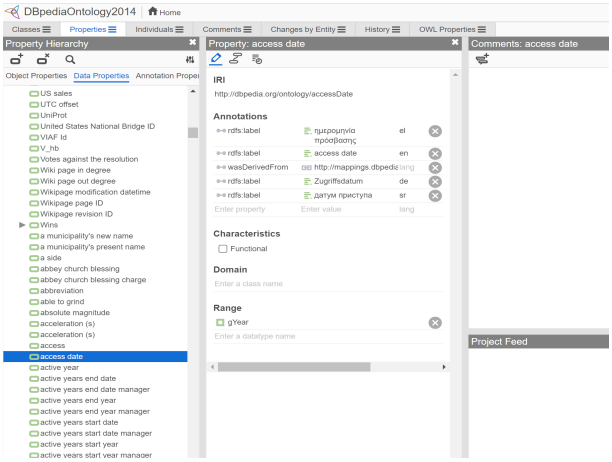

Fig. 5. Screenshot of DBpedia Ontology in WebProtégé

Table 2

The distribution of the number (and the percentage) of users from both groups for each query

		ABSTAT	Protégé	Total
Query 1: "Which is the second highest mountain on Earth?"	Correct	54 (47,8%)	49 (43,4%)	103 (91,2%)
	Not Correct	5 (4,4%)	5 (4,4%)	10 (8,8%)
Query 2: "Which German cities have more than 250000 inhabitants?"	Correct	23 (21,9%)	9 (8,6%)	32 (30,5%)
	Not Correct	32 (30,5%)	41 (39,0%)	73 (69,5%)
Query 3: "Who is the Formula 1 race driver with the most races?"	Correct	47 (45,6%)	40 (38,9%)	87 (84,5%)
	Not Correct	7 (7%)	9 (9%)	16 (15,5%)

Table 3

The average time to answer to queries for both groups and the T-test significance

		ABSTAT	Protégé
Query 1: "Which is the second highest mountain on Earth?"	Avg. time (sec)	165,3	948,6
	t-test	t(101) = -8,198; p <0.001	
Query 2: "Which German cities have more than 250000 inhabitants?"	Avg. time (sec)	979,0	1861,6
	t-test	t(30) = -3,108; p <0.004	
Query 3: "Who is the Formula 1 race driver with the most races?"	Avg. time (sec)	309,3	375,7
	t-test	t(85) = -1,545; p<0,13	

completion task, we report in Fig. 6 for all users, only the average time for correctly answering to each query for each level of SPARQL knowledge that participants reported.

Response to Q3: The average time needed for all participants to correctly answer to the queries, regardless of the SPARQL knowledge, is lower for ABSTAT user than Protégé users. For the first query, even for Protégé users who reported to have "Good Knowledge", the time needed is almost 10 times greater than for ABSTAT users. While for the second query, in general users from the Protégé group required more time to answer the query despite their SPARQL knowledge. However, only for user that have "Good Knowledge" the time is slightly higher for ABSTAT users (637s vs 523s). Finally for the third query, the easiest one, the average time between two groups is relatively in the same range for each SPARQL knowledge level participants report. In general, with respect to Protégé, ABSTAT helps all users despite their SPARQL knowledge to answer correctly and in less time to all queries. Moreover, ABSTAT users took advantage of ABSTAT profiles in answering correctly to medium and more difficult queries.

Q4. How intuitive is ABSTAT?

To answer such question we gather evidence on the usability of ABSTAT with feedback questions: (i) We asked users about their perception of the difficulty in

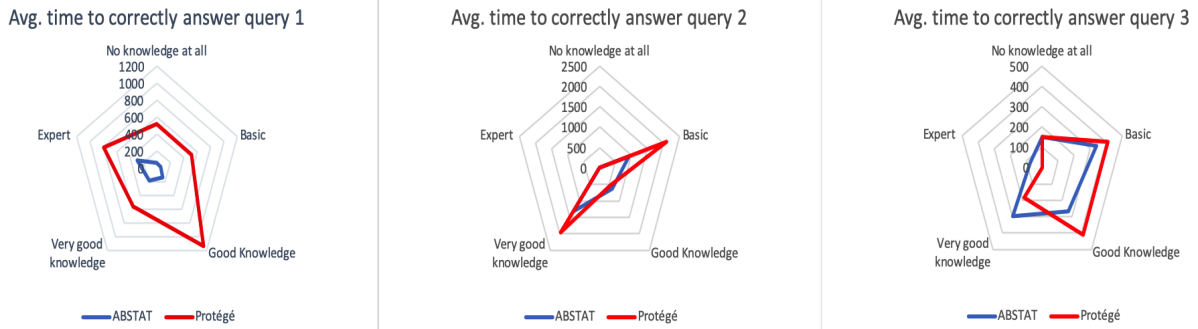

Fig. 6. The average time in answering correctly to the queries for both groups with respect to their SPARQL knowledge

answering the queries, (ii) For each query we asked the number of attempts users submitted to correctly answer them, (iii) Users reported if they made use of other tools / information to answer the queries, and (iv) We analyse users performance on answering queries correctly with respect to the fact that they watched the introductory video of ABSTAT.

First, participants reported their perception about the difficulty of answering queries by means of the tool chosen for the survey. Fig. 7 shows the perception of the participants who answer correctly to the three queries for both groups. ABSTAT users interpret each query to be easier than Protégé users. None of the participants from ABSTAT group reported any query as "Very difficult". Moreover, the number of the participants from Protégé group who reported the query to be "Difficult" is greater than ABSTAT users and vice versa the perception of the simplicity for each query. Even for query number 2, the most difficult one, 6% of the users who belong to ABSTAT group reported the query to be "Easy", while none of users from Protégé reported such query as easy.

In order to determine whether there is a significant difference about users perception we use the Mann-Whitney test [26]. The null hypothesis is: There is no evidence for an association between the difficulty in answering the queries and the tool used. The Mann-Whitney test for the perception of the difficulty in answering to the first two queries, has the p-value smaller than our chosen significance level ($p = 0.05$), thus we reject the null hypothesis. Rather, we conclude that there is enough evidence to suggest an association between the tool used to complete the queries and the perception of the difficulty in answering them. Participants using ABSTAT perceive the queries to be less difficult than those using Protégé.

Secondly, we asked both groups the attempts to correctly answer each query. Table 4 shows within each group the percentage of the users and the attempts made to correctly answer to each query. For all queries the percentage of the users who correctly answered with the first try is greater for ABSTAT, while the percentage of the users who made several attempts (for both properties and classes) is greater for Protégé users. Moreover, for query 2, the most difficult one, more than half of Protégé users made several attempts for both classes and predicates. Moreover, for the second query, the most difficult one, there are no Protégé users who answered with the first attempt, and more than half of them made several attempts, while there are around 18% of ABSTAT users who answered with the first attempt.

Third, we asked users if the information provided by ABSTAT or Protégé was enough to answer the queries. Fig. 8 shows the distribution in percentage of the users who used only the tool (ABSTAT or Protégé) chosen at the beginning of the survey, to answer queries. For each query, the percentage of the users who choose to make use of other sources to answer the queries, is almost twice greater for Protégé users. For the second query, the number of users who need also other support is around 40% for Protégé group. The main reason why ABSTAT users reported to use other tools to answer to the queries, is that they were more familiar with the other chosen tools (DBpedia data set, Google, etc). The comparison between the two groups shows that the percentage of the users who use other methods except the one requested by the survey to answer to the queries is greater for Protégé users. None of the users who answer to the queries using ABSTAT made a research in Google while this was quite often for Protégé users.

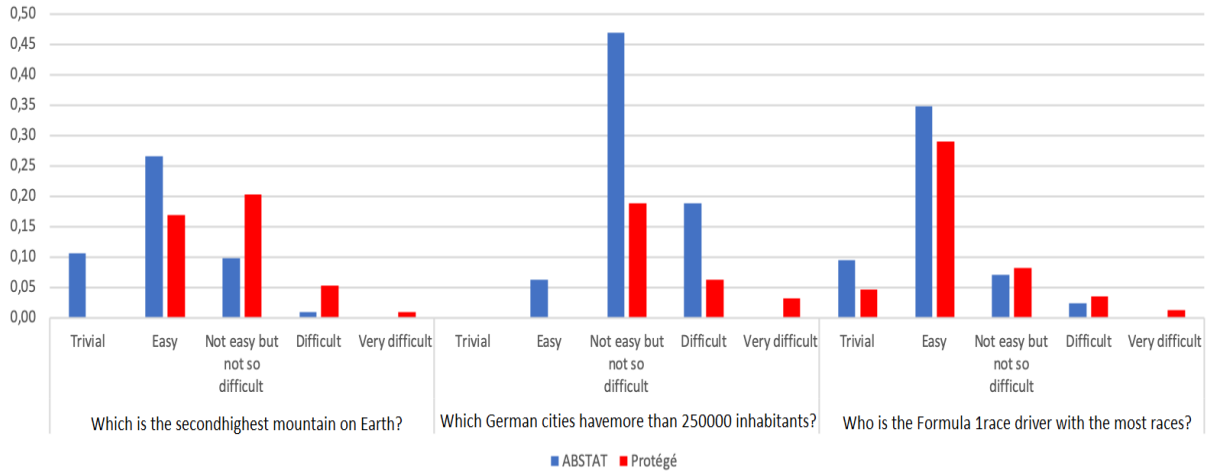

Fig. 7. Perception of the difficulty to answer correctly to each query from both groups

Table 4
Attempts to correctly answer to queries by both groups

Attempts to correctly answer queries	Query 1		Query 2		Query 3	
	ABSTAT	Protégé	ABSTAT	Protégé	ABSTAT	Protégé
No answer	0,00%	0,00%	0,00%	0,00%	2,10%	0,00%
I found both variables with the first try	61,10%	22,40%	17,40%	0,00%	59,60%	60,00%
I found the class with the first try while I made more than one attempt for the property	14,80%	46,90%	21,70%	0,00%	8,50%	20,00%
I found the property with the first try while I made more than one attempt for the object	22,20%	16,30%	21,70%	44,40%	19,10%	7,50%
I made several attempts for both variables	1,90%	12,20%	34,80%	55,60%	8,50%	12,50%
Other	0,00%	2,00%	4,30%	0,00%	2,10%	0,00%
Total	52,40%	47,60%	71,90%	28,10%	54,00%	46,00%

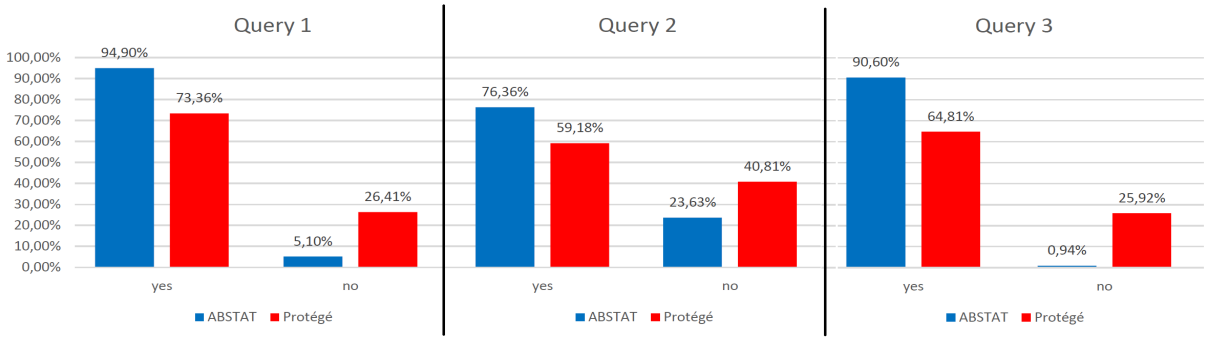

Fig. 8. Distribution of users who used only ABSTAT or Protégé to answer to each query.

Fourth, we analysed the performance of users who watched the introductory video on how ABSTAT works at the beginning of the survey. Table 5 sums up the distribution of ABSTAT users who watched the introductory video before taking the survey and correctly answer to all queries.

We run the Chi-square test to verify if there is a significant statistical evidence between users who

Table 5
Distribution of users who watched or not the introductory video of ABSTAT and correctly answer to the queries.

	Did you watch the video on how to use ABSTAT?	
	Yes	No
Query 1	29	25
Query 2	12	11
Query 3	24	23

watched the introductory video of ABSTAT and those who did not. The Chi-square test showed that there is no statistical difference between users who watched the introductory video before taking the survey and those who did not still answering the queries correctly. This result shows that using ABSTAT is easy and ABSTAT profiles are intuitive to understand and use for query completion tasks even without a preliminary explanation.

4.3. Qualitative Analysis

The qualitative analysis seeks to explore the phenomena, thus in this section we answer to questions such as: *Why did ABSTAT users use only ABSTAT to answer the queries and why Protégé users also need other support? Why exploring the ontology in Protégé to answer to the queries is not enough?*

The two most used strategies to answer the queries by participants that used Protégé were: to directly access the public web page describing the DBpedia named individuals mentioned in the query and to submit exploratory SPARQL queries to the endpoint (Table 6). Many users searched on Google for some entity in the query, then consulted DBpedia web pages to find the correct answer. DBpedia is arguably the best searchable data set, which is the reason why this exploratory approach was successful for relatively simple queries (query number 3). However, this exploratory approach does not work with other non-indexed data sets (e.g., LinkedBrainz) and for complex queries (query 1 and 2). Instead, participants that use ABSTAT took advantage of the profile, obtaining huge benefits in terms of average completion time, accuracy, or both, for all queries.

Another interesting result is given from the analysis of users who did not answer correctly to the queries. We profiled users in order to know, *what*, *how* and *why* did they fail to give the right answer. In order to answer to the *what* question we profiled users in base of their errors. For all queries, ABSTAT users have failed more to find the right classes rather than the right properties. However, this difference is more evident for difficult queries such as query 2, rather than simpler ones. In order to answer to the *how* question, we ask users after completing each query, to describe the steps they took to answer it. For the first query, users who failed to answer correctly used wrong properties such as `dbo:highestMountain` or `dbo:height` instead of `dbo:elevation`. However, even though the users did not find the class `dbo:Mountain` in ABSTAT patterns for the chosen wrong properties (there are no patterns Mountain highestMountain or Mountain height), users were "sure" that `dbo:Mountain` was the right class. Obviously, with the above configurations for the class and property, users could not retrieve any results. For Protégé group, users reported that was easier for them to find the right class but it was harder to understand the right property as from the ontology is impossible to understand how such classes are used. Answering to the *why* question, users failed to find the correct classes and properties because, they did not explore the patterns that describe the relationship between classes, but rather translate the question in natural language "as-is" into SPARQL query. Such problem is more evident to nonnative English speakers. A native speaker will, to a significant degree, ignore the syntax of the question and focus on the intention and the meaning it represents. For the second query, errors could be categorised in two types: logical and semantic errors. Logical errors are those errors caused by the use of wrong syntax in executing the query. Semantic errors instead, are those errors which do not cause query failure, as they are semantically correct predicates and classes, but that are not correct with respect to what it is asked by the query. For example a logical error is when users used classes instead of properties and vice versa (`dbo:City` as predicate). As an example for semantic errors, users used semantically similar properties and classes with respect to the correct answer (`dbo:Location`

Table 6
Usage of different methods to answer to the queries from both groups.

	Which is the second highest mountain on Earth?		Which German cities have more than 250000 inhabitants?		Who is the Formula 1 race driver with the most races?	
	ABSTAT	Protege	ABSTAT	Protege	ABSTAT	Protege
DBpedia Ontology (not in Protege)	1	4	4	5	1	4
DBpedia web page describing the resource in the query	2	7	9	12	3	7
A research in Google	0	3	0	3	0	3
Other	0	0	0	0	1	0
Total	3	14	13	20	5	14

instead of `dbo:City`). Table 7 shows the mistakes done from both groups for Query 2. Most of the users from both groups failed to give the right class/es. Still, most of them were semantic errors. Users mostly choose a similar concept to the right one for answering the query, for example `dbo:Place` or `dbo:PopulatedPlace`. Both groups could find the correct answer for one class (e.g., `dbo:City`) but failed to find the second one (`dbo:Town`). In the hierarchy of concepts, *City* and *Town* are siblings of *Settlement*. However, finding the second concept for such query, is a problem that is more observable for Protégé users even though they could visualise the hierarchy of concepts of DBpedia ontology in WebProtégé. Users could take advantage of such opportunity in order to find the right concepts (the union between two siblings), but most of them chose a more generic concept or even distinct concepts. While ABSTAT users, even though could not visualise the hierarchy of concepts, could find the right answer because in the profile of DBpedia, users could see two patterns, one for <`dbo:City dbo:populationTotal xmls:nonNegativeInteger` > and <`dbo:Town dbo:populationTotal xmls:nonNegativeInteger`> that use populationTotal as predicate. Of course there are also other patterns, but because of their similar meaning ABSTAT users choose *City* and *Town* as concepts to complete and answer to Query 2. Protégé users did not have the possibility to explore the relations between concepts. In fact, when Protégé users where asked to explain the steps they took to answer the query, they made several exploratory queries such as first they found a German city, e.g., Berlin and explored the relative predicates. Many users have reported that initially they chose the predicate `dbo:population`, but after exploring all the predicates of the entities of type *City*, they could change to the correct predicate `dbo:populationTotal`. A lot of users reported that they choose a superclass to *City* such as *PopulatedPlace* or even Place to complete the query as it was very difficult for them to find the right answer. All users tried to explore the query by first accessing the webpage in DBpedia of a German city so they could explore its predicates or relative concepts. Moreover, there were users who searched such information in Google. Such approach can work only with indexed data sets such as DBpedia. For the same query, ABSTAT users who failed to give the correct answer said that they chose *Settlement* but they could use also *Town*. Some said that they were searching for specific entities in ABSTAT' profiles such as for example "...I tried to put res:Germany in the 'object type (occurrences)' section, but I did not get anything..." or "used the full text search to search for German somewhere, and to verify that there are no instances related to cities in German..". However, also several ABSTAT users failed to find the right predicate (`dbo:populationTotal`) as they did not use exploratory queries. Despite the errors, still ABSTAT users achieve higher accuracy and better query completion time.

Table 7
Errors from two groups for Query 2.

	ABSTAT	Protégé	Total
Wrong 1 class	25	34	59
Wrong 2 classes	0	3	3
Wrong 1 predicate	1	0	1
Wrong 2 predicates	0	1	1
Wrong 1 class and 1 predicate	5	0	5
Wrong 1 class and 2 predicates	0	1	1
Wrong 2 classes and 2 predicates	1	2	3
Total	32	41	73

5. Related Work

In this section we first discuss the related work on supporting data set understanding (Section 5.1) and second, we discuss approaches that support Knowledge Graph Profiling (Section 5.2). Although the number of works on data understanding is high (as it will be described in the following), we have not found an agreed data protocol to data understanding.

5.1. Understanding Unknown Data

Several approaches, categorised as in the following, have been proposed to support users understand the content of a data set.

Data Visualization is an effective way of gaining insights into data [12, 25] as it improves interpretability and understanding of the data at hand, facilitating exploration. There exist two paradigms for sensemaking of the data by visualization: global views (top-down) and local views (bottom-up). Approaches based on the global views follow Shneiderman's mantra "overview, zoom & filter, details-on-demand" pattern [55], and provide to the users a big picture of the data, and let them focus on a particular area of the data as she/he zooms and filters [8]. However, such approaches have several challenges when applied to graphs with millions or billions of nodes and edges [35]: (i) such graph overviews are time-consuming to generate, and (ii) there are no perfect overviews. Approaches based on local views aim to support users to identify the nodes to explore at a low level of abstraction, followed by the generation of an overall visualization [43]. The challenges of such approaches are: (i) difficulty to identify the node from which to start the exploration [3], (ii) difficulty to decide which path to follow because nodes have many neighbours [61], and (iii) no users support to write SPARQL queries and understand the relations among types in the graph.

Exploring the data by using *faceted search* is an effective way as they are intuitive to understand and use [24, 45]. However, although facets are very useful to reduce the search space, a large number of facets and facet values might become misleading for users with choice overload and visual clutter. There exist some user studies on how people use facets; e.g., what components of the interface searchers looked at, for how long, and in what order, etc., and they also provide qualitative summaries of users' comments on how facets help their searches [27, 45].

Another direction towards data understanding is to automatically construct SPARQL queries based on user-provided examples known as *query-by-example* approaches [20, 39]. To understand the data users need to interactively execute many queries using different predicates aiming to balance the trade-off between collecting all relevant information and reducing the size of returned data [20]. However, such approaches (i) focus on simple inputs; e.g., a single entity or a pair of entities, but with the increase of the data available in KGs, one or two entities are not satisfactory anymore, (ii) might have different ways on how to expand a given query, thus not all expansions may be of interest to the user, and a large number of expansions may overload the user, (iii) allow to query the graph locally, and thus, do not provide a general overview of the relations among types used in the graph.

Ontologies are considered to be the pillar of the Semantic Web and help users to understand the data [21]. The user, through the support of a tool such as Protégé (implements an "overview first, details-on-demand" approach), accesses first the entire taxonomy, which s/he might expand, and then explore properties, check their domain and range, etc. Understanding the data by looking only at the ontology is complicated; (i) sometimes they do not even exist, (ii) sometimes they are unspecified, and (iii) it may happen that data uses multiple ontologies.

Profiles generated by ABSTAT implement some of the above features. ABSTAT profiles support users in understanding better the data by: (i) allowing the visualisation and exploration of patterns, (ii) adopting faceted search over the patterns, and (iii) enabling full-text search over a single summary or over several summaries.

5.2. Knowledge Graph Profiling

There exist several works on data profiling as it is reviewed in this section and in another recent work [4]. However, to the best of our knowledge, none of the existing profiling approaches have evaluated their ability to produce concise and meaningful profiles from a user perspective. Therefore, in the following we review related work only considering the algorithm of the profiling approach.

RDF graph profiling has been intensively studied, and various approaches and techniques have been proposed to provide a concise and meaningful representation of an RDF KGs. Inspired by the work in [10], we use the same classification of the related works on

KGs profilings. However, the works discussed in each group are different as we focus not only on the summarization approaches but also on the profiling ones.

- *Structural summarization approaches:* These approaches aim to create a summary to support data understanding and visualisation of compex graphs: (i) by considering a set of rules that extract subtypes and subproperties to represent many nodes and edges, or (ii) by extracting clusters to group a set of similar concepts and properties. A formal model for a data graph summary that considers vocabulary usage to assist users in formulating queries across multiple data sources is proposed in [9]. The authors define the concept of *node collections* which is a set of nodes sharing similar characteristics and develop an *Assisted SPARQL* , which is an application that leverages the data graph summary to help users into effectively formulating complex SPARQL queries. S+EPPS is a system that support users by providing summaries based on bisimilarity notion and SPARQL engines to support summary-based exploration [13]. Summaries are constructed by blocks where each block represents a non-overlapping subset of the original data set. ExpLOD is used to summarize a data set based on a mechanism that combines text labels and bisimulation contractions [31]. SchemEX is a stream-based approach used to provide a summary of triples that can be found in a data source [32]. Given a SPARQL query, SchemEX performs a lookup in the index structure to find which data sets contain instances of a specific RDF schema concept that can contribute to answering the query. Our work differs from these works since we introduce concise profiles thanks to the minimalization technique and do not cluster nodes with similar characteristics. Moreover, ABSTAT does not have an interface to help users formulate SPARQL queries but instead patterns are used to support users write SPARQL queries into the KG endpoint.

- *Pattern mining methods*: are used to extract patterns from the RDF graph that "best" represent the input graph. The algorithm of approximate graph pattern matching [57] produces summaries that are capable of summarizing entities in terms of their neighborhood similarity up to a certain distance and a specified bound to indicate the maximum number of the desired patterns. The summaries/patterns are chosen to satisfy and maximize *informativeness* (which should capture the total amount of information; entities and their relationships in a knowledge graph) and *diversity* (which should cover diverse concepts with informative summaries). A scalable lossless compression approach for RDF data sets is presented in [29]. Such an approach uses automatic generation of decompression rules and removes triples that can be inferred from these rules. In [41] authors propose a summarization technique called k-SNAP that integrates an interactive querying scheme by allowing users to customize the summaries based on user-selected node attributes and relationships. Such method allows users to select node attributes and relationships that are of interest and to fix a priori the size of the graph. An approach that includes in the summary a weighted graph composed of supernodes connected by superedges as a result of the partitioning of the original set of vertices in the graph is proposed in [52]. The superedge weights are the edge densities between vertices in the corresponding supernodes. A reconstruction error is proposed to introduce the error for the dissimilarity between the original graph and the summary.

Differently from the above approaches, ABSTAT does not partition the data set based on rules, but instead processes the whole data set and only the final summary is presented to the user. Further, ABSTAT uses a different approach to summarize data and does not consider edge densities. Finally, the ABSTAT profiles provide a onetime profile and does not generate personalized profiles according to a user query.

- *Statistical methods*: In this class are classified all approaches that aim to produce summaries that quantitatively represent the content of the RDF graph. The quantitative information might be used by users to decide if the data set is useful for them or not. LODSight [22] is a web-based tool that displays a summary of classes, datatypes and predicates used in the data set. The visualization allows to quickly and easily find out what kind of data the data set contains and its structure. It also shows how vocabularies are used in the data set. Another tool that tackles data exploration based aggregation is SPADE [19]. Spade uses OLAP-style aggregation to provide users meaningful content of an RDF graph. It chooses aggregates that are visually interesting, a property formally based on statistic properties of the aggregation query results. Users may refine a given aggregate, by selecting and exploring its subclasses. The aggregation is centered around a set of facts, which are nodes of the RDF graph. RDFStats generates statistics for data sets behind SPARQL endpoint and RDF documents [36]. These statistics include the number of anonymous subjects and different types of histograms; URIHistogram for URI subject and histograms for each property and the associated range(s). It also uses methods to fetch the total number

of instances for a given class, or a set of classes and methods to obtain the URIs of instances. LODStats is a profiling tool which can be used to obtain 32 different statistical criteria for RDF data sets [6]. These statistics describe the data set and its schema and include statistics about number of triples, triples with blank nodes, labeled subjects, number of owl:sameAs links, class and property usage, class hierarchy depth, cardinalities etc. These statistics are then represented using Vocabulary of Interlinked Datasets (VOID) and Data Cube Vocabulary[29]. LODOP is a framework for executing, optimizing and benchmarking profiling tasks in Linked Data [23]. ProLOD++ is a web browser tool that implements several algorithms with the aim to compute different profiling, mining or cleansing tasks [2]. ProLOD++ can also identify predicates combinations that contain only unique values as key candidates to distinctly identify entities.

Differently from the above approaches, ABSTAT provides not only statistics on the use of the vocabulary (classes and properties), but also represents KGs content by representing patterns and their respective frequency. ABSTAT does not use aggregation methods for different summary resolution, but instead, it produces only one summary by making use of a terminology graph to extract only patterns that describe relationships between instances of the most specific types.

-*Hybrid methods*: Most of the approaches developed so far combine methods from the structural, statistical and pattern-mining classes in order to provide meaningful summaries. The approach most similar to ABSTAT is Loupe [44]. Loupe extracts types, properties and namespaces, along with a rich set of statistics about their use within the data set. It offers a triple inspection functionality, which provides information about triple patterns that appear in the data set and their frequency. Triple patterns have the form < subjectType, property, objectType >. RDF graphs might be more comprehensible by reducing their size as proposed by [5]. Size reduction is a result of bisimulation and agglomerative clustering (one of the most common types of hierarchical clustering) which discovers subgraphs that are similar with respect to their structure. The semi-structured data summarization approach proposed in [11] is query-oriented. The summary enables static analysis and helps formulating and optimizing queries. The scope of such summaries is to reflect whether the query has some answers against this graph, or finding a simpler way to formulate the query. In summary, information that can be easily inferred is excluded. Such approach has a very high computational complexity.

Differently from the above approaches, ABSTAT does not use clustering but instead reduces the number of patterns based on the minimalisation technique producing profiles that are more concise. In this way, ABSTAT excludes from the summary patterns that can be easily inferred by the subtype graph.

6. Limitations and Lessons Learned

In this paper, we have presented a set of experiments to evaluate a data profiling tool from a user experience perspective in data understanding. Next, we summarize the issues we encountered during the realisation of the experiments, the lessons learned, and some ideas for the future work.

Usefulness of ABSTAT. We received very positive feedback from the participants of the user study with respect to the necessity of having an up and running profiling tool. The appreciation regarded the usefulness of ABSTAT in finding right objects and predicates and especially for the autocompletion suggestions that users founded particularly helpful. Moreover, users stated that it was easier to learn the tool while using.

ABSTAT (tool) limitations. Feedback from the user experiment enlighten us in different directions for improving our tool. First, users reported that even though statistics about the usage of the types and predicates are very useful, the way it is presented is a bit difficult to be quickly processed by humans. Thus, ABSTAT profiles would be more easy and readable if the statistics reported for each pattern would be represented by e.g., a graphical pie chart. Second, that was mostly suggested by the participants is the inclusion of labels or other human-readable description for classes and predicates. Third, users suggested to include in the profile a list of synonyms used to describe a concept or a property. ABSTAT improvement could benefit a lot from such suggestions, thus they are considered for future work.

ABSTAT (approach) limitations. However the usefulness of ABSTAT, it has some limitations. First, more compact profiles might be generated. For example, by considering equivalent classes or properties, the number of patterns in the profile would be lower, thus the number of patterns that user have to explore

[29] http://www.w3.org/TR/vocab-data-cube/

would be lower. At the moment of writing this paper, ABSTAT does not consider equivalence. Second, being a schema-pattern profiling tool, ABSTAT profiles do not include information about entities. Thus, ABSTAT supports users only on writing queries regarding the structure of the KG. Often, users need to have "templates" of how entities are modelled so that they can start their exploration. Third, a limitation that was highlighted also from our first experiment lies in representing concepts that are used as entities (e.g., Surfing, CEO, etc. [58, 59]). Such issues might be solved by applying state-of-the-art approaches for type inference on RDF data, or by including in the profiles, values for concepts that are defined by closed and relatively small instance sets.

User study limitations. The user study highlighted the valuable support provided by ABSTAT in writing SPARQL queries, but has also one limitation. When designing the experimental setup, the assignment of the tool used as a support for completing the query was left to the user choice. For this reason, the distribution of the self-reported SPARQL-related competency was not equally distributed between the two groups. More precisely, ABSTAT group included two self-reported expert users, while no self-reported expert used Protégé. However, one of the experts could only answer to first query and quit the survey while the other expert completed the survey and answered correctly to the first and third query while giving the wrong answer to the second query. In order to verify if the statistical significance was influenced by these two experts, we removed them from the analysis and run again the Chi-Square test. Even removing the expert users from the evaluation, the significance is maintained, meaning that all non expert users took advantage of using ABSTAT in completing the queries faster and more accurately.

7. Conclusions

Understanding big knowledge graphs can be a complex and challenging task. In this paper, we present a method to evaluate data profiling tools from a user perspective related to data understanding. Based on the experimentation we show that our profiling framework is able to provide both *concise* and *informative* profiles for a given data set. Ontology-based ABSTAT profiles are more **concise** than similar pattern-based profiles that do not apply ABSTAT's minimalization mechanism. ABSTAT profiles are also **informative** as they help users understand the structure of complex KGs like DBpedia.

The latter feature has been measured in a user study where 113 participants had the task to complete a set of queries in SPARQL using ABSTAT profiles and Web Protégé. The analysis showed that all users, independently of their proficiency in Semantic Web technologies could gain advantage of ABSTAT profiles: on average, they completed the queries more accurately and in less time than users that used Web Protégé as baseline method for exploring the structure of the KG. Statistical evidence suggest that ABSTAT is significantly more helpful for queries of medium-high difficulty, and as much helpful for simple queries. An interesting phenomenon emerged in our study concerns the users' perception on the difficulty of the queries: users that performed the task using ABSTAT seem to have perceived queries as less difficult. Moreover, ABSTAT users had to make fewer attempts to submit their queries, and resorted much less frequently to alternative methods, such as trying exploratory queries over search engines and explore pages of DBpedia entities, used by many users from the Web Protégé group. It is particularly remarkable because these alternative methods are not available for most of the KG available on the web, which cannot be reached effectively via web searches. Otherwise, very few users from the ABSTAT group accessed the DBpedia web page and none of the user made a complementary web search. Finally, ABSTAT interface is very easy to be used even for users who are not trained before. The statistical test showed that there is no statistical difference between users who are trained before performing the task and the ones who use ABSTAT for the first time while performing the task.

We plan to extend ABSTAT profiles with other relevant statistics that might help user in understanding better the data such as the general information about the usage of classes and properties (object / datatype) and value distributions for numerical properties. Moreover, we plan to use such statistics to capture also quality errors in the data.

Acknowledgements

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

References

[1] Ziawasch Abedjan, Lukasz Golab, Felix Naumann, and Thorsten Papenbrock. Data profiling. *Synthesis Lectures on Data Management*, 10(4):1–154, 2018.

[2] Ziawasch Abedjan, Toni Grütze, Anja Jentzsch, and Felix Naumann. Profiling and mining rdf data with prolod++. In *2014 IEEE 30th International Conference on Data Engineering*, pages 1198–1201. IEEE, 2014.

[3] Leman Akoglu, Duen Horng Chau, U Kang, Danai Koutra, and Christos Faloutsos. Opavion: Mining and visualization in large graphs. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 717–720, 2012.

[4] Renzo Arturo Alva Principe, Andrea Maurino, Matteo Palmonari, Michele Ciavotta, and Blerina Spahiu. Abstat-hd: a scalable tool for profiling very large knowledge graphs. *The VLDB Journal*, pages 1–26, 2021.

[5] Anas Alzogbi and Georg Lausen. Similar structures inside rdf-graphs. *LDOW*, 996, 2013.

[6] Sören Auer, Jan Demter, Michael Martin, and Jens Lehmann. Lodstats–an extensible framework for high-performance dataset analytics. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 353–362. Springer, 2012.

[7] Nikos Bikakis and Timos Sellis. Exploration and visualization in the web of big linked data: A survey of the state of the art. *arXiv preprint arXiv:1601.08059*, 2016.

[8] Katy Börner, Chaomei Chen, and Kevin W Boyack. Visualizing knowledge domains. *Annual review of information science and technology*, 37(1):179–255, 2003.

[9] Stephane Campinas, Thomas E Perry, Diego Ceccarelli, Renaud Delbru, and Giovanni Tummarello. Introducing rdf graph summary with application to assisted sparql formulation. In *DEXA*, pages 261–266. IEEE, 2012.

[10] Šejla Čebirić, François Goasdoué, Haridimos Kondylakis, Dimitris Kotzinos, Ioana Manolescu, Georgia Troullinou, and Mussab Zneika. Summarizing semantic graphs: a survey. *The VLDB Journal*, 28(3):295–327, 2019.

[11] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. Query-oriented summarization of rdf graphs. *Proceedings of the VLDB Endowment*, 8(12):2012–2015, 2015.

[12] Min Chen, David Ebert, Hans Hagen, Robert S Laramee, Robert Van Liere, Kwan-Liu Ma, William Ribarsky, Gerik Scheuermann, and Deborah Silver. Data, information, and knowledge in visualization. *IEEE computer graphics and applications*, 29(1):12–19, 2008.

[13] Mariano P Consens, Valeria Fionda, Shahan Khatchadourian, and Giuseppe Pirro. S+ epps: construct and explore bisimulation summaries, plus optimize navigational queries; all on existing sparql systems. *Proceedings of the VLDB Endowment*, 8(12):2028–2031, 2015.

[14] Marco Cremaschi, Flavio De Paoli, Anisa Rula, and Blerina Spahiu. A fully automated approach to a complete semantic table interpretation. *FGCS*, 112:478–500, 2020.

[15] Vincenzo Cutrona, Federico Bianchi, Ernesto Jiménez-Ruiz, and Matteo Palmonari. Tough tables: Carefully evaluating entity linking for tabular data. In *International Semantic Web Coborrownference*, pages 328–343. Springer, 2020.

[16] Vincenzo Cutrona, Michele Ciavotta, Flavio De Paoli, and Matteo Palmonari. Asia: a tool for assisted semantic interpretation and annotation of tabular data. In *ISWC Satellites*, pages 209–212, 2019.

[17] Vincenzo Cutrona, Flavio De Paoli, Aljaž Košmerlj, Nikolay Nikolov, Matteo Palmonari, Fernando Perales, and Dumitru Roman. Semantically-enabled optimization of digital marketing campaigns. In *International Semantic Web Conference*, pages 345–362. Springer, 2019.

[18] Tommaso di Noia, Andrea Maurino, Corrado Magarelli, Matteo Palmonari, and Anisa Rula. Using ontology-based data summarization to develop semantics-aware recommender systems. In *The Semantic Web - ESWC 2018*, 2018.

[19] Yanlei Diao, Pawel Guzewicz, Ioana Manolescu, and Mirjana Mazuran. Spade: A modular framework for analytical exploration of RDF graphs. *Proc. VLDB Endow.*, 12(12):1926–1929, 2019.

[20] Kyriaki Dimitriadou, Olga Papaemmanouil, and Yanlei Diao. Explore-by-example: An automatic query steering framework for interactive data exploration. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 517–528, 2014.

[21] John Domingue, Dieter Fensel, and James A Hendler. *Handbook of semantic web technologies*. Springer Science & Business Media, 2011.

[22] Marek Dudáš, Vojtěch Svátek, and Jindřich Mynarz. Dataset summary visualization with lodsight. In *ESWC*, pages 36–40. Springer, 2015.

[23] Benedikt Forchhammer, Anja Jentzsch, and Felix Naumann. Lodop-multi-query optimization for linked data profiling queries. In *PROFILES@ ESWC*, 2014.

[24] Luis Fuenmayor, Diego Collarana, Steffen Lohmann, and Sören Auer. Farbie: A faceted reactive browsing interface for multi rdf knowledge graph exploration. In *VOILA@ ISWC*, pages 111–122, 2017.

[25] Juan Gómez-Romero, Miguel Molina-Solana, Axel Oehmichen, and Yike Guo. Visualizing large knowledge graphs: A performance analysis. *FGCS*, 89:224–238, 2018.

[26] Anna Hart. Mann-whitney test is not just a test of medians: differences in spread can be important. *Bmj*, 323(7309):391–393, 2001.

[27] Philipp Heim and Jürgen Ziegler. Faceted visual exploration of semantic data. In *Workshop on Human-Computer Interaction and Visualization*, pages 58–75. Springer, 2009.

[28] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4):71:1–71:37, 2021.

[29] Amit Krishna Joshi, Pascal Hitzler, and Guozhu Dong. Logical linked data compression. In *ESWC*, pages 170–184. Springer, 2013.

[30] Akrivi Katifori, Constantin Halatsis, George Lepouras, Costas Vassilakis, and Eugenia G. Giannopoulou. Ontology visualization methods - a survey. *ACM Comput. Surv.*, 39(4):10, 2007.

[31] Shahan Khatchadourian and Mariano P Consens. Explod: summary-based exploration of interlinking and rdf usage in the linked open data cloud. In *Extended Semantic Web Conference*, pages 272–287. Springer, 2010.

[32] Mathias Konrath, Thomas Gottron, Steffen Staab, and Ansgar Scherp. Schemex-efficient construction of a data catalogue by stream-based indexing of linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 16:52–58, 2012.

[33] Danai Koutra, U Kang, Jilles Vreeken, and Christos Faloutsos. Summarizing and understanding large graphs. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8(3):183–202, 2015.

[34] Samit Kumar and Vikram Singh. Understanding data exploration search a brief study of user's exploratory search facets. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–6, 2018.

[35] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics*, 22(3):277–286, 2003.

[36] Andreas Langegger and Wolfram Woss. Rdfstats-an extensible rdf statistics generator and library. In *2009 20th International Workshop on Database and Expert Systems Application*, pages 79–83. IEEE, 2009.

[37] Huiying Li. Data profiling for semantic web data. In *International Conference on Web Information Systems and Mining*, pages 472–479. Springer, 2012.

[38] Matteo Lissandrini, Davide Mottin, Themis Palpanas, and Yannis Velegrakis. Multi-example search in rich information graphs. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 809–820. IEEE, 2018.

[39] Matteo Lissandrini, Davide Mottin, Themis Palpanas, and Yannis Velegrakis. Graph-query suggestions for knowledge graph exploration. In *Proceedings of The Web Conference 2020*, pages 2549–2555, 2020.

[40] Vanessa Lopez, Christina Unger, Philipp Cimiano, and Enrico Motta. Evaluating question answering over linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 21:3–13, 2013.

[41] Amine Louati, Marie-Aude Aufaure, Yves Lechevallier, and France Chatenay-Malabry. Graph aggregation: Application to social networks. In *HDSDA*, pages 157–177, 2011.

[42] Nicolas Marie and Fabien Gandon. Survey of linked data based exploration systems. In *IESD 2014 at (ISWC 2014)*, volume 1279 of *CEUR Workshop Proceedings*, 2014.

[43] Gonzalo Gabriel Méndez, Uta Hinrichs, and Miguel A Nacenta. Bottom-up vs. top-down: trade-offs in efficiency, understanding, freedom and creativity with infovis tools. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 841–852, 2017.

[44] Nandana Mihindukulasooriya, María Poveda-Villalón, Raúl García-Castro, and Asunción Gómez-Pérez. Loupe-an online tool for inspecting datasets in the linked data cloud. In *International Semantic Web Conference (Posters & Demos)*, 2015.

[45] Xi Niu, Xiangyu Fan, and Tao Zhang. Understanding faceted search from data science and human factor perspectives. *ACM Transactions on Information Systems*, 37(2):1–27, 2019.

[46] Natalya F Noy, Michael Sintek, Stefan Decker, Monica Crubézy, Ray W Fergerson, and Mark A Musen. Creating semantic web contents with protege-2000. *IEEE intelligent systems*, 16(2):60–71, 2001.

[47] Heiko Paulheim. Towards profiling knowledge graphs. In *PROFILES@ ISWC*, 2017.

[48] Adam Perer and Ben Shneiderman. Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 265–274. ACM, 2008.

[49] Igor O Popov, MC Schraefel, Wendy Hall, and Nigel Shadbolt. Connecting the dots: a multi-pivot approach to data exploration. In *International semantic web conference*, pages 553–568. Springer, 2011.

[50] Renzo Arturo Alva Principe, Blerina Spahiu, Matteo Palmonari, Anisa Rula, Flavio De Paoli, and Andrea Maurino. Abstat 1.0: Compute, manage and share semantic profiles of rdf knowledge graphs. In *European Semantic Web Conference*, pages 170–175. Springer, 2018.

[51] Azzurra Ragone, Paolo Tomeo, Corrado Magarelli, Tommaso Di Noia, Matteo Palmonari, Andrea Maurino, and Eugenio Di Sciascio. Schema-summarization in linked-data-based feature selection for recommender systems. In *Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017*, pages 330–335, 2017.

[52] Matteo Riondato, David García-Soriano, and Francesco Bonchi. Graph summarization with quality guarantees. *Data mining and knowledge discovery*, 31(2):314–349, 2017.

[53] Md. Kamruzzaman Sarker, Adila Krisnadhi, David Carral, and Pascal Hitzler. Rule-based OWL modeling with rowltab protégé plugin. In Eva Blomqvist, Diana Maynard, Aldo Gangemi, Rinke Hoekstra, Pascal Hitzler, and Olaf Hartig, editors, *The Semantic Web - 14th International Conference, ESWC 2017, Portorož,*, volume 10249 of *Lecture Notes in Computer Science*, pages 419–433, 2017.

[54] Jeff Sauro and James R Lewis. *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann, 2016.

[55] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE symposium on visual languages*, pages 336–343. IEEE, 1996.

[56] Ben Shneiderman and Catherine Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI workshop*, pages 1–7, 2006.

[57] Qi Song, Yinghui Wu, Peng Lin, Luna Xin Dong, and Hui Sun. Mining summaries for knowledge graph search. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1887–1900, 2018.

[58] Blerina Spahiu, Andrea Maurino, and Matteo Palmonari. Towards improving the quality of knowledge graphs with data-driven ontology patterns and shacl. In *ISWC (Best Workshop Papers)*, pages 103–117, 2018.

[59] Blerina Spahiu, Riccardo Porrini, Matteo Palmonari, Anisa Rula, and Andrea Maurino. Abstat: ontology-driven linked data summaries with pattern minimalization. In *European Semantic Web Conference*, pages 381–395. Springer, 2016.

[60] York Sure, Juergen Angele, and Steffen Staab. Ontoedit: Multifaceted inferencing for ontology engineering. In *Journal on Data Semantics I*, pages 128–152. Springer, 2003.

[61] Jian Zhao, Christopher Collins, Fanny Chevalier, and Ravin Balakrishnan. Interactive exploration of implicit and explicit relations in faceted datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2080–2089, 2013.