International Journal of Transdisciplinary AI
Vol. 1, No. 1 (2019) 19-43

DOI: 10.35708/TAI1868-126244

# Enabling the Continuous Evolution of Ontologies for Ontology-Based Data Management

André Pomp[1], Johannes Lipp[1], and Tobias Meisen[2]

[1] Institute of Information Management in Mechanical Engineering, RWTH Aachen University, Dennewartstr. 27, 52068 Aachen, Germany
{andre.pomp,johannes.lipp}@ima.rwth-aachen.de
[2] Chair of Technologies and Management of Digital Transformation, University of Wuppertal, Rainer-Gruenter-Str. 21, 42119 Wuppertal, Germany
meisen@uni-wuppertal.de

Received (07/15/2019)
Revised (08/22/2019)
Accepted (09/30/2019)

**Abstract.** Companies suffer from heterogeneous data sources that are distributed across all business units and locations. Accessing, discovering, and understanding these data sources is challenging because data scientists have to deal with different protocols, data formats, and even company-specific organizational issues such as firewalls and privacy policies. Ontology-Based Data Management (OBDM) provides different aspects for reducing the barriers of integrating, accessing and managing heterogeneous data sources by using ontologies. For that, we establish a mapping between data sources and one or multiple target ontologies. However, the biggest challenges in ODBM are creation and administration of required ontologies. Usually, these ontologies are created in advance, for instance, by ontology engineers and domain experts that closely work together for manually designing and even maintaining the ontology.

In order to enhance the process of designing and maintaining an ontology, we propose an approach consisting of an evolving knowledge graph that includes an internal ontology, which continuously evolves on demand as domain experts add new data sources and define the mapping between the ontology and that data source. For this purpose, we develop an intuitive, user-oriented wizard and combine it with a semi-supervised evolution strategy that supports the user with the help of external knowledge databases. Moreover, we equip the system with additional logic that allows to automatically link related concepts of the ontology. We evaluate the accuracy and usability of our approach by conducting a user

study. The results show that mappings become more objective and consistent with our provided user wizard, resulting in a knowledge graph with higher connectivity and stability.

**Keywords:** Ontology-Based Data Management; Knowledge Graph; ESKAPE; Semantic Modeling

## 1 Introduction

The latest trends in smart manufacturing result in an enormous amount of data from a variety of available data sources, mainly stored in isolated silos. Approaches such as data catalogs and data warehouses are useful for indexing or centralizing all these sources, but have several drawbacks. For example, data catalogs only allow data to be indexed, so that data scientists still have to deal with the different formats and protocols. Data warehouses, on the other hand, are not suitable for collecting large amounts of streaming data arriving at high frequencies. In addition, they are mainly designed for the use with structured and semi-structured data but not with unstructured data.

Another option that is often used today to centralize data storage are data lakes. Data lake architectures first manage the complexity associated with high data volumes by enabling raw data collection for structured, semi-structured, and unstructured data. However, current data lake architectures avoid data semantics or homogenization because they do not take into account that as the number of data sources increases, they become less transparent, discoverable, and understandable. Data scientists, who are responsible for analyzing the collected data, are therefore no longer able to discover and understand the data sources that can contribute to their analysis. One reason for this is that data scientists do not have the information available to understand the semantics of a data source. For instance, a data label that is used within a data source might be an abbreviation (e.g., *temp*) or the semantics of the label may not be clear (e.g., a data attribute labeled with temperature does not declare the unit used for measuring). In addition, understanding the relationship between multiple data sources poses another challenge.

In order to overcome these limitations and to deal with heterogeneous data sources, Ontology-Based Data Management (OBDM) addresses these challenges by clarifying the semantics and relationships of data sources via a mapping between a data source and one or more target ontologies [25]. The areas of OBDM include Ontology-Based Data Integration (OBDI), Ontology-Based Data Access (OBDA), Ontology-Based Data Governance (OBDG) and many others (cf. [26]). The ontology chosen for this purpose is used as an explicit, formal specification for conceptualization, i.e., the concept formation of a domain [11]. While most of the research in OBDI and OBDA focuses on enabling the integration and access of relational databases (cf. [4,43]), more recent approaches also apply these paradigms for non-relational data stores, like it is used in most data lake architectures (cf. [13]). In fact, even the use for accessing streaming data (cf. [21]) is already possible through OBDM approaches. However, one of the major drawbacks

of all the different approaches covered by OBDM is that their functionality and quality always depends on the underlying ontology that was created in advance so that it can be used for establishing a mapping between the data sources and the ontology. However, this ontology creation is either done manually by domain experts and ontology engineers or the ontologies are automatically inferred with approaches coming from the research area of ontology learning. In both cases, the ontology creation is time-consuming and costly, especially when it comes to a company-wide standardization caused by the fact that the created ontology must either be designed or manually reviewed. Nevertheless, the initial ontology creation is only the first step. Modern environments, especially in enterprises, frequently change, which leads to the fact that the ontology must be extended or adapted as well [12]. However, not only the changing environments lead to the necessity of maintaining ontologies. Other reasons might just be that concepts or relations were forgotten during the ontology creation or that the experts had the opinion that those concepts or relations are unimportant [40]. Independent of the reason for the missing concepts or relationships, a data provider, who wants to offer a data source based on OBDM, runs into the problem that the ontology does not match their needs. In order to counter this lack of information, the ontology is usually adapted manually by an expert on request. The user performing the modeling process has no chance to change the ontology according to their needs, which means that the data source can only be integrated after the extension.

In order to overcome the issues of conventional ontologies used in OBDM, we propose an approach featuring an evolutionary knowledge graph that consists of an internal growing ontology (universal knowledge) and data source specific mappings (local knowledge) (cf. knowledge graph definition given in [36]). These two building blocks together form a domain conceptualization, serve as a data source index and are used to continuously adapt and extend the knowledge graph on-demand. As mappings, we do not rely on commonly used semantic annotations but on more sophisticated semantic models, which allow to express more details about the original data source. Compared to traditional mapping approaches that are used in OBDM and which only allow mapping between a data source and one or more target ontologies, we additionally allow the user, who creates the mapping, to define and use concepts and relationships in their semantic models that were previously unknown to the knowledge graph's ontology. This is, for instance, the case if the concept the user wants to use is missing in the current ontology. This novel knowledge must be integrated into the ontology of the knowledge graph, and it is important to do this in a controlled and validated manner. To achieve this goal, we follow a multi-folded approach. First, we develop an intuitive user-oriented wizard. If a user wants to add a new concept during the creation of the semantic model, the wizard guides the user by suggesting knowledge from both the knowledge graph's ontology and additional external publicly available knowledge bases. Second, to learn additional knowledge from the external knowledge bases, our wizard offers a semi-supervised evolution strategy to gain different kinds of knowledge from external knowledge

bases. Here, we have the hypothesis that the wizard improves the objectiveness and consistency of the created semantic models in those cases where users introduce new concepts and relations. Third, we develop a strategy for the controlled and evolutionary development of the knowledge graph. This strategy evaluates the relations and concepts that a user selects or introduces, and validates if the user introduces contradictions to the current knowledge graph's ontology. In addition, the strategy identifies additional relationships between concepts that improve the density of the underlying ontology. Examples of such improvements are the identification of terms that are used interchangeably, i.e., synonyms.

We implement our approach based on the semantic data platform ESKAPE [35], which already offers Ontology-Based Data Management for batch and streaming data. We extend ESKAPE's knowledge graph model [35] and user interface [37] to fit our requirements. In addition, we add a validation logic to find inconsistencies and we develop a semi-supervised evolution strategy for evolving the knowledge graph based on the external knowledge bases. To evaluate our approach, we conducted a user study with a heterogeneous group of participants, including people with and without experience in semantic modeling. During the study, participants defined semantic models based on predefined and well documented public data sets. Based on the created semantic models, we measure how well our user wizard improves the quality of semantic models, as these models form the basis for a stable, interconnected and consistent knowledge graph. The evaluation shows that the use of our user wizard leads to an increased quality and consistency of semantic models compared to scenarios where users had to create semantic models without our wizard. The semantic models that were created with our user wizard lead to a more stable development of the knowledge graph ontology.

This paper is an extension of our work [38]. In comparison, to our original work, this paper includes the following improvements:

- more details about our architecture and implementation
- an additional algorithm for the continuous evolution of the knowledge graph's ontology
- three extra data sets included in the evaluation
- more evaluation details and results

The remainder of this paper is organized as follows: First, we present a motivating in example in Section 2 and discuss related work in Section 3. Based on the current state of the art, we discuss our approach and its implementation in Section 4. Finally, we present the evaluation results in Section 5, before we conclude in Section 6.

## 2 Motivating Example

In this section, we present a motivating example that illustrates the need for enterprises to ensure that ontologies are not static but dynamic.

The scenario consists of a multinational manufacturing enterprise with multiple sites in different countries that is already centralizing its data in a data

lake. The data that is stored consists of various structured, semi-structured and unstructured internal enterprise data. In addition, the company already applies the pattern of Ontology-Based Data Integration, i.e., it semantically integrates the data into the lake based on a predefined ontology. The ontology itself was initially created by a group of external ontology engineers together with a small group of five experts from the company. The company's data scientists are already able to query the data based on the underlying ontology, which simplifies the data analysis process considerably. The mappings between the data sources and the ontology are created in a semi-automatic fashion, i.e., the Ontology-Based Data Integration platform tries to automatically create a semantic model for each new data source. The user, who added the data source to the platform, i.e., the data provider, then reviews this semantic model and adjusts it according to their needs. The data provider, for instance, corrects possible errors, adds mappings that could not be established or removes mappings that do not fit to this data source.

Two months after the first creation of ontology, one of the older production machines breaks down and the company decides to buy a new, modern machine. The old machine had no native interfaces for data acquisition. The only data collected by this machine came from sensors that were manually installed on the machine. By contrast, the new machine has a modern OPC UA[3] interface and collects significantly more different data attributes. Following the company's strategy of centralizing data storage, a data provider becomes responsible for connecting this machine to the data lake. The data provider connects the OPA-UA interface with the Ontology-Based Data Integration platform and the platform recommends a semantic model. Unfortunately, more than half of the data attributes provided cannot be mapped to a concept of the ontology because they were forgotten during creation. The reason they were forgotten is because none of the experts knew that such values could be measured with a new machine as the company did not own such a machine before. Hence, the data provider has to contact the external ontology engineers so that they extend the ontology according to their needs.

In considering this scenario, we identify several disadvantages of common solutions for the current process of publishing data. First, the ontology is missing concepts because none of the experts knew that these concepts existed or will be important in the future. The reason for this is that the ontology was created in advance and only covers domain knowledge, but typically knowledge evolves over time which implies that the ontology must also change over time. Second, a third party must be involved to extend the ontology, since the company does not employ ontology engineers. However, these ontology engineers are again reliant on the knowledge gained by the data provider. Third, the extension of the ontology requires time as the ontology engineers are not directly available. This implies that no data is integrated into the lake until the extension took place resulting in lost data.

[3] https://opcfoundation.org/about/opc-technologies/opc-ua/

While defining a comprehensive ontology may be possible in a closed and controllable environment that does not change frequently, it is not suitable for a global company with multiple sites in different countries. Here, we especially note that this scenario only considered company-internal data. Today, external data like social media data, weather data etc. are often included in data analytics processes as well, and storing them within the own data lake has multiple advantages, such as guaranteed availability and accessibility. Adapting the ontology for each new external data source again results in a large maintenance effort. Hence, we conclude that we require an approach that supports the continuous evolution of ontologies without the necessity to involve ontology engineers.

## 3 Related Work

For applying Ontology-Based Data Management, we require two essential building blocks. On the one hand, we need a suitable and expressive ontology and, on the other hand, we need an algorithm for establishing a mapping between the ontology and corresponding resources. In the following, we discuss recent approaches in the area of semantic mapping as well as ontology creation/maintenance. Moreover, we consider the research direction of semantic tagging as it is also close to our research.

The problem of creating suitable ontologies that can be used for mapping attributes (e.g., column labels of a table) of structured data sources to concepts, and thus add a meaning to each of those, is a well known problem [10]. However, creating these ontologies is very time consuming [24]. Hence, the automatic creation and maintenance of ontologies is an important research topic. In the area of ontology learning, various approaches exist that deal with the automatic creation of ontologies. He et al. present in [14] an ontology learning method from newspaper texts whereas Zhang et al. [46] as well as Kotis et al. [22] use query logs for creating initial ontologies. Instead of using unstructured text documents, Jiménez-Ruiz et al. [18] or Abbes et al. [1] use database schemes from relational as well as non-relational databases in order to infer a first ontology. While these approaches only create initial ontologies, other approaches deal with the evolution of ontologies. For instance, Braun et al. [3] evolves ontologies based on SPARQL queries whereas Hu et al. [16] rely on external knowledge databases. However, none of the evolution approaches deals with evolving ontologies in real-time. After a certain time span, they publish a new version of the ontology. Hence, if a concept is missing during the mapping process, it cannot be used.

The creation of suitable knowledge bases is not only issue for ontologies but also for other types of knowledge bases, such as semantic networks or knowledge graphs. DBpedia [2] as well as Milne et al. [29, 30] extract content from Wikipedia on a large scale and make it available in structured (linked) form, while YAGO [27], UNIpedia [19, 20] and BabelNet [31] add knowledge from other sources as well. Paulheim examines in [33] methods for refining knowledge graphs and suggests in [28] how to recognize relationship assertion errors in knowledge graphs.

However, these knowledge bases, such as the ontologies in OBDM, are important for establishing the mapping. Defining comprehensive semantic models is

only possible if suitable knowledge bases are available. Knoblock et al. [44] introduce an approach to automatically define a semantic model for a data set based on a predefined domain ontology. Semantic relationships can be automatically derived for attributes that are already associated with concepts from semantic models that were defined for other data sets. Auer et al. [42] propose an approach that automatically merges concepts from multiple isolated structured data sets. Heyvaert et al. [15] improve the generation of semantic mappings by considering Data, Schema, Query and Mapping (DSQM) knowledge in combination with multiple target ontologies. However, all the presented semantic modeling approaches still face the problem that the vocabulary used by the researchers is predefined and develops statically or periodically based on automatic content extraction. If knowledge is missing during the semantic model creation, the user has no chance to add new knowledge on demand.

In contrast to semantic modeling, the field of semantic tagging concentrates on tagging unstructured and binary data, such as documents, images, or videos, in order to enable better retrievability [7]. Instead of using a predefined and static knowledge base, semantic tagging approaches usually follow a flexible on-demand strategy. For instance, social networking platforms such as Facebook[4] or Flickr[5] allow users to freely choose tags. The system of Torres et al. [45] allows users to tag web resources with categories, properties and attributes using a Firefox plugin. The tags created by these non-automated semantic tagging techniques are based on the knowledge of each user and therefore contain specific domain knowledge. At the same time, common vocabulary that is constructed over a longer period of time can be used to suggest tags to other users.

While this degree of freedom is excellent for gathering knowledge from many experts, it leads to other challenges due to ambiguous or noisy tags. For example, users searching for documents tagged by others may not know the terms that were used for the tags. Du et al. [7] and Laniado et al. [23] solve these problems with inferior or noisy tags. Their approaches filter out high-value tags or verify them with a predefined vocabulary. To avoid ambiguities and noise, Gil et al. propose in [9] a method that allows users to add metadata that can be immediately adopted by others. Unfortunately, semantic tagging approaches only deal constructing vocabularies consisting of terms. More complex vocabularies cannot be created.

Altogether, we conclude that there already exist different approaches for constructing initial knowledge bases, like ontologies or knowledge graphs and that there also exist approaches for evolving this knowledge. Moreover, there also exist approaches that use the provided knowledge for performing the mapping. However, none of the approaches deals with the problem of evolving the knowledge base, such as the ontology, on-demand during the mapping phase if a concept is missing. However, this is exactly done in the research area of semantic tagging. Hence, our goal is to combine the advantages of ontology learning and evolution, semantic modeling and semantic tagging by developing an approach that com-

[4] https://www.facebook.com
[5] https://www.flickr.com

bines these strategies to benefit from their advantages and eliminate drawbacks at the same time.

## 4 Concept

In this section, we describe our approach that we implement in the semantic data platform ESKAPE [35]. ESKAPE is a semantic data hub that offers Ontology-Based Data Management (OBDM) by allowing data stewards to publish and describe data sources so that other users, such as data scientists, can retrieve these data sets later on. To describe the data sources, the data steward has to create a semantic model for a data set. To create semantic models, ESKAPE provides users with a graphical interface [37]. Based on this UI, users choose concepts and relations from ESKAPE's underlying knowledge graph, which encourages them to build choices upon that shared terminology. Compared to traditional OBDM approaches, ESKAPE's knowledge graph, which Pomp et al. define in [36] and whose implementation details are described in [35], allows users to extend its vocabulary by introducing new concepts or relations on-demand directly to the knowledge graph to make them available for others.

While this previous work is a good first step for managing a knowledge graph and extending it on-demand, its implementation still had limitations. Although ESKAPE already offered users the definition of new concepts, it was not checked whether newly introduced concepts already existed in different forms such as synonyms. In addition, ESKAPE did not check whether newly added relationships bring contradictions with them. Also, users did not get any support when introducing new concepts and relationships.

To overcome these limitations, we extended ESKAPE's current approach as follows. First, we modified ESKAPE's current knowledge graph model to meet our new requirements (cf. Section 4.1). Next, we extended ESKAPE's user interface with a user wizard (cf. Section 4.2) that extends the functionalities presented in [37] by (i) previewing concepts with their valuable context, (ii) avoiding duplicates and ambiguity by guiding users, and (iii) leveraging external sources to provide quality-assurance. This means that if users want to introduce new concepts during the semantic modeling process, the wizard provides them with suggestions coming from ESKAPE's current knowledge graph as well as from external knowledge bases, such as BabelNet. If users want to import knowledge from an external knowledge base, we follow a semi-supervised evolution strategy where the user can decide which additional knowledge from the external knowledge base they want to import. As soon as the user finishes their semantic model, we introduce a validation and evolution step (cf. Section 4.3) which validates the new knowledge that is added with this new semantic model. Therefore, we check for conflicts and contradictions. As soon as the semantic model was validated, it is added to ESKAPE's knowledge graph and can be used for the evolution of the knowledge graph's ontology. In order to identify additional useful relationships between already existing concepts, we add an additional autonomous evolution step that relies on external knowledge databases (cf. Section 4.4).

### 4.1 Modifying ESKAPE's Knowledge Graph Model

In a first step, we modified the existing knowledge graph model of ESKAPE, which was initially presented in [35], to meet the requirements that arise when enriching the graph with additional knowledge from external knowledge bases.

Therefore, we modified the ESKAPE's Entity Concept element (cf. [35]) to additionally contain synonym labels as well as information from which source the concept was obtained. Thus, an Entity Concept now consists of (i) a main label, (ii) any number of synonym labels to clarify its meaning and avoid ambiguity, (iii) a human readable description with usually 1-2 sentences, (iv) the origin such as user-provided or extracted from a particular knowledge base, and (v) connections to other concepts (relations) to provide context and clarify its meaning. For the elements that describe relations in ESKAPE, called Relation Concepts, we introduced a property *origin* for describing the source the relation was obtained from. We additionally added properties that allow us to describe (ir-)reflexive, (a-/anti-)symmetric, and transitive relationships. These properties imply restrictions when applied, and we use them in the validation and evolution step to identify if contradictions where introduced by an added semantic model.

Although we describe how we extended the knowledge graph model of the semantic data platform ESKAPE, we want to note that this method can also be transferred to other Ontology-Based Data Management approaches. Instead of modeling the knowledge graph based on the ESKAPE implementation, it is also possible to model the knowledge graph based on RDF and OWL. However, we choose ESKAPE as it already is a sophisticated semantic data integration platform with a flexible knowledge graph and an advanced user interface that we can adapt to implement our semi-supervised evolution strategy. This enables us to involve the user more actively in adding additional knowledge to the underlying knowledge graph.

### 4.2 User Wizard

To support and improve a precise knowledge graph evolution, we propose a user wizard that guides the user through the selection process for semantic concepts matching the intended meaning. The user wizard helps users to find the best matching concept for their semantic model so that semantic models among different users become objective and consistent.

The workflow of our proposed user wizard is depicted in Figure 1 and starts with the user entering a search term. The user wizard first tries to find a matching concept in the ontology of ESKAPE's knowledge graph. For instance, if the data set contains the data label *hotspot* and the knowledge graph already knows the concept *access point*, then the wizard would suggest *access point* as it serves as a synonym for *hotspot*. Here, the wizard provides a rich preview for each queried concept together with its current context as shown in Figure 3(a). We do not only describe a concept with its name and description, but also with additional features such as synonym labels and its neighborhood[6]. We limited the

[6] We define the *neighborhood* of a concept as all nodes that are one hop away via one of the following directed semantic relations: *isA, partOf, memberOf, substanceOf.*

neighborhood to these specified relations as those are the most important ones that we adopt from the external knowledge bases and they represent hierarchical links to more general concepts, which improves understanding.

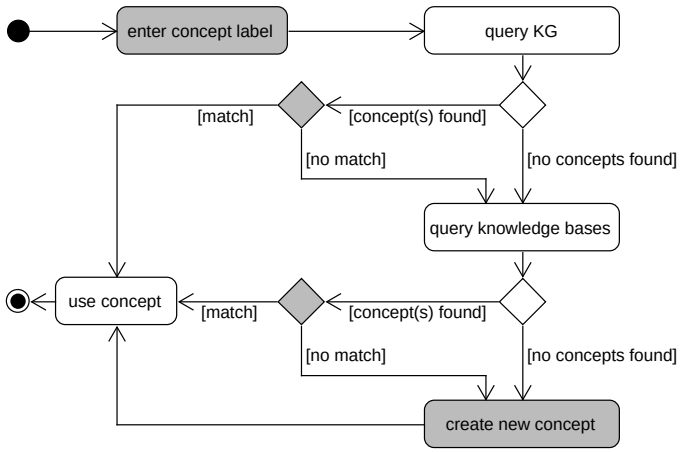



**Fig. 1.** Workflow we propose to find a concept for a label. User actions are marked with grey and automatic steps with white background. We first try to find a matching concept in the KG, then in external knowledge bases and as last resort, the user adds a concept manually.

If a concept is not present in the knowledge graph, it needs to be added. To propose concepts suggestions to the user, which are thoroughly specified according to the above criteria, the user wizard queries external knowledge bases based on the framework provided by Paulus et al. [34] as depicted in Figure 2. This framework already gathers and combines semantic concepts from multiple knowledge bases. We choose external knowledge sources, as it is not practical to prompt users to enter any synonyms they can think of, and to facilitate phrasing a description by suggesting a prefabricated one. In addition, we have the hypothesis that the use of such external knowledge bases results in more consistent and accurate semantic models, which will result in a stable ontology in ESKAPE's knowledge graph. Thus, if a concept suggestion from an external knowledge base matches the user's intention, they import it to their semantic model and use it. Subsequently, ESKAPE can add it to the ontology of its knowledge graph (cf. Section 4.3).

If the user decides to add concepts from external knowledge bases, we think it is advantageous for clarity to import the concept itself together with the gathered context, including synonyms, description and neighborhood that contains relations to more general concepts. This approach strives for a strong interconnection among concepts and avoids isolated concepts with little to no semantic significance. Since we automatically extract the neighborhood from external knowl-

edge bases, it might hold too many connected concepts to be comprehended, or might be confusing due to errors. To cope with such a gathered neighborhood that might not be flawless, we implement a *semi-supervised evolution strategy* in which the user decides for each axiom (relation to another concept) whether it meets their understanding or not, as depicted in Figure 3(b). This strategy enables us to improve the quality of the knowledge graph evolution as we only import knowledge that was inspected by the users. However, if none of the external knowledge bases contains the required concept, we finally allow the user to manually define a new one.

As soon as the user finishes the modeling process, the semantic model is submitted to the ESKAPE platform and is validated before the actual evolution of the knowledge graph's ontology takes place.

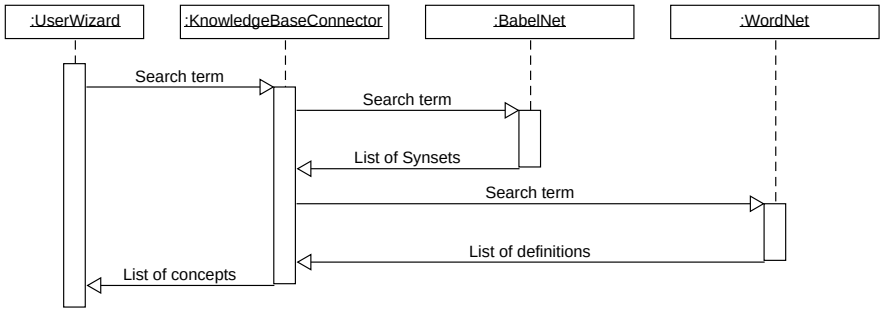
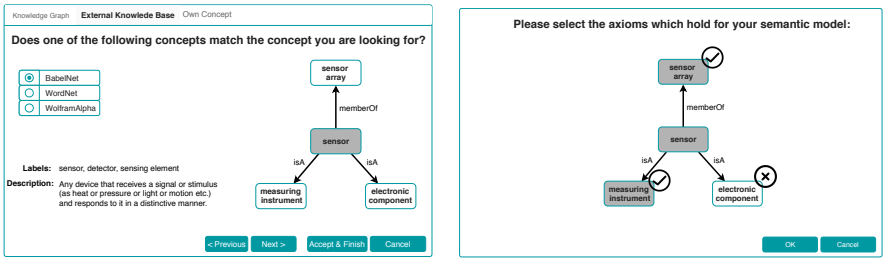



**Fig. 2.** Software architecture for retrieving a list of concept suggestions from multiple external knowledge bases, based on the framework provided by Paulus et al. in [34]. This is an exemplary setup with two knowledge bases activated, BabelNet and WordNet.

### 4.3 Knowledge Validation and Evolution

It is beneficial to detect and resolve inconsistencies or contradictions between the knowledge graph's ontology and the semantic model before it is added to the knowledge graph, in order to achieve high accuracy and to support understanding.

We classify problems that can occur in constructed semantic models into two categories, namely conflicts and inconsistencies. Conflicts are caused by modeling slips by the user in the semantic model itself, which are relations that are used wrongly. Examples are irreflexive relations that yet contain self-loops, asymmetric ones that are used in both directions, or a chain of generalizations via *isA*-relations that is a loop. Users are asked to review conflicts via a pop-up in the UI and can eliminate these by revising their semantic model. This might imply to add, remove or change relations, or to use different ones. Contrary, inconsistencies are lightweight problems, because they represent a disagreement




(a) Concept Visualization [38] (b) Neighborhood Selection [38]

**Fig. 3.** Screenshot of an example concept our implementation produces from the knowledge base *BabelNet*. The concept is placed in the center (a), with its main label and related concepts including their relations are arranged on a circle around. On the left we list synonym labels as well as a description. In a final step (b), the user chooses what parts of the neighborhood to import.

between the local understanding provided by the user as semantic model and universal facts stored in the ontology of the knowledge graph. It is tolerable that these two are contrary and we still allow the user to commit such a semantic model, after they confirmed that the stated axioms truly hold for their data set. A second possible inaccuracy occurs, when multiple local knowledge representations (semantic models) disagree with each other. That is, when several users have distinct (conflicting) statements, and we suggest a user to apply widely-used ones. Again, the user can still stick with their initial statement, because it might indeed hold for this particular data set.

If the user decides that the semantic model is complete and all problems are resolved with the help of our user wizard, the semantic model is finally committed to the knowledge graph, which saves it in order to share provided knowledge with others. This triggers the *Knowledge Evolution* component to continuously learn insights from the provided semantic models as follows: The *origin* field introduced in Section 4.1 determines whether a used concept or relation initially belongs to the knowledge graph's ontology or if the semantic model introduced it. Like the learning process of humans, it assimilates and monitors knowledge provided by users via semantic models, and subsequently learns insights that then expand the ontology of ESKAPE's knowledge graph. We realize this feature by additionally adding local weights to any relation in the knowledge base, and update a relation's weight whenever its adjacent concepts are used in a semantic model. By doing so, we enable a learning process to work with thresholds in order to strengthen or weaken semantic relations (flag/unflag as universal) based on their usage. The weights enable us to differentiate between frequently used knowledge, i.e., common knowledge multiple users agreed on, and specialized knowledge or inconsistencies, which is only valid in a small number of cases. In our current approach, we add new knowledge to the knowledge graph's ontology after it was specified in at least three semantic models provided by different users (all users are equal). We flag an axiom as universal when the majority (50%) of semantic models uses it.

This strategy is just one approach for dealing with upcoming inconsistencies and conflicts. We already thought about increasing/decreasing trust in users, depending on how they perform during the modeling process. However, we think that even trustworthy users might still conduct errors. That is why we evaluate if more users use the same concepts and relations for expressing their intention instead.

### 4.4 Autonomous Evolution from External Knowledge Bases

After semantic models have been stored in the knowledge graphs, it may be useful to determine whether other related semantic models already exist that use similar concepts. For example, suppose the knowledge graph contains a semantic model that describes a data set using WiFi access points, i.e. the concept *access point* was used in the semantic model. Now, a new semantic model is added to the knowledge graph that also contains data about WiFi access points. However, the data provider who created the semantic model did not want to use the suggested concept *access point*. Instead, the *hotspot* concept was preferred. Obviously, both data sets speak of the same data, but the providers have used different terms, i.e. the same heterogeneity problem that occurs when creating data schemes and models in general also applies for the knowledge graph. In order to directly identify these similarities, we include an autonomous learning step which runs as a batch job every 24 hours.

The goal of this job is to identify additional synonymous relationships between concepts that are already stored as universal facts stored in the ontology of the knowledge graph. Algorithm 1 shows our algorithm that we currently use to identify additional synonymous relationships. Since the ontology becomes quite large, we do only check new relationships for concepts that were newly added to the ontology or for those that were last checked 2 weeks ago. The reason why we also check old concepts is that the external databases we are using are also changing frequently. For the identified concepts, we query *BabelNet*[7], *WordNet*[8] and *Datamuse*[9]. To improve the quality of the identified synonyms, we only use synonyms that exist in at least two of the selected external databases. This implies that we might not get all possible synonyms, but, at the same time, it increases the probability that we get the right ones. After we identified the synonyms, we check if we already have the synonyms as concepts in the knowledge graph. If yes, we add a *synonymous edge* between the concepts if it does not already exist. If we do not have the concept in the knowledge, we do not add it. The reason for this is that we only want to add concepts that are really used by the community, i.e., we want to build a vocabulary that really represents the community. Otherwise, we could directly start to build a large knowledge graph that covers the knowledge from all external knowledge databases. Following this autonomous evolution strategy enables us to better connect the knowledge that is already part of the knowledge graph without just copying knowledge from external knowledge databases.

[7] https://babelnet.org
[8] https://wordnet.princeton.edu/
[9] https://www.datamuse.com/

**Algorithm 1** Synonym algorithm that is executed every 24 hours.

```
1:  for all concepts c in kgs_Ontology do
2:    if lastChecked c == NULL or lastChecked ≥ 336 hours then
3:      c_l = get label of c
4:
5:      datamuseSynonyms = query datamuse synonyms for c_l
6:      babelnetSynonyms = query babelnet synonyms for c_l
7:      wordnetSynonyms = query wordnet synonyms for c_l
8:
9:      i_1 = datamuseSynonyms ∩ babelnetSynonyms
10:     i_2 = datamuseSynonyms ∩ wordnetSynonyms
11:     i_3 = wordnetSynonyms ∩ babelnetSynonyms
12:
13:     i_all = i_1 ∪ i_2 ∪ i_3
14:     remove c_l from i_all
15:
16:     for all i_l in i_all do
17:       c_onto = query kgs_Ontology for i_l
18:       if c_onto not NULL then
19:         if c_onto not synonym of c_l then
20:           create synonymEdge(c_l,c_onto)
21:         end if
22:       end if
23:     end for
24:   end if
25: end for
```

## 5 Evaluation

In this section, we evaluate the design and implementation of our proposed approach. The goal is to evaluate how well our user wizard improves the quality of semantic models and subsequently the quality of the growing knowledge graph. Therefore, we set up a user study in which users created semantic models using our user wizard. To evaluate if our wizard increases the quality of the semantic models, users create two semantic models for each data set, once with and once without our wizard. We then compare these two with each other (A/B testing). In the following, we first describe our participant recruitment and setup, followed by chosen data sets, a definition of quality for semantic models that fits our evaluation and finally the evaluation results.

To perform the evaluation, we recruit seven persons from our research institute with different levels of experience in semantic modeling. Note that our institute is interdisciplinary, with an even share of humanities, engineering and computer science. Three persons are familiar with semantic modeling, whereas the other four are unexperienced in this field. We aim for this diversity in order to observe and equalize possible different styles in the semantic modeling and overall usage of our implemented approach. To assess the quality of the semantic

**Table 1.** Overview of the different data sets used for our evaluation, based on [38] and extended by three more data sets.

| Data Set | Source | Data Attributes |
|---|---|---|
| Crimes in Vancouver | Open Data Catalogue [5] | type, month, day, x, y, ... |
| Flight status | Knoblock et al. [39, 41] | airline, time, altitude, ... |
| Recycling sites | European Data Portal [8] | name, town, latitude, ... |
| Parking meter | DataSF [6] | cap color, time limit, ... |
| Greenhouse gas | DataSF [32] | CO2 emissions, quantity, ... |

models, we choose a qualitative evaluation. This evaluation requires comparing the attendees' semantic models pairwise, which is time-consuming. Thus, we decided to use a small group of participants.

All participants follow the same procedure on the same initial setup. We prepare a dedicated room with a laptop that runs ESKAPE. We limit the external knowledge bases in our evaluation to BabelNet to circumvent possible race conditions resulting from the merge algorithm of the underlying semantic concept gathering framework of Paulus et al. [34]. The attendees do not execute the steps of this study isolated, but have an evaluation expert sitting next to them for two purposes. First, to observe actions they make, where they face challenges and how they solve these, and second, to answer questions that are not related to the data itself (i.e., the user interface). For each participant, we follow an identical sequence to ensure that we can compare all results with each other. First, we start with a motivation why semantic modeling is important and why they want to annotate their data set with semantic concepts. Next, we give each attendee an introduction to ESKAPE covering its functionalities, behavior and input methods.

Table 1 depicts the five data sets we use, "Vancouver crime data", "Flight status", "Recycling sites", "Parking meter" and "Greenhouse gas". All are publicly available, are in English language and provide six to ten data attributes used for semantic model bootstrapping. The data sets are from different domains and we provide the participants with two files for each data set. First, the raw data set itself in form of an Excel sheet, which contains headers (column names) as well as many data points (rows). To properly represent the fact that users mostly are familiar with the data sets they operate with, we provide a detailed description, which contains definitions for all included data attributes and describes their purpose.

All persons conduct the evaluation the same way. Each participant starts with an empty knowledge graph in ESKAPE and first creates semantic models for one data set and then continuous with the others without our user wizard. In this unsupervised approach, the participant must create each concept manually, since no concepts are already present. We then clear the knowledge graph and the user again creates semantic models for all five data sets, but with the help of our user wizard. The wizard allows users to select what additional knowledge to add to the knowledge graph in order to achieve a decent but precise growth in semantic

information with exactly the concepts and relations the user chooses. We store snapshots of both the knowledge graph state and created semantic models after each step in order to analyze the results. The goal of this evaluation is to test our hypothesis: Whether semantic models are more objective and consistent when created with our user wizard and whether it improves the quality of the growing knowledge graph.

### 5.1 Analysis and Discussion

Current research in the field of semantic modeling suggests different measurements for the quality of semantic models. As stated by Paulheim, there is no gold standard for a semantic model [33]. Measurements based on precision and recall are limited in the semantic modeling field, because determining the recall would include to tell the number of positive results that should have been added to the semantic model. This is complex or even impossible due to the non-existence of evidence to tell how many further concepts are needed in order to describe one concept's meaning to satisfaction, because there are infinitely many possible candidates.

Therefore, we consider a user's semantic model in this evaluation as good if it has a large overlap with other's semantic models for the same data set. We strive for objective and unified semantic models in order to achieve a stable growth of the knowledge graph, including important expert knowledge but excluding personal views not truly related to the content. We believe to achieve the best knowledge graph growth by adding semantic models that have much in common and thus create strong interconnectivity, while strongly personalized models would create isolated subgraphs.

For our purposes, we use the Jaccard index [17] to measure similarity between semantic models. This calculation considers the number of concepts two semantic models have in common. We judge two concepts $c_1$ and $c_2$ from the respective data set one and two as the same if they share exactly the same name. In addition, we introduce advanced rules to detect two equal concepts, which we derived from previous semantic modeling observations. These common behaviors are (a) capitalization like *flight* vs. *Flight*, (b) word spacing like *flight number* vs. *flightnumber*, (c) camel case usage as combination of these like *FlightNumber* vs. *flight number* and (d) underscores like *flight_ number* vs. *flight number*. We thus also detect two concepts from different participants as the same, whenever they match one of the above rules. Paulus et al. describe in [34] how one can unify these possible inconsistencies for names and data attributes.

The Jaccard index is defined as ratio of number of elements two sets have in common and the number of elements in their union. The formula to calculate the Jaccard index of two sets is $J(A,B) = \frac{A \cap B}{A \cup B}$. We let A and B be the concepts that occur in two semantic models, respectively. The Jaccard index is zero when two semantic models do not share any concept and it is one when two semantic models have all their concepts in common. For this evaluation, we measure the average Jaccard index (similarity) between every semantic model and all other ones for the same setting (same data set and same un-/assisted state). For data

set one in the unassisted setting, we calculate the Jaccard index between participant 1's semantic model with those of the others, followed by participant 2's, and so on.

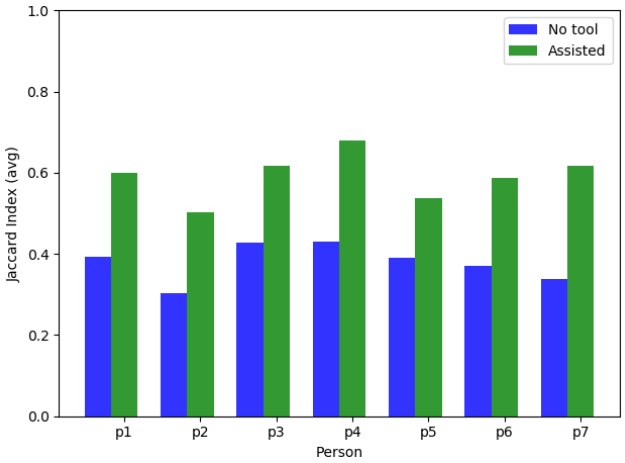



**Fig. 4.** Evaluation results: The average Jaccard index of each attendee's semantic model with those of others for data set one, comparing unassisted and assisted modeling [38].

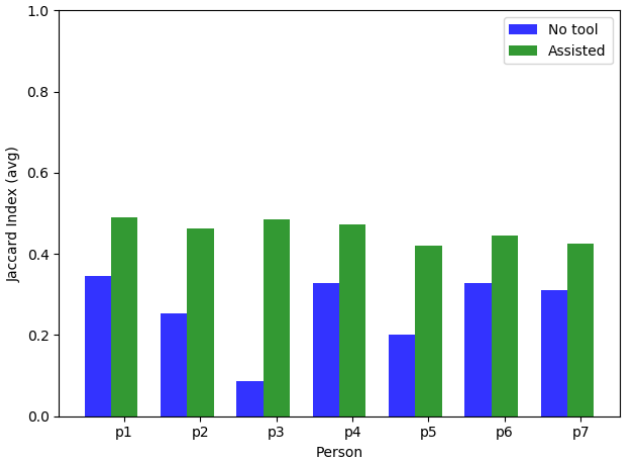



**Fig. 5.** Evaluation results: The average Jaccard index of each attendee's semantic model with those of others for data set two, comparing unassisted and assisted modeling [38].

Figure 4 shows for data set one the average Jaccard indexes for each participant when comparing the no tool setup with the assisted run, while Figure 5 shows the results for data set two, respectively. The results show for both data sets that number of shared concepts between the different semantic models increases if the participants use our user wizard. This confirms our hypothesis that semantic models are more consistent and objective if the users have access to additional knowledge that supports them during the semantic model creation. Since the number of different used concepts decreases by using our user wizard, we argue that the stability of the knowledge graph's ontology will subsequently increase since the community agrees on a common vocabulary.

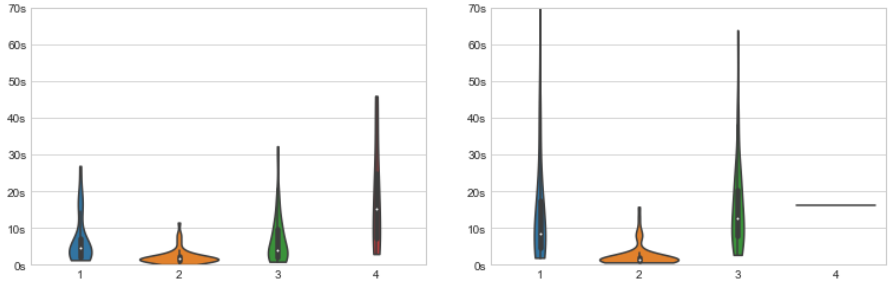
### 5.2 Additional Observations

Beside the primary results, we gathered additional insights during this evaluation. When users used our wizard (cf. Figure 3) to find the right concept for annotation and when they came to the step that neither knowledge graph nor external knowledge bases could provide a concept that matches their understanding, multiple users did go back one step multiple times for searching a different term. This strengthens our hypothesis that the community will agree on a common vocabulary over time and thus will build more consistent and objective semantic models. As soon as users introduced their own concept, i.e., they did not use any suggested one, they specified a name but usually did not enter a proper description. For instance, they entered descriptions like *airline: "airline of a flight"*. It would be desirable to enter more detailed descriptions as those would help other users for reusing the concept later on. Moreover, the participants used relations in different directions when connecting multiple properties with a central node representing the core of a data set. For example, we found instances of semantic models with relations from a general concept to all of its properties with relations, such as *has*, while others point them the other way around with a respective relation like *partOf*. When asked how challenging the semantic modeling process for the provided data sets are, five participants stated that it would surely be possible to create much bigger semantic models, whereas two people said that the limit of complexity is already reached.

In addition to the above mentioned additional insights, we investigated one extra hypothesis: *People with more experience in semantic modeling create more sophisticated semantic models.* This includes two points. On the one hand, they take their time for choosing the concept that matches perfectly instead of just finishing the semantic model somehow. On the other hand, semantic modeling experts choose to reuse existing concepts from either the knowledge graph or external ontologies over manually introducing new ones whenever possible. This is most probably because they know about the advantages of shared common concepts compared to concepts that were created according to different people's own intuition. We evaluated this hypothesis via two measurements during our user studies, namely precise time measurements of each step in the user wizard and tracking which of the suggested concepts the users chose.

Figure 6 depicts the time semantic expert participants spent compared to those with no expertise in semantic modeling, called nonspecialists. There are

significant differences for determining the right concept label (page 1), where nonspecialist mostly decided in about 7 seconds and never took longer than 27 seconds. Experts took more time, which is 14 seconds in average and up to 125 seconds, which even exceeds the chosen graph limits for this figure due to readability. There is no significant difference for the selection of concepts from the local knowledge graph (page 2), because all experiments started with an empty one and this wizard page has no benefit. Selecting suggested concepts from external knowledge bases (page 3) again leads to large variations between the two user groups. Nonspecialists took 7 seconds on average and 32 seconds maximum to select the best matching concepts from the given suggestions, while one user indeed finished the selection after only 0.8 seconds. Experts took 15.9 seconds on average and up to 64 seconds. Creating a novel concept took nonspecialists between 3 and 46 seconds, with a mean of about 18, while experts only created one novel concept within 16 seconds.




(a) Nonspecialists time spent per wizard page (b) Experts time spent per wizard page

**Fig. 6.** Violin plots of time spent in seconds per wizard page (cf. Figure 3) for (a) nonspecialists and (b) experts. Experts took significantly more time for defining the search term (page 1) and selecting the best matching concept from external sources (page 3). Nonspecialists tend to create new concepts (page 4) while experts more likely reuse existing ones instead.

We took into account a second measurement besides the quantitative evaluation of time spent on the wizard pages. In this qualitative evaluation, we investigate how users formulate their search terms when searching for matching concepts. A selection of results is shown in Table 2, where nonspecialists simply adopt the given label from the data set as their search term. This behavior leads to bad results in terms of expressiveness, because the search terms do not properly represent the concepts they are looking for. For the labels in Table 2, the search term "X" returned nine concepts from external sources, but non matched and thus a new concept was created manually. The rest of the nonspecialist's search terms could not be found in any external knowledge base and needed to be created. On the other side, the expert user replaced given labels with their

**Table 2.** Qualitative evaluation of entered search terms for given labels. Nonspecialists mostly adopt the given label, whereas experts formulate a detailed search term.

| Label | X | Post ID | Cap Color | Quantity_Units | Flight status |
|---|---|---|---|---|---|
| Nonspecialist | X | Post ID | Cap Color | Quantity_Units | Flight status |
| Expert | Coordinate | Identifier | Color | Unit | Status |

actual concepts, such as "color" instead of "cap color" and thereby found and reused existing concepts from external knowledge bases.

The quantitative and qualitative observations confirm our hypothesis that *people with more experience in semantic modeling create more sophisticated semantic models*. We compared the behavior of semantic modeling expert with nonspecialists in different ways and got the following results. Nonspecialists tend to put less effort into executing tasks within the semantic modeling process, resulting in worse semantic models in terms of precision and objectiveness. They often get no results from external knowledge bases, because the simply adopt the given column labels of a data set as search terms, while experts precisely formulate these in order to find the correct concepts. Experts spend much more time for selecting the right concept from suggestions from external knowledge bases (cf. Figure 6), while nonspecialists often just pick the first suggestion to continue quickly. Although our goal was to create an approach that enables domain experts without semantic model experience to describe data sets on their own, our evaluation showed that, as expected, experienced users build more sophisticated models. This means that we need to put more effort into supporting nonspecialists.

In addition to the evaluation of the results, we want to discuss two limitations of our user study. First, we simulated the situation where a user has a data set at hand and wants to annotate it in order to publish it via a semantic data platform. Different from the real world use case, our participants are not familiar with the data set and its meaning. We provide meta-data and additional information together with the data sets to handle this, but pay with extra time the attendees require in order to understand meaning of each data sets properly. Second, we focus on detecting the average Jaccard index for semantic models and compare the scenarios supported by our user wizard with those where no wizard was used. This evaluation implies to reset the knowledge graph between test runs to achieve comparability of these. Therefore, we are not able to monitor a comprehensive knowledge graph growth over a long period in which different users work on the same knowledge graph. To evaluate the knowledge graph growing characteristics at glance, we indeed plan a long-term testing as well as more participants and data sets.

## 6 Conclusion

In this paper, we explored the need for OBDM approaches to create sophisticated ontologies in advance, as they are required to define a mapping between data sets and a domain ontology. In order to reduce the effort required to design and

manage such ontologies, we developed an approach that consists of a knowledge graph, which includes an internally growing ontology and data source-specific semantic models. To achieve this goal, we presented a multi-folded approach, which is advantageous in several points. First, we developed an intuitive user-oriented wizard. The wizard guides the user through the construction of semantic models by recommending knowledge that is already available in the knowledge graph as well as by assisting him in creating new concepts with the help of external knowledge databases. Second, we implemented a semi-supervised strategy in the user wizard, which lets the user exactly choose what parts of external knowledge they want to introduce so that not only the original concept but also additional context is imported into the knowledge graph. Third, we ensure the completeness of all currently stored semantic models through further enhancements as soon as a new semantic model requires knowledge that is still missing in the knowledge graph. We have therefore added a validation step that recognizes and solves problems with semantic models based on previously gained knowledge stored in the knowledge graph's ontology. Hence, all these building blocks ensure that we add the knowledge in a controlled and validated manner. Nevertheless, these approaches might not be capable of identifying further synonymous relationships that might be available in the graph. For that, we additionally provided an algorithm that improves the density of the ontology by automatically identifying synonymous relationships with the help of external knowledge databases.

We conducted a user study to evaluate how well our user wizard improves the quality of semantic models and thus the quality of growing knowledge graphs. The idea of this evaluation was to evaluate that user build more consistent semantic models if they are assisted by a user wizard that helps them to introduce new knowledge. The results of our evaluation indeed show that users' semantic models actually become more consistent and objective when they are created using our user wizard. This results in a knowledge graph with higher interconnectivity and stability. At the same time, the evaluation also showed that users with experience in semantic modeling build more sophisticated semantic models, which means, that we need to improve our wizard to better support nonspecialists.

Hence, as future work, we plan to improve the user wizard and its usability for daily use. To additional improve the quality of semantic models, we plan to add a recommendation engine that continuously suggests concepts and relationships that might be useful for this data set. In addition, we found that when people enter useful information into the description field, the field can be used to identify additional context about the data set. Therefore, our goal is furthermore improve the semantic model creation and recommendation by leveraging techniques from the area of natural language processing. In addition, we plan to conduct a more detailed user study in the future, in which we will observe and evaluate the growth of the knowledge graph in more detail over a longer period of time. The extended user study will include more participants to make it more representative, as well as the use of several external knowledge databases. We are also aware that ontology evolution does not only consist of

adding knowledge. Removing knowledge as well as changing knowledge over time are required as well. Our approach does not yet support this, but we are working on a solution in which in the course of time all universal aspects of knowledge can be forgotten. The last part of work we are planning to add in the near future is the creation of semantic models for unstructured data, such as textual documents, images or videos. Here, we are planning to apply machine learning techniques around object detection in order to detect and annotate objects as well as techniques around natural language processing in order to extract information from the text. Identified concepts in these documents can then be added to the knowledge graph again, like it is done in the area of ontology learning as well as knowledge graph construction.

## References

1. Abbes, H., Boukettaya, S., Gargouri, F.: Learning Ontology from Big Data through MongoDB Database. In: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA). pp. 1–7 (2015). https://doi.org/10.1109/AICCSA.2015.7507166
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The semantic web, pp. 722–735. Springer (2007)
3. Braun, G.A., Cecchi, L.A., Fillottrani, P.R.: Integrating Graphical Support with Reasoning in a Methodology for Ontology Evolution Winter of Ontology co-located with the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, July 25-27, 2015. In: Papini, O., Benferhat, S., Garcia, L., Mugnier, M.L., Fermé, E.L., Meyer, T., Wassermann, R., Hahmann, T., Baclawski, K., Krisnadhi, A., Klinov, P., Borgo, S., Kutz, O., Porello, D. (eds.) Proceedings of the Joint Ontology Workshops 2015 Episode 1: The Argentine Winter of Ontology co-located with the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, July 25-27, 2015. CEUR Workshop Proceedings, CEUR-WS.org (2015), `http://ceur-ws.org/Vol-1517/JOWO-15_WoMO_paper_5.pdf`
4. Calvanese, D., de Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., Savo, D.F.: The MASTRO System for Ontology-Based Data Access. Semantic web **2**(1), 43–53 (2011)
5. City of Vancouver: Crime 2017. `https://data.vancouver.ca/datacatalogue/crime-data.htm` (2018), [Online; accessed 25-September-2018]
6. Drew Taylor, SFpark (info@sfpark.org): Meter operating schedules. `https://data.sfgov.org/Transportation/Meter-Operating-Schedules/6cqg-dxku` (2014), [Online; accessed 23-July-2019]
7. Du, W.H., Rau, J.W., Huang, J.W., Chen, Y.S.: Improving the quality of tags using state transition on progressive image search and recommendation system. In: Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on. pp. 3233–3238. IEEE (2012)
8. East Sussex County Council: Waste and recycling sites in east sussex. `https://www.europeandataportal.eu/data/?#/datasets/east-sussex-county-council-recycling-sites`, [Online; accessed 23-July-2019]
9. Gil, Y., Garijo, D., Ratnakar, V., Khider, D., Emile-Geay, J., McKay, N.: A controlled crowdsourcing approach for practical ontology extensions and metadata

annotations. In: International Semantic Web Conference. pp. 231–246. Springer (2017)
10. Goel, A., Knoblock, C.A., Lerman, K.: Exploiting structure within data for accurate labeling using conditional random fields. In: Proceedings on the International Conference on Artificial Intelligence (ICAI). p. 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) (2012)
11. Gruber, T.: What is an ontology. WWW Site http://www-ksl.stanford.edu/kst/whatis-an-ontology.html (accessed on 24-01-2018) (1993)
12. Haase, P., Horrocks, I., Hovland, D., Hubauer, T., Jiménez, E., Kharlamov, E., Klüwer, J., Pinkel, C., Rosati, R., Santarelli, V., et al.: Optique System: Towards Ontology and Mapping Management in OBDA Solutions (2013)
13. Hai, R., Geisler, S., Quix, C.: Constance: An Intelligent Data Lake System. In: Proceedings of the 2016 International Conference on Management of Data. pp. 2097–2100 (2016)
14. He, S., Zou, X., Xiao, L., Hu, J.: Construction of diachronic ontologies from people's daily of fifty years. In: LREC. pp. 3258–3263 (2014)
15. Heyvaert, P., Dimou, A., Verborgh, R., Mannens, E.: Ontology-based data access mapping generation using data, schema, query, and mapping knowledge. In: Blomqvist, E., Maynard, D., Gangemi, A., Hoekstra, R., Hitzler, P., Hartig, O. (eds.) The Semantic Web. pp. 205–215. Springer International Publishing, Cham (2017)
16. Hu, Y., Janowicz, K.: Enriching Top-Down Geo-Ontologies Using Bottom-Up Knowledge Mined from Linked Data. Advancing Geographic Information Science: The Past and Next Twenty Years **183** (2016)
17. Jaccard, P.: Nouvelles recherches sur la distribution florale. Bull. Soc. Vaud. Sci. Nat. **44**, 223–270 (1908)
18. Jiménez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., Horrocks, I., Pinkel, C., Skjæveland, M.G., Thorstensen, E., Mora, J.: BootOX: Practical Mapping of RDBs to OWL 2. In: International Semantic Web Conference. pp. 113–132 (2015)
19. Kalender, M., Dang, J.: Skmt: A semantic knowledge management tool for content tagging, search and management. In: 2012 Eighth International Conference on Semantics, Knowledge and Grids (SKG). pp. 112–119. IEEE (2012)
20. Kalender, M., Dang, J., Uskudarli, S.: Unipedia: A unified ontological knowledge platform for semantic content tagging and search. In: 2010 IEEE Fourth International Conference on Semantic Computing (ICSC). pp. 293–298. IEEE (2010)
21. Kharlamov, E., Mailis, T., Mehdi, G., Neuenstadt, C., Özçep, Ö., Roshchin, M., Solomakhina, N., Soylu, A., Svingos, C., Brandt, S., Giese, M., Ioannidis, Y., Lamparter, S., Möller, R., Kotidis, Y., Waaler, A.: Semantic Access to Streaming and Static Data at Siemens. Journal of Web Semantics **44**, 54–74 (2017). https://doi.org/10.1016/j.websem.2017.02.001
22. Kotis, K., Papasalouros, A., Maragoudakis, M.: Mining Query-Logs Towards Learning Useful Kick-off Ontologies: an Incentive to Semantic Web Content Creation. Int. J. Knowl. Eng. Data Min. **1**(4), 303–330 (2011). https://doi.org/10.1504/IJKEDM.2011.040652, `http://dx.doi.org/10.1504/IJKEDM.2011.040652`
23. Laniado, D., Eynard, D., Colombetti, M., et al.: Using wordnet to turn a folksonomy into a hierarchy of concepts. In: Semantic Web Application and Perspectives-Fourth Italian Semantic Web Workshop. pp. 192–201 (2007)
24. Lehmann, J., Voelker, J.: An Introduction to Ontology Learning. Perspectives on Ontology Learning. Amsterdam: IOS Press (2014)

25. Lenzerini, M.: Ontology-Based Data Management. In: Ounis, I., Ruthven, I., Macdonald, C. (eds.) Proceedings of the 20th ACM International Conference on Information and Knowledge Management - CIKM '11. p. 5. ACM Press, New York, New York, USA (2011). https://doi.org/10.1145/2063576.2063582
26. Lenzerini, M.: Ontology-Based Data Management: Keynote (2013), http://ceur-ws.org/Vol-866/keynote3.pdf
27. Mahdisoltani, F., Biega, J., Suchanek, F.: Yago3: A knowledge base from multilingual wikipedias. In: 7th Biennial Conference on Innovative Data Systems Research. CIDR Conference (2014)
28. Melo, A., Paulheim, H.: Detection of relation assertion errors in knowledge graphs. In: Proceedings of the Knowledge Capture Conference. p. 22. ACM (2017)
29. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: Proceedings of the 17th ACM conference on Information and knowledge management. pp. 509–518. ACM (2008)
30. Milne, D., Witten, I.H.: An open-source toolkit for mining wikipedia. Artificial Intelligence **194**, 222–239 (2013)
31. Navigli, R., Ponzetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence **193**, 217–250 (2012)
32. OpenData: San francisco communitywide greenhouse gas inventory. https://data.sfgov.org/Energy-and-Environment/San-Francisco-Communitywide-Greenhouse-Gas-Invento/btm4-e4ak (2016), [Online; accessed 23-July-2019]
33. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web **8**(3), 489–508 (2017)
34. Paulus, A., Pomp, A., Poth, L., Lipp, J., Meisen, T.: Gathering and Combining Semantic Concepts from Multiple Knowledge Bases. In: Proceedings of the 20th International Conference on Enterprise Information. SCITEPRESS - Science and Technology Publications (2018), to appear
35. Pomp, A., Paulus, A., Jeschke, S., Meisen, T.: ESKAPE: Information Platform for Enabling Semantic Data Processing. In: Proceedings of the 19th International Conference on Enterprise Information. SCITEPRESS - Science and Technology Publications (Apr 2017)
36. Pomp, A., Paulus, A., Kirmse, A., Kraus, V., Meisen, T.: Applying semantics to reduce the time to analytics within complex heterogeneous infrastructures. Technologies **6**(3) (2018). https://doi.org/10.3390/technologies6030086, http://www.mdpi.com/2227-7080/6/3/86
37. Pomp, A., Paulus, A., Klischies, D., Schwier, C., Meisen, T.: A Web-based UI to Enable Semantic Modeling for Everyone. In: SEMANTiCS 2018 – 14th International Conference on Semantic Systems (2018)
38. Pomp, A., Lipp, J., Meisen, T.: You are Missing a Concept! Enhancing Ontology-Based Data Access with Evolving Ontologies. In: 2019 IEEE 13th International Conference on Semantic Computing (ICSC). pp. 98–105. IEEE (2019). https://doi.org/10.1109/ICOSC.2019.8665620
39. Ramnandan, S.K., Mittal, A., Knoblock, C.A., Szekely, P.: Assigning semantic labels to data sources. In: European Semantic Web Conference. pp. 403–417. Springer (2015)
40. Rümmele, N., Tyshetskiy, Y., Collins, A.: Evaluating Approaches for Supervised Semantic Labeling. CoRR **abs/1801.09788** (2018)

41. S. K. Ramnandan, A. Mittal, C. A. Knoblock, and P. Szekely: Flight status. `https://github.com/usc-isi-i2/eswc-2015-semantic-typing/blob/master/Datasets/Files_flightstatus/f1.txt` (2015), [Online; accessed 25-September-2018]
42. Sadeghi, A., Lange, C., Vidal, M.E., Auer, S.: Integration of scholarly communication metadata using knowledge graphs. In: International Conference on Theory and Practice of Digital Libraries. pp. 328–341. Springer (2017)
43. Savo, D.F., Lembo, D., Lenzerini, M., Poggi, A., Rodrıguez-Muro, M., Romagnoli, V., Ruzzi, M., Stella, G.: MASTRO at Work: Experiences on Ontology-Based Data Access. Proc. of DL **573**, 20–31 (2010)
44. Taheriyan, M., Knoblock, C.A., Szekely, P., Ambite, J.L.: Learning the semantics of structured data sources. Web Semantics: Science, Services and Agents on the World Wide Web **37**, 152–169 (2016)
45. Torres, D., Diaz, A., Skaf-Molli, H., Molli, P.: Semdrops: A social semantic tagging approach for emerging semantic data. In: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01. pp. 340–347. IEEE Computer Society (2011)
46. Zhang, J., Xiong, M., Yu, Y.: Mining Query Log to Assist Ontology Learning from Relational Database. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) Frontiers of WWW Research and Development - APWeb 2006. pp. 437–448. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)